Cloud Search Service

User Guide

Issue 01

Date 2026-10-26



HUAWEI

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2026. All rights reserved.

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

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Using IAM to Grant Access to CSS 1
1.1 Creating IAM Users and Granting Them Permissions to Use CSS 1
1.2 Example: Granting IAM Users the Permission to Create CSS Clusters 4
2 Elasticsearch 11
2.1 Procedure 11
2.2 Cluster Planning 15
2.2.1 Planning Cluster AZs and HA 15
2.2.2 Planning the Cluster Version and Security Mode 19
2.2.3 Planning Node Types and Resource Specifications 23
2.3 Creating a Cluster (New Version) 34
2.4 Creating a Cluster 52
2.5 Network Settings 63
2.5.1 Overview 63
2.5.2 Configuring Public Network Access 65
2.5.3 Configuring a Dedicated Load Balancer 68
2.5.4 Configuring VPC Endpoints 80
2.5.5 Configuring Routes for a Cluster 84
2.6 Security Settings 86
2.6.1 Creating Cluster Users and Granting Permissions 86
2.6.2 Configuring LDAP Authentication 97
2.7 Connecting to Clusters 102
2.7.1 Cluster Connection Methods 102
2.7.2 Connecting to a Cluster Through Kibana 105
2.7.3 Connecting to a Cluster Through Cerebro 110
2.7.4 Connecting to a Cluster Using Java 110
2.7.4.1 Connecting to a Cluster Using the High Level REST Client 111
2.7.4.2 Connecting to a Cluster Using the Low Level REST Client 119
2.7.4.3 Connecting to a Cluster Through the Transport Client 136
2.7.4.4 Connecting to a Cluster Through Spring Boot 137
2.7.4.5 Connecting to a Cluster Through the Elasticsearch Java API Client 144
2.7.5 Connecting to a Cluster Using Python 157
2.7.6 Connecting to a Cluster Using Go 161
2.7.7 Connecting to a Cluster Using PHP 164

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2.7.8 Connecting MRS Hive to a Cluster 171
2.7.9 Connecting to a Cluster Using cURL Commands 177
2.8 Ingesting Data 179
2.8.1 Data Ingestion Methods 179
2.8.2 Migrating Data Using CDM 180
2.8.3 Ingesting Data Using Self-Managed Logstash 183
2.8.4 Ingesting Data Using Open-Source Elasticsearch APIs 188
2.9 Managing the Index Lifecycle 193
2.9.1 Configuring Index State Management (ISM) Policies 193
2.9.2 Automating Index Rollover Using ISM 199
2.9.3 Decoupling Storage and Compute Using ISM 204
2.10 Search and Analytics 206
2.10.1 Using DSL for Search 206
2.10.2 Using SQL for Search 209
2.10.3 Configuring Custom Dictionaries 214
2.10.4 Configuring Simplified-Traditional Chinese Conversion Search 225
2.11 Core Service Enhancements 233
2.11.1 CSS Enhancements Over Open-Source Elasticsearch 234
2.11.2 Configuring Decoupled Storage and Compute 237
2.11.3 Switching Between Hot and Cold Data 254
2.11.4 Configuring Read/Write Splitting Between Leader and Follower 259
2.11.5 Configuring Flow Control 2.0 282
2.11.6 Configuring Flow Control 1.0 298
2.11.7 Configuring Query Traffic Isolation 313
2.11.8 Configuring Large Query Isolation 316
2.11.9 Configuring the Kibana Query Response Circuit Breaker 326
2.11.10 Configuring Automatic Retry for Failed Shard Allocation 330
2.11.11 Enhancing Data Ingestion Performance 334
2.11.12 Configuring Enhanced Aggregation 341
2.11.13 Enabling the Index Recycle Bin 353
2.11.14 Tracking Query Resource Consumption 358
2.12 Monitoring 367
2.12.1 Using Cloud Eye to Monitor Clusters 367
2.12.2 Monitoring P99 Latency and HTTP Status Codes 371
2.12.3 Configuring Index-Level Monitoring 377
2.12.4 Supported Metrics 407
2.13 Managing Logs 464
2.13.1 Querying Logs 464
2.13.2 Ingesting Logs into a Cluster 470
2.13.3 Backing Up Logs to OBS 472
2.13.4 Viewing Audit Logs 477
2.14 Managing Alarms 483

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2.14.1 Configuring Open Distro Alert Notifications (via SMN) 483
2.14.2 Synchronizing Alerts and Metrics to Prometheus 489
2.15 Intelligent Diagnostics 494
2.16 Managing Cluster Changes 502
2.16.1 Scaling Out/Up 502
2.16.2 Scaling In/Down 508
2.16.3 Changing Node Specifications 513
2.16.4 Replacing a Faulty Node 520
2.16.5 Upgrading the Cluster Version 523
2.16.6 Switching AZs 534
2.16.7 Changing the Security Mode 538
2.17 Backup and Restoration 545
2.17.1 Creating Snapshots to Back Up Data 545
2.17.2 Restoring Data Using Snapshots 554
2.17.3 Deleting a Cluster Snapshot 559
2.18 Managing Clusters 560
2.18.1 Viewing Cluster Details 560
2.18.2 Checking Default Cluster Plugins 567
2.18.3 Modifying Cluster Parameters 568
2.18.4 Managing Cluster Tags 574
2.18.5 Associating a Cluster with an Enterprise Project 576
2.18.6 Restarting a Cluster 578
2.18.7 Deleting a Cluster 583
3 OpenSearch 585
3.1 Procedure 585
3.2 Cluster Planning 589
3.2.1 Planning Cluster AZs and HA 589
3.2.2 Planning the Cluster Version and Security Mode 593
3.2.3 Planning Node Types and Resource Specifications 598
3.3 Creating a Cluster (New Version) 608
3.4 Creating a Cluster 626
3.5 Network Settings 635
3.5.1 Overview 636
3.5.2 Configuring Public Network Access 637
3.5.3 Configuring a Dedicated Load Balancer 640
3.5.4 Configuring VPC Endpoints 651
3.5.5 Configuring Routes for a Cluster 655
3.6 Security Settings 657
3.6.1 Creating Cluster Users and Granting Permissions 657
3.6.2 Configuring LDAP Authentication 668
3.7 Connecting to Clusters 673
3.7.1 Cluster Connection Methods 673

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3.7.2 Connecting to Cluster Through OpenSearch Dashboards 676
3.7.3 Connecting to a Cluster Through Cerebro 678
3.7.4 Connecting to a Cluster Using Java 679
3.7.4.1 Connecting to a Cluster Using the High Level REST Client 679
3.7.4.2 Connecting to a Cluster Using the Low Level REST Client 697
3.7.4.3 Connecting to a Cluster Through the Elasticsearch Java API Client 731
3.7.5 Connecting to a Cluster Using Python 744
3.7.6 Connecting to a Cluster Using cURL Commands 750
3.8 Ingesting Data 752
3.8.1 Data Ingestion Methods 752
3.8.2 Migrating Data Using CDM 753
3.8.3 Ingesting Data Using Self-Managed Logstash 755
3.8.4 Ingesting Data Using Open-Source OpenSearch APIs 761
3.9 Managing the Index Lifecycle 764
3.9.1 Configuring Index State Management (ISM) Policies 764
3.9.2 Automating Index Rollover Using ISM 770
3.9.3 Decoupling Storage and Compute Using ISM 774
3.10 Search and Analytics 777
3.10.1 Using DSL for Search 777
3.10.2 Using SQL for Search 779
3.10.3 Configuring Custom Dictionaries 784
3.10.4 Configuring Simplified-Traditional Chinese Conversion Search 795
3.10.5 Implementing OpenSearch AI Search with a Search Foundation Model 803
3.11 Core Service Enhancements 830
3.11.1 CSS Enhancements Over Open-Source OpenSearch 830
3.11.2 Configuring Decoupled Storage and Compute 833
3.11.3 Switching Between Hot and Cold Data 850
3.11.4 Configuring Read/Write Splitting Between Leader and Follower 855
3.11.5 Configuring Flow Control 878
3.11.6 Configuring Query Traffic Isolation 894
3.11.7 Configuring Large Query Isolation 897
3.11.8 Configuring the OpenSearch Dashboards Query Response Circuit Breaker 907
3.11.9 Configuring Automatic Retry for Failed Shard Allocation 911
3.11.10 Enhancing Data Ingestion Performance 915
3.11.11 Configuring Enhanced Aggregation 922
3.11.12 Enabling the Index Recycle Bin 930
3.11.13 Tracking Query Resource Consumption 935
3.12 Monitoring 946
3.12.1 Using Cloud Eye to Monitor Clusters 946
3.12.2 Configuring Index-Level Monitoring 951
3.12.3 Supported Metrics 959
3.13 Managing Logs 1019

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3.13.1 Querying Logs ... 1019
3.13.2 Ingesting Logs into a Cluster ... 1025
3.13.3 Backing Up Logs to OBS ... 1027
3.13.4 Viewing Audit Logs ... 1032
3.14 Managing Alarms ... 1038
3.14.1 Configuring OpenSearch Alert Notifications via SMN ... 1039
3.14.2 Synchronizing Alerts and Metrics to Prometheus ... 1052
3.15 Intelligent Diagnostics ... 1059
3.16 Managing Cluster Changes ... 1067
3.16.1 Scaling Out/Up ... 1067
3.16.2 Scaling In/Down ... 1073
3.16.3 Changing Node Specifications ... 1077
3.16.4 Replacing a Faulty Node ... 1085
3.16.5 Upgrading the Cluster Version ... 1088
3.16.6 Switching AZs ... 1098
3.16.7 Changing the Security Mode ... 1102
3.17 Backup and Restoration ... 1109
3.17.1 Creating Snapshots to Back Up Data ... 1109
3.17.2 Restoring Data Using Snapshots ... 1118
3.17.3 Deleting a Cluster Snapshot ... 1123
3.18 Managing Clusters ... 1124
3.18.1 Viewing Cluster Details ... 1124
3.18.2 Checking Default Cluster Plugins ... 1131
3.18.3 Modifying Cluster Parameters ... 1132
3.18.4 Managing Cluster Tags ... 1138
3.18.5 Associating a Cluster with an Enterprise Project ... 1140
3.18.6 Restarting a Cluster ... 1142
3.18.7 Deleting a Cluster ... 1147

4 Logstash ... 1149

4.1 Procedure ... 1149
4.2 Cluster Planning ... 1151
4.2.1 Planning Cluster AZs and HA ... 1151
4.2.2 Planning Node Specifications and Capacity ... 1152
4.3 Creating a Cluster (New Version) ... 1156
4.4 Creating a Cluster ... 1161
4.5 Network and Security Settings ... 1164
4.5.1 Configuring Public Network Access ... 1164
4.5.2 Configuring Routes for a Cluster ... 1167
4.5.3 Managing CA Certificates ... 1169
4.6 Configuring Migration Tasks ... 1173
4.6.1 Creating a Configuration File ... 1173
4.6.2 Starting Migration Tasks ... 1178

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

4.6.3 Stopping Migration Tasks 1181
4.6.4 Configuration File Templates 1183
4.7 Monitoring 1199
4.7.1 Using Cloud Eye to Monitor Clusters 1200
4.7.2 Supported Metrics 1205
4.8 Managing Logs 1214
4.8.1 Querying Logs 1214
4.8.2 Backing Up Logs to OBS 1216
4.8.3 Viewing Audit Logs 1220
4.9 Changing Clusters 1224
4.9.1 Adding New Nodes 1224
4.9.2 Reducing Nodes 1226
4.9.3 Changing Node Specifications 1227
4.10 Managing Clusters 1230
4.10.1 Viewing Cluster Details 1230
4.10.2 Managing Cluster Tags 1233
4.10.3 Associating a Cluster with an Enterprise Project 1235
4.10.4 Forcibly Restarting a Cluster 1237
4.10.5 Deleting a Cluster 1239

5 Vector Database 1240

5.1 CSS Vector Engine vs. OpenSearch Vector Engine 1240
5.2 CSS Vector Database 1242
5.2.1 Procedure 1242
5.2.2 Preparing the Environment 1246
5.2.3 Creating a Vector Index 1250
5.2.4 Importing Vector Data 1264
5.2.5 Performing Vector Search 1267
5.2.6 Implementing Nested Vector Search 1279
5.2.7 Optimizing Query and Write Performance 1283
5.2.8 Managing the Vector Search Cache 1288
5.2.9 Client Code Examples 1291
5.2.9.1 Elasticsearch Vector Search Sample Code (Java) 1291
5.2.9.2 Elasticsearch Vector Search Sample Code (Python) 1294
5.2.9.3 OpenSearch Vector Search Sample Code (Java) 1296
5.2.9.4 OpenSearch Vector Search Sample Code (Python) 1303

6 CSS Resource Monitoring and Alarms 1306

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

1 Using IAM to Grant Access to CSS

1.1 Creating IAM Users and Granting Them Permissions to Use CSS

You can use Identity and Access Management (IAM) for fine-grained permissions control for CSS. With IAM, you can:

- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user has their own identity credentials for accessing CSS resources.
- Grant users only the permissions required to perform a given task based on their job responsibilities.
- Entrust an account or a cloud service to perform efficient O&M on your CSS resources.

If your account does not require individual IAM users, you may skip this section.

This section describes the procedure for granting permissions.

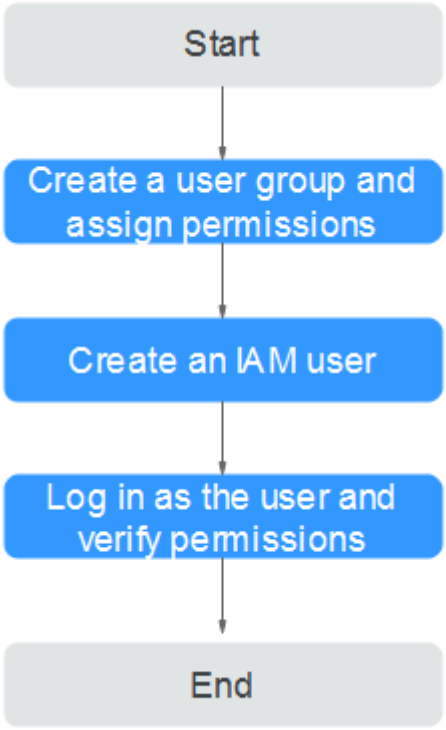
Prerequisites

You know what permissions to assign to which users. CSS supports administrator permissions and read-only permissions. For details, see Permissions Management.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Process Flow

Figure 1-1 Process of granting CSS permissions



1. Create and authorize a user group by following the instructions in the *Identity and Access Management User Guide*.

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

2. Create a user and add it to a user group by following the instructions in the *Identity and Access Management User Guide*.

   Create a user on the IAM console and add it to the user group created in **1**.

   By adding a user to a user group, you assign all the permissions that this user group has to that user. All users in a group have the same permissions.

3. Log in and verify permissions.

   Log in to the console as the created user, switch to the authorized region, and verify the permissions.

   - Choose **Service List** > **Cloud Search Service**. Then click **Create Cluster** on the CSS console. If the cluster cannot be purchased (assuming that the current permissions include only **CSS ReadOnlyAccess**), the **CSS ReadOnlyAccess** policy has already taken effect.
   - Choose any other service from **Service List**. (Assume that the current policy contains only **CSS ReadOnlyAccess**.) If a message appears indicating insufficient permissions to access the service, the **CSS ReadOnlyAccess** policy has already taken effect.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Examples of CSS Custom Policies

Custom policies can be created to supplement the system-defined policies of CSS. For the actions supported for custom policies, see section "Permissions Policies and Supported Actions" in the *Cloud Search Service API Reference*.

You can create custom policies on the cloud service platform in either of the following ways:

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.
- JSON: Create a JSON policy or edit an existing one.

For details about how to create custom policies, see section "Creating a Custom Policy" in the *Identity and Access Management User Guide*. The following provides examples of custom CSS policies.

NOTE

the IAM permissions and data plane cluster permissions of CSS are managed separately. To enhance data-plane security, you need to use the security mode.

To let an IAM user access an OBS bucket, you need to grant the **GetBucketStoragePolicy**, **GetBucketLocation**, **ListBucket**, and **ListAllMyBuckets** permissions to the user.

- Example 1: Allowing users to create CSS clusters

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

- Example 2: Denying cluster deletion

  A policy with only **Deny** permissions must be used in conjunction with other policies for it 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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

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. Example of a policy containing multiple actions:

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

1.2 Example: Granting IAM Users the Permission to Create CSS Clusters

To implement fine-grained permission management for CSS, you can use Identity and Access Management (IAM) to create independent IAM users and assign policies or roles to IAM user groups. The policies and roles can be used to control access to CSS resources.

This section describes how to create an IAM user and add the IAM user to a user group, so that the IAM user has the permission to create CSS clusters.

1. **Creating a User Group**: The staff of company A is divided into an admin group and a developer group. The default group **admin** is generated after company A's administrator registers an account. The administrator needs to create another group in IAM for the developer team.
2. **Assigning Permissions to a User Group**: Developers in the company need to use cloud services CSS and OBS, so the administrator needs to assign the required permissions to the developer group to enable access to these services. After permissions are assigned, users in the two groups can access the corresponding resources.
3. **Creating IAM Users**: Create IAM users for your staff and add them to different groups. The users have their own credentials and can perform

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

operations on cloud resources based on the permissions inherited from their respective groups.

4. **Logging In Using an IAM User and Verifying Permissions**: After using the account of company A (**A-Company**) to create users **James**, **Alice**, **Charlie**, **Jackson**, and **Emily**, the administrator needs to send the account name, IAM user names, and IAM users' initial passwords to the corresponding employees, who can then use their own usernames and passwords to access the cloud platform. Then, they can create clusters on the CSS console and by doing so verify their permissions.

Creating a User Group

Step 1 (Company A administrator) Log in to the Huawei Cloud management console.

Step 2 On the management console, hover the mouse pointer over the username in the upper right corner, and choose **Identity and Access Management** from the drop-down list.

Step 3 On the IAM console, choose **User Groups** and click **Create User Group**.

Step 4 In the displayed dialog box, enter the user group name in **Name**, for example, **developer**, and click **OK**.

----End

Assigning Permissions to a User Group

Step 1 In the user group list, click **Authorize** in the row that contains the developer user group.

Step 2 In the **Select Policy/Role** step, search for **CSS FullAccess** and **OBS Administrator** in the search box, select them, and click **Next**.

- Both the **CSS FullAccess** and **Elasticsearch Administrator** permissions enable users to create CSS clusters.
- If users in the group need to view resource usage, attach the **BSS Administrator** role to the group for the same project.

Step 3 Select a scope. Set **Scope** to **Region-specific projects**, and select the region where your resources are.

Step 4 Click **OK**.

----End

Creating IAM Users

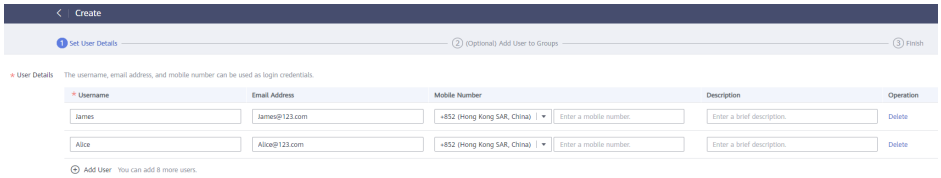
Step 1 Log in to the IAM console as the administrator and choose **Users** from the left navigation pane.

Step 2 On the **Users** page, click **Create User** in the upper right corner.

Step 3 Set basic user information. On the **Create User** page, configure **User Details** and **Access Type**. To create more users, click **Add User**. A maximum of 10 users can be created at a time.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 1-2 Configuring user information



NOTE

- Users can log in to Huawei Cloud using the username, email address, or mobile number.
- If users forget their passwords, they can reset them through email address or mobile number verification. If no email addresses or mobile numbers have been bound to users, users need to contact the administrator to reset their passwords.

Table 1-1 User information

Parameter	Description	Example Value
Username	(Mandatory) Username used by an IAM user to log in to Huawei Cloud.	James, Alice
Email Address	This parameter is mandatory if you select **Set by user** for **Credential Type**. Enter the email address of the IAM user to use as a login credential. IAM users can bind email addresses by themselves after being created.	Skip it.
Mobile Number	(Optional) Mobile number of the IAM user, which can be used as a login credential. IAM users can bind mobile numbers by themselves after being created.	Skip it.
Description	(Optional) Additional information about the IAM user.	Skip it.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 1-3 Specifying the access type

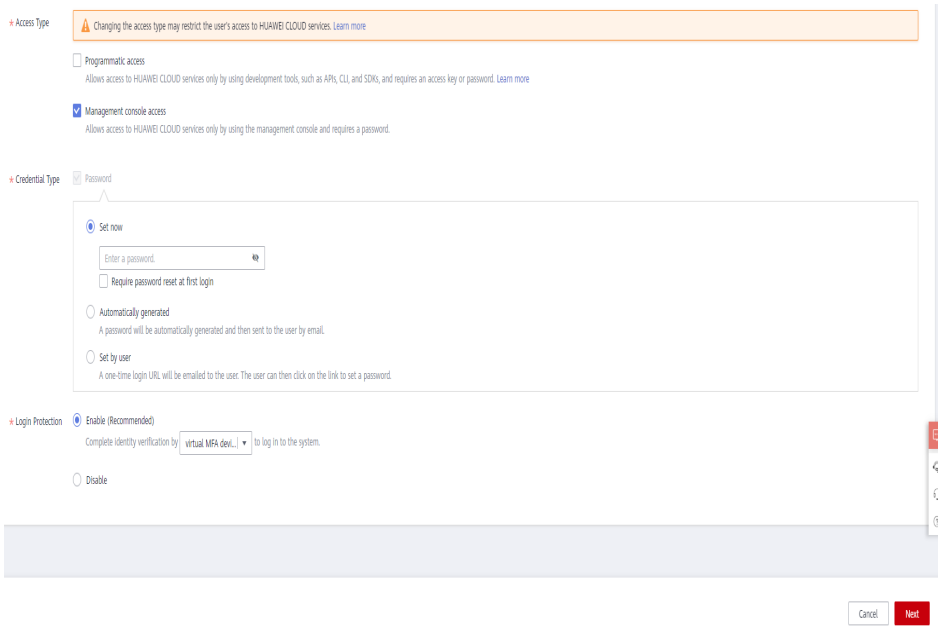


Table 1-2 Access types

Access Type	Description	Example Value
Programmatic access	An access key or password is generated for the IAM user. This type allows access to cloud services using development tools, such as APIs, CLI, and SDKs.	Select it.
Management console access	A password is generated for the IAM user. This type allows access to cloud services using the management console. If you select this option, **Password** must be selected for **Credential Type**.	Select it.

NOTE

- If the user accesses cloud services only by using the management console, select **Management console access** for **Access Type** and **Password** for **Credential Type**.
- If the user accesses cloud services only through programmatic calls, select **Programmatic access** for **Access Type** and **Access key** for **Credential Type**.
- If the user needs to use a password as the credential for programmatic access to certain APIs, select **Programmatic access** for **Access Type** and **Password** for **Credential Type**.
- If the user needs to perform access key verification when using certain services on the console, select **Programmatic access** and **Management console access** for **Access Type**, and select **Access key** and **Password** for **Credential Type**. For example, the user needs to perform access key verification when creating a data migration job in the Cloud Data Migration (CDM) console.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 1-3 Setting the credential type and login protection

Credential Type and Login Protection		Description	Example Value
Access Key		An access key comprises an access key ID (AK) and secret access key (SK), and is used as a long-term identity credential to **sign your requests for Huawei Cloud APIs**. After creating the user, you can download the **access key (AK/SK)** generated for the user.	Deselect it.
Password	Set now	Set a password for the user and determine whether to require the user to reset the password upon first login. If you plan to use the IAM user yourself, you are advised to select this option, set a password, and deselect **Require password reset at first login**.	Select it.
	Auto generate	The system automatically generates a login password for the user. After the user is created, download the EXCEL password file and provide the password to the user. The user can then use this password for login. **This option is available only when you create a single user.**	-
	Set by user	A one-time login URL will be emailed to the user. The user can click the link to log in to the console and set a password. If you will not be using the IAM user account, select this option and enter the email address and mobile number of the IAM user. The user can then set a password by clicking the one-time login URL sent over email. The login URL is valid for **two days**.	-
Login Protection	Enabled	After login protection is enabled, IAM users need to enter verification codes in addition to their usernames and passwords during console login. It is a security practice to enable login protection. You can verify your identity by SMS, email, or virtual MFA device during login.	-

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Credential Type and Login Protection		Description	Example Value
	Disabled	If login protection is disabled, you can enable it later by following the instructions provided in **Viewing or Modifying IAM User Information**.	Select it.

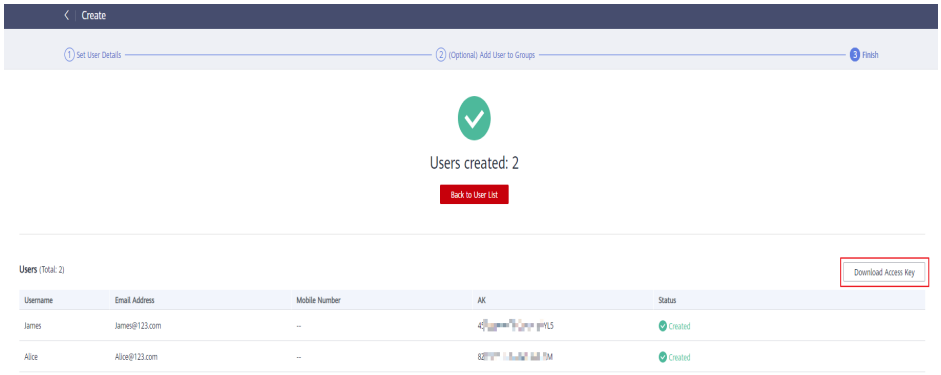
Step 4 Click **Next** and add the IAM users James and Alice to the created developer user group.

NOTE

- To create an administrator, add the user to the default group **admin**.
- You can add a user to multiple user groups.

Step 5 Click **Next**. The created IAM users James and Alice are displayed in the user list. If you have selected **Access key** for **Credential Type** in step 3, you can download the access key on the displayed page.

Figure 1-4 Users created successfully



----End

Logging In Using an IAM User and Verifying Permissions

Step 1 Click **IAM User** on the login page, and then enter your **Tenant name or Huawei Cloud account name**, **IAM username or email address**, and **IAM user password**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 1-5 Logging in as an IAM user

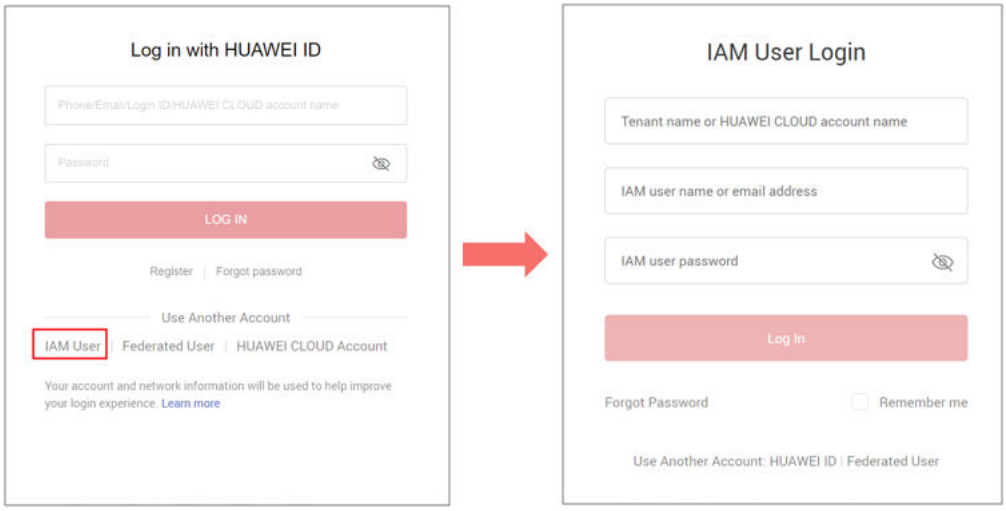


Table 1-4 Login parameters

Parameter	Example	Description
Tenant name or Huawei Cloud account name	**Company-A**	Account where the IAM user belongs, for example, **Company-A**.
IAM username or email address	**Alice**	IAM username or email address entered during user creation. You can obtain the IAM username and IAM user's initial password from the administrator.
IAM user password	********	Password of the IAM user, rather than the account. Enter the downloaded password.

Step 2 Click **Log In**.

Step 3 Choose **Service List**, search for CSS, and click the service name to go to the CSS console.

Step 4 In the upper right corner of the **Dashboard** page, click **Create Cluster**. Create a cluster by following the steps provided in . If the cluster can be created, the permissions have taken effect.

----End

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2 Elasticsearch

2.1 Procedure

This topic introduces the end-to-end procedure for using CSS Elasticsearch to build enterprise-grade search and analytics applications, covering key steps from environment setup to application rollout.

Figure 2-1 Procedure for using CSS Elasticsearch

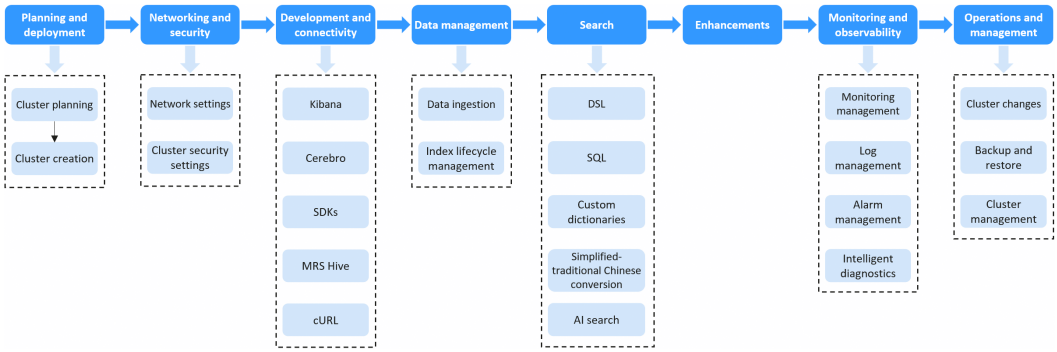


Table 2-1 End-to-end procedure for using Elasticsearch

Stage	Key Step	Description	Details
Planning and deployment	Cluster planning	Based on projected data volumes and intended use cases, plan node specifications, multiple-AZ deployment, and security mode in advance to minimize the need for changes after cluster creation.	**Planning Cluster AZs and HA** **Planning the Cluster Version and Security Mode** **Planning Node Types and Resource Specifications**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Stage	Key Step	Description	Details
	Creating a cluster	Create an Elasticsearch cluster based on your planning and service requirements.	**Creating a Cluster (New Version)**
Networking and security	Network settings	CSS Elasticsearch clusters support four networking modes. Choose the configuration that best suits your connectivity needs.	**Overview**
	Security settings	Configure user permissions and authentication policies to ensure cluster access security.	**Creating Cluster Users and Granting Permissions** **Configuring LDAP Authentication**
Development and connectivity	Connecting to your cluster	CSS provides multiple connection options tailored to different roles, programming languages, and network environments. At the application layer, CSS supports open-source RESTful APIs and SDKs for popular languages. It also includes built-in visualization and management tools such as Kibana and Cerebro.	**Cluster Connection Methods** **Connecting to a Cluster Through Kibana** **Connecting to a Cluster Through Cerebro**
Data management	Data ingestion	CSS Elasticsearch clusters provide multiple data ingestion options, including Logstash, open-source Elasticsearch APIs, and the Cloud Data Migration (CDM) service. Choose one that best suits your service requirements.	**Migrating Data Using CDM** **Ingesting Data Using Self-Managed Logstash** **Ingesting Data Using Open-Source Elasticsearch APIs**
	Index lifecycle management	CSS Elasticsearch clusters provides the Index State Management (ISM) plugin that you can use to configure and manage index lifecycle policies and automate index rollover. This helps reduce costs and improve efficiency.	**Configuring Index State Management (ISM) Policies**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Stage	Key Step	Description	Details
Search	Search and analytics	CSS Elasticsearch clusters support both DSL and SQL as query languages, along with an efficient tokenization plugin. It helps you build an intelligent search and big data analytics platform for efficient data mining and accurate insights.	**Using DSL for Search** **Using SQL for Search** **Configuring Custom Dictionaries** **Configuring Simplified-Traditional Chinese Conversion Search**
Enhancements	CSS enhancements over open-source Elasticsearch	CSS offers major enhancements over open-source Elasticsearch, delivering better performance, stability, and more cost-effective storage.	**CSS Enhancements Over Open-Source Elasticsearch**
Monitoring and observability	Monitoring management	Cloud Eye enables efficient visualization and monitoring of cluster resources and key SLA metrics (such as P99 latency and HTTP status codes), as well as index-level monitoring. The O&M team can monitor cluster health and performance bottlenecks in real time, ensuring high service availability.	**Using Cloud Eye to Monitor Clusters** **Monitoring P99 Latency and HTTP Status Codes** **Configuring Index-Level Monitoring** **Supported Metrics**
	Log management	The O&M team can analyze run logs, slow query logs, and operation audit records to quickly demarcate issues, analyze performance, and ensure security and compliance. Logs can be ingested into Elasticsearch in real time or backed up to OBS for long-term storage, meeting diverse, enterprise-grade operational and auditing requirements.	**Querying Logs** **Ingesting Logs into a Cluster** **Backing Up Logs to OBS** **Viewing Audit Logs**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Stage	Key Step	Description	Details
	Alarm management	CSS Elasticsearch clusters support custom alarm rules, flexible notification channels, and seamless integration with the native Prometheus monitoring ecosystem. Integration with SMN enables real-time push of cluster exceptions, shifting from passive response to proactive defense and ensuring service continuity.	**Configuring Open Distro Alert Notifications (via SMN)** **Synchronizing Alerts and Metrics to Prometheus**
	Intelligent diagnostics	The O&M team can easily perform periodic health checks on clusters, and comprehensive results enable prompt risk detection and handling, improving cluster reliability and availability.	**Intelligent Diagnostics**
Operations and management	Managing cluster changes	You can easily manage cluster changes throughout the entire lifecycle, such as cluster scaling, version upgrade, multi-AZ configuration, and HA configuration, all while ensuring service continuity.	**Scaling Out/Up** **Scaling In/Down** **Changing Node Specifications** **Replacing a Faulty Node** **Upgrading the Cluster Version** **Switching AZs** **Changing the Security Mode**
	Backup and restore	CSS Elasticsearch provides comprehensive backup and restore capabilities for clusters. Using snapshots, you can save point-in-time copies of cluster data in OBS buckets and quickly restore them in the event of data loss or other operational needs. This improves data reliability.	**Creating Snapshots to Back Up Data** **Restoring Data Using Snapshots** **Deleting a Cluster Snapshot**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Stage	Key Step	Description	Details
	Cluster management	Cluster resources can be isolated by enterprise project and tagged for refined management. You can easily restart and delete clusters and optimize them through YML parameters, ensuring efficient cluster management.	**Viewing Cluster Details** **Checking Default Cluster Plugins** **Modifying Cluster Parameters** **Managing Cluster Tags** **Associating a Cluster with an Enterprise Project** **Restarting a Cluster** **Deleting a Cluster**

2.2 Cluster Planning

2.2.1 Planning Cluster AZs and HA

This topic describes how to improve cluster availability by deploying a cluster across multiple AZs, including the rules for distributing cluster nodes among AZs, the recommended number of replicas, and policies for handling AZ failures.

An availability zone (AZ) is a physical region where resources use independent power supplies and networks. AZs in the same region can communicate with each other through internal networks but are physically isolated.

Multi-AZ deployment is a high availability feature provided by CSS. Deploying a cluster across two or three AZs located in the same region can help prevent data loss and lower the possibility of service outages.

Suggestions on Multi-AZ Deployment

If you select multi-AZ deployment when creating a cluster, CSS automatically enables cross-AZ HA to ensure that cluster nodes will be evenly distributed across the selected AZs (the difference between the number of nodes in each AZ cannot exceed 1).

You are advised to select three AZs, instead of two AZs, for multi-AZ deployment. If only two AZs are selected and one AZ becomes faulty, the cluster may not be able to elect a master node. As a result, the cluster may become unavailable.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Node Distribution Rules

Table 2-2, Table 2-3, and Table 2-4 show the distribution of data nodes, cold data nodes, master nodes, and client nodes across AZs, depending on the node and AZ counts. The node distribution rules for multi-AZ clusters are as follows:

- For a multi-AZ cluster, the system evenly distributes nodes across AZs. The difference in the number of same-type nodes across different AZs will not exceed 1.
- To ensure high service availability and prevent individual node overload in a multi-AZ deployment, the following node count requirements must be met:
  - **For a two-AZ cluster**, there must be at least four data nodes (or cold data nodes), two for each AZ; and there must be at least two client nodes (if configured), one for each AZ.
  - **For a three-AZ cluster**, each available node type must have at least one node in each AZ. This means each node type must have at least three nodes in total.
  - For node types that are not deployed (for example, no cold data nodes), the minimum node count requirement does not apply.
- If the number of data nodes or cold data nodes is not an integer multiple of the number of AZs, for example, five data nodes in a two-AZ cluster, uneven data distribution may occur, which can impact query and ingestion performance. For this reason, you should always ensure that the number of nodes is an integer multiple of the number of AZs.

Table 2-2 Number of data nodes/cold data nodes and AZ distribution

Total Nodes	Single AZ	Two AZs		Three AZs		
	AZ1	AZ1	AZ2	AZ1	AZ2	AZ3
1	1	Not supported		Not supported		
2	2	Not supported		Not supported		
3	3	Not supported		1	1	1
4	4	2	2	2	1	1
5	5	3	2	2	2	1
6	6	3	3	2	2	2
7	7	4	3	3	2	2
...	...	...	...	...	...	...

Table 2-3 Number of master nodes and AZ distribution

Total Nodes	Single AZ	Two AZs		Three AZs		
	AZ1	AZ1	AZ2	AZ1	AZ2	AZ3

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3	3	2	1	1	1	1
5	5	3	2	2	2	1
7	7	4	3	3	2	2
9	9	5	4	3	3	3

Table 2-4 Number of client nodes and AZ distribution

Total Nodes	Single 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
...	...	...	...	...	...	...

Replica Configuration Suggestions

For a multi-AZ cluster, configure the number of index replicas in a manner that can better capitalize on the high availability that comes with such as deployment.

- With a dual-AZ deployment, if one AZ becomes unavailable, the other AZ continues to provide services. In this case, configure at least one replica. If higher query performance is desired, you can increase the number of replicas.
- In the case of a three-AZ deployment, if one AZ becomes unavailable, the other AZs can continue to provide services. In this case, also configure at least one replica. To enhance the cluster's query performance, increase the number of replicas.

For Elasticsearch and OpenSearch clusters, the default number of index replicas is 1. To enable more replicas, modify relevant settings. The following are two examples.

- Adjust the number of replicas for an existing index:

  curl -XPUT http://ip:9200/{index_name}/_settings -d '{"number_of_replicas":2}'
- Set the number of replicas for new indexes using a template:

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

where, **ip** indicates the private IP address of the cluster; **index_name** indicates the index name; **templatename** indicates the template name; **template** indicates the index name matching rule (meaning the template will automatically apply to indexes that match this rule. The asterisk (*) indicates that the template will apply

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

to all new indexes); and **number_of_replicas** indicates the number of index replicas to change to. In this example, the number of index replicas is changed to 2.

Service Outage Pattern Analysis

Master nodes manage cluster-wide operations, including metadata, indexes, and shard allocation. In a cluster with master nodes, the master nodes perform these tasks. In a cluster without them, data nodes and cold data nodes share the responsibilities of the master nodes.

In the case of a single AZ failure, if fewer than half (including half) of the nodes assuming master node responsibilities are still available, service availability will be affected. In this case, you need to restore services by referring to **Table 2-5**.

For example, a cluster has three master nodes distributed in two AZs. If the AZ that contains one master node becomes unavailable while the other AZ remains available, services will still be available. However, if it is the other AZ, which contains two of the three master nodes, that fails, services will be interrupted, because fewer than half of the master nodes (or nodes that assume master node responsibilities) are available. In this case, restore services by referring to **Table 2-5**.

Table 2-5 Possible service outage patterns in the face of the failure of a single AZ

Number of AZs	Number of Master Nodes	Service Outage Patterns and Handling Suggestions
2	0 (In this case, data nodes and cold data nodes share the responsibilities of master nodes.)	• If the number of data nodes plus cold data nodes is an even number: If half of the nodes become unavailable due to the failure of a single AZ, you must add one node to the still available AZ to ensure a master node can be elected for the cluster. To add nodes to your cluster, contact technical support. • If the number of data nodes plus cold data nodes is an odd number: – If the failed AZ is the one that has one more node, you must add two nodes to the surviving AZ to ensure a master node can be elected for the cluster. To add nodes to your cluster, contact technical support. – If the failed AZ is the other one—the one that has one less node, services will not be interrupted and a master node can be elected.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Number of AZs	Number of Master Nodes	Service Outage Patterns and Handling Suggestions
2	3	There is a 50% chance of service interruption when one AZ fails. In this situation, one AZ has two master nodes, and the other has one: • If the AZ that has one master node fails while the other AZ remains available, two master nodes are still available, so a master node can be elected from among them. In this case, services remain available. • If the failed AZ is the other one, only one master node survives, in which case, no master node can be elected, and services will be interrupted. In this case, contact technical support.
3	0 (In this case, data nodes and cold data nodes share the responsibilities of master nodes.)	There is a small chance of service interruption when a single node fails. However, there is one high-risk scenario: If the cluster spans three AZs and the total number of data nodes plus cold data nodes is 4, so the node distribution across the three AZs is 2-1-1, the failure of the AZ containing two nodes will cause a service interruption. To avoid this, avoid selecting four nodes for a three-AZ cluster.
3	3	A single AZ failure will not result in service interruptions.

2.2.2 Planning the Cluster Version and Security Mode

This topic introduces different Elasticsearch cluster versions, including features supported by each version and its application scenarios. It also describes the characteristics and application scenarios of different cluster security modes (including different application-layer protocols).

Introduction to Different Cluster Versions

When selecting an Elasticsearch cluster version, consider factors such as service requirements, available features, performance, security updates, and long-term support, ensuring that the selected version can meet both current and future needs and provide a stable, secure environment for your data.

- If you are deploying CSS Elasticsearch clusters for the first time, you are advised to use the latest version.
- If you are migrating an in-house built or third-party Elasticsearch cluster to CSS without altering the cluster, keep the version of the source cluster.
- If you are migrating a self-managed or third-party Elasticsearch cluster to CSS while recoding it, choose Elasticsearch 7.10.2.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-6 CSS cluster version enhancements

Category	Feature	Elasticsearch 7.6.2	Elasticsearch 7.10.2	Related Documents
Vector search	Open-source vector search	x	x	-
	CSS vector search	√	√	**CSS Vector Database**
Cost optimization	Decoupled storage and compute	√	√	**Configuring Decoupled Storage and Compute**
	Switching between hot and cold storage tiers	√	√	**Switching Between Hot and Cold Data**
Enhanced stability	Read/write splitting	√	√	**Configuring Read/Write Splitting Between Leader and Follower**
	Flow Control 2.0	√	√	**Configuring Flow Control 2.0**
	Flow Control 1.0	√	√	**Configuring Flow Control 1.0**
	Query traffic isolation	x	√	**Configuring Query Traffic Isolation**
	Large query isolation	√	√	**Configuring Large Query Isolation**
	Kibana query response circuit breaker	x	√	**Configuring the Kibana Query Response Circuit Breaker**
	Automatic retry for failed shard allocation	x	√	**Configuring Automatic Retry for Failed Shard Allocation**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Category	Feature	Elasticsearch 7.6.2	Elasticsearch 7.10.2	Related Documents
Enhanced performance	Enhanced ingestion performance	x	√	Enhancing Data Ingestion Performance
	Enhanced aggregation performance	x	√	Configuring Enhanced Aggregation
Enhanced operations	Index recycle bin	x	√	Enabling the Index Recycle Bin
	Query resource tracker	x	√	Tracking Query Resource Consumption
	CSS-enhanced monitoring	√	√	Monitoring P99 Latency and HTTP Status Codes
	Index-level monitoring	√	√	Configuring Index-Level Monitoring

Cluster Security Modes

Configure the security mode based on the security needs of your cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-7 Cluster security modes

Cluster Type	Key Settings	Description	Applicable Scenario
Non-security mode cluster	**Security Mode**: disabled	Access to such a cluster requires no user authentication, and data will be transmitted in plaintext using HTTP.	Use when creating a cluster for internal testing or workloads that have a low security standard. • Advantage: easy to access the cluster. • Disadvantage: poor security, as anyone can access the cluster. When security mode is disabled, public network access and Kibana public network access cannot be enabled. To ensure data security, deploy the cluster in a secure environment. Do not expose the cluster's network interface to the public network.
Security-mode cluster	Security-mode cluster + HTTP: • **Security Mode**: enabled • **HTTPS Access**: disabled	Such a cluster requires user authentication. It supports access control and data encryption, and it uses HTTP to transmit data in plaintext.	Use to balance security and performance. • Advantage: User authentication improves cluster security. HTTP-based access ensures high performance of the cluster. User permissions can be configured to ensure proper isolation. • Disadvantage: Public network access is not supported. To ensure data security, deploy the cluster in a secure environment. Do not expose the cluster's network interface to the public network.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Cluster Type	Key Settings	Description	Applicable Scenario
	Security-mode cluster + HTTPS: • **Security Mode**: enabled • **HTTPS Access**: enabled	Such a cluster requires user authentication. It supports access control and data encryption, and it uses HTTPS to encrypt communication and enhance data security.	Use when security takes precedence over performance and public network access is required. • Advantage: User authentication improves cluster security. HTTPS enhances cluster security by encrypting all communication over the public network. User permissions can be configured to ensure proper isolation. • Disadvantage: When HTTPS is used, data encryption and decryption introduce computational overhead and impact the cluster's read and write performance.

To access a security-mode cluster, a username and password need to be provided. CSS supports authentication for the following two types of users:

- Administrator: The default administrator username is **admin**, and the password is the one specified during cluster creation.
- Cluster user: created by the cluster administrator on Kibana. User permissions can be configured to ensure proper isolation. For details, see **Creating Cluster Users and Granting Permissions**.

2.2.3 Planning Node Types and Resource Specifications

This topic describes common application scenarios and provides configuration recommendations for each type of node in a cluster, including data nodes, master nodes, client nodes, and cold data nodes. The goal is to help you select and configure the appropriate nodes based on your specific service requirements. It also provides suggestions on selecting node specifications and configuring the node storage type, storage capacity, and node quantity, helping you properly plan the capacities of your cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Key Points for Node Configuration

Table 2-8 Key points for node configuration

Configuration Item	Data Nodes	Master Nodes	Client Nodes	Cold Data Nodes
Mandatory or Not	Mandatory	Optional	Optional	Optional
Primary role	Stores data. Can also serve as a master, ingest, or analytics node.	Manages the cluster, including metadata and shard allocation.	Receives and coordinates external requests, such as search and indexing requests.	Stores infrequently accessed historical data.
Node specifications	Select node specifications based on your service requirements (such as data volume, performance requirements, and budget). For details, see **Node Specifications**.	Only EVS disks are supported. Select higher node specifications for large clusters.	Only EVS disks are supported. Select higher node specifications for heavy read/write workloads.	Select node specifications based on your service requirements (such as data volume, performance requirements, and budget). For details, see **Node Specifications**.
Storage capacity	• EVS disk: Configure as needed (must be a multiple of 20 GB). For details, see **Recommended Node Storage Capacity**. • Local disk: Fixed capacity.	Fixed at 40 GB.	Fixed at 40 GB.	• EVS disk: Configure as needed (must be a multiple of 20 GB). For details, see **Recommended Node Storage Capacity**. • Local disk: Fixed capacity.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuration Item	Data Nodes	Master Nodes	Client Nodes	Cold Data Nodes
Number of nodes	• With dedicated master nodes: 1–200; without dedicated master nodes: 1–32. • A minimum of 3 is recommended. For details, see **Recommended Node Quantity**.	3, 5, 7, or 9. For details, see **Recommended Master Node Quantity**.	1–64. For details, see **Recommended Client Node Quantity**.	1–32. For details, see **Recommended Node Quantity**.

Node Types

Before creating a cluster, determine the types of nodes to use based on service requirements, query load, data growth patterns, and performance goals.

For each cluster, data nodes are mandatory, while master, client, and cold data nodes are optional. Choose whether to enable the latter three based on your service needs and performance requirements.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-9 Introduction to different node types

Node Type	When to Use
Data node (ESS)	Data nodes are primarily used to store the cluster's data. The functions of the data nodes vary depending on the node types enabled for the cluster. ● If **Master node** and **Client node** are both selected, data nodes will be used for data storage only. ● If **Master node** and **Client node** are both unselected, data nodes will be used for all of the following purposes: cluster management, data storage, cluster access, and data analysis. To ensure reliability, a cluster should have at least three nodes. ● If **Master node** is selected but **Client node** is not, data nodes will be used for data storage, cluster access, and data analysis. ● If **Master node** is unselected but **Client node** is selected, data nodes will be used for data storage and cluster management.
Master node (ess-master)	Master nodes manage cluster-wide operations, including metadata, indexes, and shard allocation. For large-scale deployments, using dedicated master nodes enhances cluster stability, service availability, and centralized control. ● Large-scale cluster: For a cluster that has more than 10 nodes, you are advised to add dedicated master nodes to effectively manage the cluster status and metadata. ● Large quantities of indexes and index shards: If the number of indexes or shards exceeds 10,000, master nodes will have better performance handling complex cluster management tasks, avoiding impact on the performance of data nodes. ● Better management of cluster nodes: Master nodes maintain the cluster metadata, including index mapping, settings, and aliases. For a complex cluster structure, dedicated master nodes offer better management, including node joining, exiting, and fault detection. Master nodes play a critical role in cluster node management. ● Improved cluster stability and reliability: Dedicated master nodes improve cluster stability and reliability by taking over cluster management responsibilities from data storage and query nodes. ● Optimized performance for data nodes: By offloading cluster management tasks from data nodes to master nodes, you can allow data nodes to focus on data processing, which leads to improved performance.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Node Type	When to Use
Client node (ess-client)	Client nodes route and coordinate search and index requests, offloading processing from data nodes for enhanced query performance and cluster scalability when there are heavy loads. • High queries per second (QPS): In the face of a high QPS, a dedicated client node can evenly distribute query requests, reducing the load of data nodes and improving the overall query performance. • Complex aggregation queries: For complex, compute-intensive aggregation queries, a client node can dedicate to the handling of aggregation results, thus improving the efficiency and response speed of such queries. • Large number of shards: In a cluster with a large number of shards, a client node can effectively coordinate and manage query requests to each shard, improving efficiency in request forwarding and processing. • Reducing the load of data nodes: A client node parses search requests, determines the locations of index shards, and coordinates different nodes to execute searches. This reduces the load of data nodes by allowing them to focus on data storage and indexing. • Improved cluster scalability: The use of client nodes allows for better cluster scalability and flexibility, enabling support for large datasets and more complex query requirements.
Cold data node (ess-cold)	Cold data nodes are used to store and query latency-insensitive data in large quantities. They offer an effective way to manage large datasets while cutting storage costs. • Storage of historical data in large quantities: Cold data nodes offer a more cost-effective solution for storing large quantities of historical data that are infrequently accessed but useful for analytical purposes. • Optimizing hot data performance: By migrating cold data to cold data nodes, you reduce the storage load of hot data nodes, thereby optimizing their query and write performance. • Insensitivity to query latency: Cold data nodes are a better option for storing data that is insensitive to a high query latency. • Cost-effectiveness: Cold data nodes usually use large-capacity disks that offer inexpensive storage.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

NOTE

- If no master, client, or cold data nodes were enabled when a cluster was created, you can add them if data nodes become overloaded later at some point. For details, see **Adding New Node Types**.
- When cold data nodes are enabled, you can switch between cold and hot data storage. For details, see **Switching Between Hot and Cold Data**. If there are no cold data nodes, we recommend that you use decoupled storage and compute, which can also cut storage costs. For details, see **Configuring Decoupled Storage and Compute**.

Suggestions on Data Node Configuration

Table 2-10 Data node configuration

Parameter	Configuration Suggestions
Node Specifications	In the node flavor list, **vCPUs | Memory** indicate the number of vCPUs and memory capacity available for each flavor, and **Recommended Storage** indicates the supported storage capacity range. We recommend that you select node specifications based on service needs, such as the data volumes, performance requirements, and your spending budget. **Node Specifications** describes the application scenarios and core features of different node specifications. It can help you properly plan your cluster. For more information about different node specifications, see **ECS Types**.
Node Storage Type and Capacity	• If the selected node flavor uses EVS disks, you need to further select **Node Storage Type** and **Capacity** based on service requirements. – For more on EVS disk performance, see **Disk Types and Performance**. – The value range of node storage capacity is determined by the node flavor you select. The value must be divisible by 20. For how to calculate the required node storage capacity, see **Recommended Node Storage Capacity**. – Node storage capacity cannot be reduced once the cluster is created. Evaluate your long-term data needs and select an appropriate size.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Configuration Suggestions
Nodes	• If master nodes are configured, the number of data nodes ranges from 1 to 200. If the combined number of data nodes and cold data nodes exceeds 100, you are advised to select a flavor with at least 16 vCPUs. This helps prevent delays in cluster status synchronization across nodes and ensures optimal read and write performance. • If no master nodes are configured, the number of data nodes ranges from 1 to 32. • To ensure cluster availability, you should configure at least three data nodes. For how to calculate the number of nodes needed, see **Recommended Node Quantity**. If the number of data nodes in your cluster is not evenly divisible by the number of AZs, data distribution may become unbalanced across nodes. This will negatively impact both query and write performance.

Node Specifications

The choice of node specifications affects disk deployment. There are two types of disks: local disks and EVS disks. For example, high-performance nodes may use local disks, while standard nodes may use cost-effective EVS disks.

- EVS disks are a virtual block storage service that is independent of ECSs. They provide high reliability and fast elasticity, making them ideal for workloads that require high data reliability and highly scalable storage capacity.

Recommended Node Storage Capacity

If you select EVS disks, you can select a node storage capacity within the range supported by the selected node flavor.

When planning the storage capacity of a cluster, consider the following factors: the original data size, number of data replicas, data bloat rate, and disk usage. The following is a recommended formula for determining the needed cluster storage capacity:

Storage capacity = Original data size x (1 + Number of replicas) x (1 + Data bloat rate) x (1 + Ratio of reserved space)

where,

- Original data size: Determine the size of original data that needs to be stored.
- Number of replicas: The default value is 1.
- Data bloat rate: Extra data may be generated due to data indexing. You are advised to use a 25% data bloat rate.
- Disk usage: Considering the space occupied by the operating system and file system and the space reserved for optimized disk performance and

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

redundancy, you are advised to keep the disk usage under 70%. That is, you need to reserve 30% of the total disk capacity.

Using the recommended values above, the formula should be: Storage capacity = Original data size x 2 x 1.25 x 1.3 = Original data size x 3.25

This formula is for quick reference only. You still need to adjust it based on the actual applications and projected data growth rate. Moreover, the storage capacity must be divisible by 20.

Recommended Node Quantity

Data nodes and cold data nodes are used to store data. Their quantities largely determine the cluster performance baseline. The following formulas can be used to estimate the number of data nodes and cold data nodes needed:

Number of data nodes + Number of cold data nodes = Number of write nodes + Number of query nodes

Number of write nodes = Traffic during peak hours (MB/s) / Number of vCPUs per node / Per-vCPU write performance baseline (MB/s) x Number of index replicas

Number of query nodes = Query QPS / (Number of vCPUs per node x 3 / 2 / Average query latency (s)) x Number of index shards

where,

- The number of vCPUs per node is determined by the selected node flavor.
- The per-vCPU write performance baseline is determined by the selected disk type. For a node that uses EVS disks, the per-vCPU write performance baseline is 1 MB/s. For a node that uses local disks, this baseline is 1.5 MB/s.
- QPS indicates the number of queries that can be processed per second.
- The average query latency is used as the query performance baseline.
- Where a high write performance is desired, we recommend setting the number of index shards to equal the total number of data nodes plus cold data nodes, thus evenly distributing the write load across all nodes.
- Where a high QPS is desired, we recommend setting the number of index shards to equal the total number of data nodes plus cold data nodes **minus 1**, thus evenly distributing the query load across all nodes.

Example:

If the peak traffic is 100 MB/s; the node specifications are 16 vCPUs and 64 GB RAM, with EVS disks; the query throughput is 1,000 QPS; the average query latency is 100 ms; and each index has three shards and two replicas, then:

Number of write nodes = 100/16/1 x 2 = 12

Number of query nodes = 1000/(16 x 3/2/0.1) x 3 = 12

Number of data nodes + Number of cold data nodes = Number of write nodes + Number of query nodes = 12 + 12 = 24

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Suggestions on Master Node Configuration

Table 2-11 Master node configuration

Parameter	Configuration Suggestions
Node Specifications	In the node flavor list, **vCPUs | Memory** indicate the number of vCPUs and memory capacity available for each flavor, and **Recommended Storage** indicates the supported storage capacity range. For a large-scale cluster with a large number of index shards, use large-capacity nodes as master nodes. Master nodes support EVS disks only. For more information about different node specifications, see **ECS Types**.
Node Storage Type and Capacity	Select an appropriate storage type and capacity for master nodes. • For more on EVS disk performance, see **Disk Types and Performance**. • The node storage capacity is fixed at 40 GB.
Nodes	The number of master nodes can be 3, 5, 7, or 9. For split-brain protection, the number of master nodes must be an odd number. For the recommended master node quantity, see **Recommended Master Node Quantity**.

Recommended Master Node Quantity

Increasing the number of master nodes can improve the cluster's fault tolerance performance, but it can also lead to issues like increased communication overhead, longer election time, and low resource utilization. You are advised to select an appropriate number of master nodes based on service requirements.

The following table provides the recommended number of master nodes based on the cluster size.

Table 2-12 Recommended number of master nodes

Cluster Size	Recommended Number of Master Nodes	Performance Expectation
Small (< 10 nodes)	3	Basic high availability
Medium (50 > number of nodes ≥ 10)	3 or 5	Balanced performance and high availability
Large (≤ 50 nodes)	5 or 7	Enhanced fault tolerance and stability

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Cluster Size	Recommended Number of Master Nodes	Performance Expectation
Extra-large (≥ 100 nodes)	5 or 7	Pay attention to excessive communication overhead.

Suggestions on Client Node Configuration

Table 2-13 Client node configuration

Parameter	Configuration Suggestions
Node Specifications	In the node flavor list, **vCPUs | Memory** indicate the number of vCPUs and memory capacity available for each flavor, and **Recommended Storage** indicates the supported storage capacity range. For a cluster with heavy read and write traffic, use large-capacity nodes as client nodes. Client nodes support EVS disks only. For more information about different node specifications, see **ECS Types**.
Node Storage Type and Capacity	Select an appropriate storage type and capacity for client nodes. • For more on EVS disk performance, see **Disk Types and Performance**. • The node storage capacity is fixed at 40 GB.
Nodes	The number of client nodes ranges from 1 to 64. For the recommended client node quantity, see **Recommended Client Node Quantity**.

Recommended Client Node Quantity

Client nodes route and coordinate search and index requests. Its quantity should be determined based on the cluster traffic. For your reference, the following table provides the recommended number of client nodes based on the number of data nodes.

Table 2-14 Recommended number of client nodes

Number of Data Nodes	Recommended Number of Client Nodes
< 5	Do not use client nodes.
≥ 5 and < 10	2

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Number of Data Nodes	Recommended Number of Client Nodes
≥ 10	Number of client nodes/Number of data nodes = 1:5

Suggestions on Cold Data Node Configuration

Table 2-15 Cold data node configuration

Parameter	Configuration Suggestions
Node Specifications	In the node flavor list, **vCPUs | Memory** indicate the number of vCPUs and memory capacity available for each flavor, and **Recommended Storage** indicates the supported storage capacity range. We recommend that you select node specifications based on service needs, such as the data volumes, performance requirements, and your spending budget. **Node Specifications** describes the application scenarios and core features of different node specifications. It can help you properly plan your cluster. For more information about different node specifications, see **ECS Types**.
Node Storage Type and Capacity	• If the selected node flavor uses EVS disks, you need to further select **Node Storage Type** and **Capacity** based on service requirements. – For more on EVS disk performance, see **Disk Types and Performance**. – The value range of node storage capacity is determined by the node flavor you select. The value must be divisible by 20. For how to calculate the required node storage capacity, see **Recommended Node Storage Capacity**. – Node storage capacity cannot be reduced once the cluster is created. Evaluate your long-term data needs and select an appropriate size. • If the selected node flavor uses local disks, there is no need to select the node storage type, and the node storage capacity is a fixed value. Both of them are determined by the local disk specifications.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Configuration Suggestions
Nodes	The value range for cold data nodes is 1 to 32. For how to calculate the number of nodes needed, see **Recommended Node Quantity**. • If the combined number of data nodes and cold data nodes exceeds 100, you are advised to select a flavor with at least 16 vCPUs. This helps prevent delays in cluster status synchronization across nodes and ensures optimal read and write performance. • If the number of cold data nodes in your cluster is not evenly divisible by the number of AZs, data distribution may become unbalanced across nodes. This will negatively impact both query and write performance.

2.3 Creating a Cluster (New Version)

You can use Elasticsearch clusters created in CSS for purposes like log analytics, data search, real-time data analytics, and more.

You can create an Elasticsearch cluster in either of the following ways:

- Method 1: Create an Elasticsearch cluster on the CSS management console. This topic provides the operation guide for the default UI (that is, the new UI). For the operation guide for the old UI, see **Creating a Cluster**.

  NOTE

  To improve efficiency in creating clusters, CSS has made several enhancements on the cluster creation page in terms of ease-of-use. Being more intuitive, the new UI streamlines cluster creation procedures.

- Method 2: Create an Elasticsearch cluster using CSS APIs. For details, see **Creating a Cluster**.

Prerequisites

- You have planned the Elasticsearch cluster configuration by following the guidelines in **Cluster Planning**, including the cluster size, node configuration, and storage capacity requirements.
- You have created a VPC, subnet, and security group beforehand, or you can create them when creating the cluster. For more information, see **Virtual Private Cloud User Guide**.
- If you are going to enable automatic snapshot creation for the cluster, make sure you have created an OBS bucket for storing snapshots. The OBS bucket must be in the same region as the Elasticsearch clusters, and the storage class must be **Standard** or **Warm**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

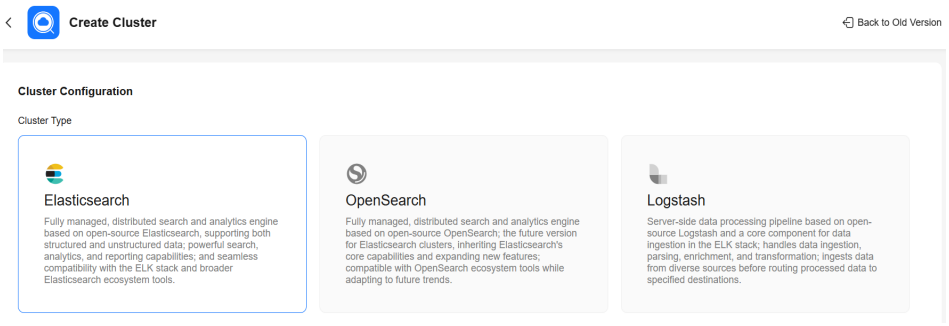
Impact on Billing

- If you enable automatic snapshot creation, automatically created snapshots will be stored in standard OBS buckets, and you will need to pay extra for the storage. For more information, see **Billing Overview**.
- VPC endpoints, if created for the cluster, will incur extra fees, depending on the resource usage. For details, see **VPC Endpoint Billing**.

Opening the Create Cluster Page

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the upper right corner, click **Create Cluster**. The new-version UI for creating a cluster is displayed by default.

Figure 2-2 Create Cluster (new version)



Cluster Configuration

Select the cluster type and version.

Table 2-16 Cluster configuration

Parameter	Description
Cluster Type	Select **Elasticsearch**.
Cluster Version	Select a cluster version from the drop-down list.

Basic Settings

Select the region and AZ of the cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-17 Basic settings

Parameter	Description
Region	Select the region where the cluster is located. A region is the location of a physical data center. Regions are defined based on their geographical location and network latency. For lower network latency and quicker resource access, select the nearest region.
AZ	Select one or more AZs associated with the cluster region. An AZ is a physical region where resources use independent power supplies and networks. AZs are physically isolated but interconnected through an internal network. A maximum of three AZs can be configured. For details about the use of multiple AZs, see **Suggestions on Multi-AZ Deployment**.

Data Nodes

Configure the cluster's data nodes.

Data nodes are primarily used to store the cluster's data. The functions of the data nodes vary depending on the node types enabled for the cluster.

- If the cluster has no client nodes, the data nodes will need to handle the additional task of query request distribution and data analysis.
- If the cluster has no master nodes, the data nodes will also need to handle cluster management.

Table 2-18 Configuring data nodes

Parameter	Description
CPU Architecture	Select the CPU architecture of the data nodes. **x86** nodes are supported. The architectures actually supported may vary depending on the regional environment.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Node Specifications	Select the specifications of the data nodes. Click **Available**. On the displayed page, select a flavor that suits your needs. The nodes may use either local or EVS disks. For more information about different node specifications, see **ECS Types**. In the node flavor list, **vCPUs | Memory** indicates the number of vCPUs and memory capacity available for each flavor, and **Recommended Storage** indicates the supported storage capacity range. The node flavors available may vary depending on the region you select. **WARNING** • Local disks are less reliable than EVS disks—if the underlying host fails, data stored on local disks may be permanently lost. For this reason, avoid using local disks to store mission-critical data. Instead, use EVS disks. • If you do use local disks, you are advised to enable automatic snapshot creation when creating the cluster, and also enable replicas when creating indexes. This helps lower the risk of data loss.
Node Storage Type and Capacity	Select the storage type and capacity of the data nodes. • If the selected node flavor uses EVS disks, you need to further select **Node Storage Type** and **Capacity** based on service requirements. – Available EVS disk types vary depending on your region. For more on EVS disk performance, see **Disk Types and Performance**. – The value range of node storage capacity is determined by the node flavor you select. The value must be divisible by 20. – Node storage capacity cannot be reduced once the cluster is created. Evaluate your long-term data needs and select an appropriate size. • If the selected node flavor uses local disks, there is no need to select the node storage type, and the node storage capacity is a fixed value. Both of them are determined by the local disk specifications.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Nodes	Set the number of data nodes for the cluster. • If master nodes are configured, the number of data nodes ranges from 1 to 200. • If no master nodes are configured, the number of data nodes ranges from 1 to 32. Configuration principles: • To ensure high service availability and prevent individual node overload, a single-AZ cluster **should** have at least three data nodes; a two-AZ cluster **must** have at least four data nodes (two in each AZ); and a three-AZ cluster **must** have at least three data nodes (one in each AZ). • For optimal performance, the number of data nodes should be an integer multiple of the number of AZs, for example, six data nodes for a three-AZ cluster. This helps prevent uneven data distribution and ensure optimal query and ingestion performance.

Master Nodes

Configure master nodes for the cluster. By default, master nodes are not enabled. To enable master nodes, select **Master Node**, and configure the node settings.

Master nodes manage cluster-wide operations, including metadata, indexes, and shard allocation. For large-scale deployments, using dedicated master nodes enhances cluster stability, service availability, and centralized control.

Table 2-19 Configuring master nodes

Parameter	Description
CPU Architecture	Select the CPU architecture of the master nodes. **x86** nodes are supported. The architectures actually supported may vary depending on the regional environment.
Node Specifications	Select the specifications of the master nodes. Click **Available**. On the displayed page, select a flavor that suits your needs. Master nodes support EVS disks only. For more information about different node specifications, see **ECS Types**. In the node flavor list, **vCPUs | Memory** indicate the number of vCPUs and memory capacity available for each flavor. The node flavors available may vary depending on the region you select.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Node Storage Type and Capacity	Select the storage type of the master nodes. • Available EVS disk types vary depending on your region. For more on EVS disk performance, see **Disk Types and Performance**. • The node storage capacity is fixed at 40 GB.
Nodes	Set the number of master nodes for the cluster. The value can be **3**, **5**, **7**, or **9**.

Client Nodes

Configure client nodes for the cluster. By default, client nodes are not enabled. To enable client nodes, select **Client Node**, and configure the node settings.

Client nodes route and coordinate search and index requests, offloading processing from data nodes for enhanced query performance and cluster scalability when there are heavy loads.

Table 2-20 Configuring client nodes

Parameter	Description
CPU Architecture	Select the CPU architecture of the client nodes. **x86** nodes are supported. The architectures actually supported may vary depending on the regional environment.
Node Specifications	Select the specifications of the client nodes. Click **Available**. On the displayed page, select a flavor that suits your needs. Client nodes support EVS disks only. For more information about different node specifications, see **ECS Types**. In the node flavor list, **vCPUs | Memory** indicate the number of vCPUs and memory capacity available for each flavor. The node flavors available may vary depending on the region you select.
Node Storage Type and Capacity	Select the storage type of the client nodes. • Available EVS disk types vary depending on your region. For more on EVS disk performance, see **Disk Types and Performance**. • The node storage capacity is fixed at 40 GB.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Nodes	Set the number of client nodes for the cluster. Value range: 1 to 64 Configuration principle: To ensure high service availability and prevent individual node overload in a multi-AZ deployment, the number of client nodes must be greater than or equal to the number of AZs. (For a cluster spanning two AZs, this means a minimum of two client nodes; for a three-AZ deployment, a minimum of three client nodes.)

Cold Data Nodes

Configure cold data nodes for the cluster. By default, cold data nodes are not enabled. To enable cold data nodes, select **Cold Data Node**, and configure the node settings.

Cold data nodes are used to store and query large quantities of latency-insensitive data. They offer an effective way to manage large datasets while cutting storage costs. When cold data nodes are enabled, you can switch between cold and hot data storage. For details, see **Switching Between Hot and Cold Data**. If there are no cold data nodes, we recommend that you use decoupled storage and compute, which can also cut storage costs. For details, see **Configuring Decoupled Storage and Compute**.

Table 2-21 Configuring cold data nodes

Parameter	Description
CPU Architecture	Select the CPU architecture of the cold data nodes. **x86** nodes are supported. The architectures actually supported may vary depending on the regional environment.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Node Specifications	Select the specifications of the cold data nodes. Click **Available**. On the displayed page, select a flavor that suits your needs. The nodes may use either local or EVS disks. For more information about different node specifications, see **ECS Types**. In the node flavor list, **vCPUs | Memory** indicates the number of vCPUs and memory capacity available for each flavor, and **Recommended Storage** indicates the supported storage capacity range. The node flavors available may vary depending on the region you select. **WARNING** • Local disks are less reliable than EVS disks—if the underlying host fails, data stored on local disks may be permanently lost. For this reason, avoid using local disks to store mission-critical data. Instead, use EVS disks. • If you do use local disks, you are advised to enable automatic snapshot creation when creating the cluster, and also enable replicas when creating indexes. This helps lower the risk of data loss.
Node Storage Type and Capacity	Select the storage type and capacity of the cold data nodes. • If the selected node flavor uses EVS disks, you need to further select **Node Storage Type** and **Capacity** based on service requirements. – Available EVS disk types vary depending on your region. For more on EVS disk performance, see **Disk Types and Performance**. – The value range of node storage capacity is determined by the node flavor you select. The value must be divisible by 20. – Node storage capacity cannot be reduced once the cluster is created. Evaluate your long-term data needs and select an appropriate size. • If the selected node flavor uses local disks, there is no need to select the node storage type, and the node storage capacity is a fixed value. Both of them are determined by the local disk specifications.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Nodes	Set the number of cold data nodes for the cluster. Value range: 1 to 32 Configuration principles: • To ensure high service availability and prevent individual node overload, a two-AZ cluster **must** have at least four cold data nodes (two in each AZ); and a three-AZ cluster **must** have at least three cold data nodes (one in each AZ). • To prevent uneven data distribution and ensure optimal query and ingestion performance, the number of cold data nodes should be an integer multiple of the number of AZs, for example, six cold data nodes for a three-AZ cluster.

Network Settings

Configure the VPC, subnet, IP addresses, and security group of the cluster.

Table 2-22 Network settings

Parameter	Description
VPC	Select a VPC for the cluster for proper network isolation. If none of the existing VPCs meets your requirements, click **Create VPC** to create one. (Make sure you have the permission to create VPCs.) For details, see **Creating a VPC and Subnet**.
Subnet	Select a subnet for the cluster. A subnet improves network security by providing exclusive network resources that are isolated from other networks. Select a subnet in the current VPC. If no existing subnets meet your requirements, click **Create Subnet** to create a new subnet. For details, see **Creating a VPC and Subnet**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
IPv4 Address	Assign IPv4 addresses to cluster nodes. • **Assign automatically**: Private IPv4 addresses will be automatically assigned to cluster nodes. • **Assign manually**: Manually assign private IPv4 addresses to cluster nodes. Before assigning an IP address, click **View In-Use IP Address** to check IP addresses that are already in use. – If you manually specify fewer IP addresses than the total number of cluster nodes, the system will automatically assign random private IPv4 addresses to the remaining nodes. – Manually specified IP addresses will be assigned in the following priority order: client nodes > data nodes > cold data nodes > master nodes.
Security Group	Select a security group for the cluster. A security group serves as a virtual firewall that provides access control policies for clusters. To select a security group that meets your requirements, click **View Security Group** to go to the security group list, where you can check the details of each security group. The selected security group must allow port 9200 in the inbound direction. Otherwise, the cluster may be inaccessible to external services. Furthermore, to enable Kibana private network access, port 5601 must also be allowed. If none of the existing security groups meets your requirements, click **Create Security Group** to create one. (Make sure you have the permission to create security groups.) For details, see **Creating a Security Group**.

Security Mode

Configure the cluster's security mode and access methods. The security mode is enabled by default when you create a cluster. To disable it, deselect **Security Mode**.

Configure the security mode based on the security needs of your cluster.

- **Disabling the security mode**: Use when creating a cluster for internal testing or workloads that have a low security standard. When the security mode is disabled, HTTP is used, data is transmitted in plaintext, and no user authentication is required for cluster access. Make sure the cluster is deployed in a secure environment. Do not expose the cluster's network interface to the public network.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- **Security mode + HTTP**: Use to balance security and performance. In this setting, user authentication is required for cluster access, yet data is transmitted via HTTP.
- **Security mode + HTTPS**: Use when security takes precedence over performance. In this setting, user authentication is required for cluster access, and data is transmitted via HTTPS in ciphertext.

CAUTION

HTTPS, public network access, and Kibana public network access can be enabled only when the security mode is enabled.

Table 2-23 Configuring the security mode

Parameter	Description
Administrator account and password	When the security mode is enabled, cluster access requires user authentication. In this case, the administrator password must be configured. • **Administrator**: The default value is **admin** and cannot be changed. • **Administrator Password**: Set and confirm the administrator password. This password will be required when you access this cluster.
HTTPS Access	Whether to enable HTTPS access for the cluster. Enabled by default, HTTPS improves security by encrypting all communication with the cluster. To use HTTP instead, deselect **HTTPS Access**. Data security cannot be guaranteed when HTTPS is disabled. Exercise caution. When HTTPS is used, data encryption and decryption introduce computational overhead, which may impact the cluster's read and write performance.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Public Network Access	Whether to enable public network access. By default, it is disabled. To enable it, select **Public Network Access**, and configure the necessary settings. Public network access can be enabled only if the security mode and HTTPS are both enabled for the cluster. When public network access is enabled, the cluster is automatically assigned a public IP address with dedicated dynamic BGP bandwidth. When **Public Network Access** is selected, further configure the following parameters: • **Bandwidth**: Set the bandwidth for public network access. The value ranges from 1 Mbit/s to 200 Mbit/s. • **Configure Whitelist** Control public network access to the cluster using a whitelist. – If a whitelist is configured, only IP addresses that are on this whitelist can access the cluster over the public network. Click **+Add**. In the displayed text box, enter IP addresses or CIDR blocks that are allowed to access the cluster from the public network. Separate them using commas (,). Each value must be unique. An example of valid values: **192.168.1.1,10.0.0.0/24**. Examples of invalid values: **0.0.0.0**, **xx.xx.xx.xx/0**, **172.16.0.0-172.16.255.255**, non-standard formats (e.g., **192.168.1**), and duplicate values. – If no whitelist is configured, all public IP addresses can access the cluster. However, this can be a security risk and should be avoided.

Cluster Management

Configure the cluster name, enterprise project, and more.

Table 2-24 Cluster management information

Parameter	Description
Cluster Name	User-defined cluster name. The cluster name must start with a letter and can contain 4 to 32 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Add Description	Add a description for the cluster for easy recognition. Click **+ Add Description** and then enter a description of the cluster. The value can contain 0 to 128 characters.
Enterprise Project	Associate the cluster with an enterprise project. An enterprise project groups cloud resources, so you can manage resources and members by project. The default project is **default**. If enterprise projects are enabled, you can select an enterprise project from the drop-down list. You can click **View Enterprise Projects** to go to the Enterprise Project Management Service console and check enterprise project details. For more information, see **Accessing the Enterprise Center**.
Tags	Adding tags to clusters helps you identify and manage your cluster resources. Click **+ Add**. You can select existing tag keys and values from the drop-down lists or enter new ones. If you want to use the same tag to identify multiple cloud resources for better resource grouping, we recommend that you predefine tags in Tag Management Service (TMS). For details, see **Predefined Tags**. Each cluster can have a maximum of 20 tags. If your organization has configured tag policies for CSS, add tags to clusters based on these policies. Tags that do not comply with predefined tag policies will cause a cluster creation failure. Contact the administrator to learn about the tag policies.

Automatic Snapshot Creation

Expand **More Settings** and configure automatic snapshot creation. Automatic snapshot creation is disabled by default. To enable it, select **Automatic Snapshot Creation**, and configure the necessary settings. You are advised to enable automatic snapshot creation to have data backed up periodically, thus improving cluster reliability. If you choose not to enable it, enable replicas when creating indexes, which helps prevent data loss.

CAUTION

If you enable automatic snapshot creation, automatically created snapshots will be stored in standard OBS buckets, and you will need to pay extra for the storage. For more information, see **Billing Overview**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-25 Configuring automatic snapshot creation

Parameter	Description
OBS Bucket	From the drop-down list, select an OBS bucket for storing snapshots. If no OBS buckets meet your requirements, click **Create Bucket** to go to the OBS console and create one. For details, see **Creating a Bucket**. An IAM user must be granted the following OBS permissions to view available OBS buckets: obs:bucket:getBucketStoragePolicy obs:bucket:listAllMyBuckets
Backup Path	Snapshot storage path in the OBS bucket. The backup path cannot: • Contain the following characters: **\:*?"<>|'{}** • Start with a slash (/). • Start or end with a period (.). • Contain more than two consecutive slashes (/) or periods (.). • Exceed 512 characters. **CAUTION** Only standard OBS storage can be used to store snapshots. Do not apply OBS lifecycle rules to the backup path.
Maximum Backup Rate (per Second)	The parameter sets the maximum backup rate per node. When it is exceeded, flow control is triggered to prevent excessive resource usage and ensure system stability. **Value format**: number (0–9999) + unit (KB, MB, GB, TB, PB, or B) **Default value**: 40 MB The value 0MB means there is no limit on how fast data is backed up to snapshots. An overly high backup rate may lead to excessive resource usage, which may impact cluster stability. Therefore, set this parameter carefully. The actual backup rate may not reach the configured value, as it depends on factors such as OBS performance and disk I/O.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description

Maximum Recovery Rate (per Second)	The parameter sets the maximum recovery rate per node. When it is exceeded, flow control is triggered to prevent excessive resource usage and ensure system stability.
	Value format: number (0–9999) + unit (KB, MB, GB, TB, PB, or B)
	Default value: 40 MB for version 7.6.2 or earlier; and 0 MB (no limit) for versions later than 7.6.2.
	An overly high recovery rate may lead to excessive resource usage, which may impact cluster stability. Configure this parameter carefully to maintain optimal performance.
	For versions later than 7.6.2, the actual recovery rate is also affected by the value of **indices.recovery.max_bytes_per_sec**. Whichever is smaller between the two parameters will be used. Additionally, the actual recovery rate may not reach the configured value, as it depends on factors such as OBS performance and disk I/O.
	NOTE
	• To check the value of **indices.recovery.max_bytes_per_sec**, run the following command:
	`GET _cluster/settings`
	• To modify **indices.recovery.max_bytes_per_sec**, run the following command:
	`PUT _cluster/settings`
	`{`
	`  "transient": {`
	`  "indices.recovery.max_bytes_per_sec": "100mb"`
	`  }`
	`}`

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
IAM Agency	Select an IAM agency to grant the current account the permission to access and use OBS. To store snapshots to an OBS bucket, you must have the required OBS access permissions. • If you are configuring an agency for the first time, click **Automatically Create IAM Agency** to create **css-obs-agency**. • If there is an IAM agency automatically created earlier, you can click **One-click authorization** to have the OBS Administrator permissions deleted automatically, and have the following custom policies added automatically instead to implement more refined permissions control. "obs:bucket:GetBucketLocation", "obs:object:GetObjectVersion", "obs:object:GetObject", "obs:object:DeleteObject", "obs:bucket:HeadBucket", "obs:bucket:GetBucketStoragePolicy", "obs:object:DeleteObjectVersion", "obs:bucket:ListBucketVersions", "obs:bucket:ListBucket", "obs:object:PutObject" • When OBS buckets use SSE-KMS encryption, the IAM agency must be granted KMS permissions. You can click **Automatically Create IAM Agency** and **One-click authorization** to have the following custom policies created automatically. "kms:cmk:create", "kms:dek:create", "kms:cmk:get", "kms:dek:decrypt", "kms:cmk:list" • To use **Automatically Create IAM Agency** and **One-click authorization**, the following minimum permissions are required: "iam:agencies:listAgencies", "iam:roles:listRoles", "iam:agencies:getAgency", "iam:agencies:createAgency", "iam:permissions:listRolesForAgency", "iam:permissions:grantRoleToAgency", "iam:permissions:listRolesForAgencyOnProject", "iam:permissions:revokeRoleFromAgency", "iam:roles:createRole" • To use an IAM agency, the following minimum permissions are required: "iam:agencies:listAgencies", "iam:agencies:getAgency", "iam:permissions:listRolesForAgencyOnProject", "iam:permissions:listRolesForAgency" **WARNING** The agency name can contain only letters (case-sensitive), digits, underscores (_), and hyphens (-). Otherwise, the backup will fail.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Snapshot Name Prefix	A snapshot name consists of the snapshot name prefix and timestamp, for example, *snapshot-1566921603720*. The snapshot name prefix can contain 1 to 32 characters. It must start with a lowercase letter, and it can only contain lowercase letters, digits, hyphens (-), and underscores (_). The default value is **snapshot**.
Time Zone	Select a time zone for the backup start time.
Backup Start Time	Specify the start time of auto backup. Select a value from the drop-down list. The interval can be **Daily**, **Hourly**, or weekly (by selecting a specific day of the week), and the backup time can be set to any hour from 00:00 to 23:00 (full hours only).
Retained Snapshots	Number of automatically created snapshots to be retained. The value ranges from 1 to 90. The system automatically deletes excess snapshots every half hour. (The expiration deletion policy applies only to the snapshots that were automatically created at the same frequency as the current automatic snapshot creation policy.) **NOTE** If the snapshot creation interval is short or if the data size of indexes is large, the number of automatic snapshots retained may not reach the value set using this parameter.

VPC Endpoint

Expand **More Settings** and configure a VPC endpoint for the cluster. VPC endpoint is disabled by default. To enable it, select **VPC Endpoint**, and configure the necessary settings.

VPC Endpoint enables you to access resources across VPCs using a dedicated gateway without exposing the network information of servers. The address of a VPC endpoint consists of an IPv4 address and private domain name. When you access a cluster through the IP address or private domain name of a VPC endpoint, you are not subject to the cluster's security group rules, and you can configure a whitelist for access control.

CAUTION

- If a shared VPC and a subnet within this shared VPC were selected earlier for the cluster, **VPC Endpoint** cannot be enabled.
- VPC endpoints, if created for the cluster, will incur extra fees, depending on the resource usage. For details, see **VPC Endpoint Billing**.
- You need specific permissions to create VPC endpoints. For details, see **VPCEP Permissions**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-26 Configuring VPC Endpoint

Parameter	Description
Create Private Domain Name	Whether to create a private domain name for the VPC endpoint. • Enable: The system automatically assigns a private domain name to the VPC endpoint. After cluster creation, you can check this private domain name on the **VPC Endpoint** tab of the cluster details page. • Disable: No private domain name will be configured for the VPC endpoint. The cluster can only be accessed through an IP address assigned to the VPC endpoint.
Configure Whitelist	Add accounts that are allowed to access the cluster through the VPC endpoint service. Click **Add** to add accounts in **Account ID**. • To obtain your authorized account ID, point to your username in the upper right corner, and choose **My Credentials**. Copy the value of **Account ID**. • If no account is added or the account ID is set to *, all users are allowed to access the cluster.

Kibana Public Network Access

Expand **More Settings** and configure Kibana public network access for the cluster. Kibana public network access is disabled by default. To enable it, select **Kibana Public Network Access**, and configure the necessary settings.

When Kibana public network access is enabled, a public IP address will be automatically assigned for Kibana access. Users will be able to access this Elasticsearch cluster's Kibana console from the public network through this address.

CAUTION

This parameter is available only when the security mode is enabled for the cluster.

Table 2-27 Configuring public network access for Kibana

Parameter	Description
Bandwidth	Bandwidth for accessing Kibana from the public network Value range: 1 Mbit/s to 200 Mbit/s

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Configure Whitelist	Control Kibana public network access using a whitelist. • If a whitelist is configured, only IP addresses that are on this whitelist can access the cluster's Kibana console over the public network. Click **+ Add**. In the displayed text box, enter IP addresses or CIDR blocks that are allowed to access the cluster's Kibana console from the public network. Separate them using commas (,). Each value must be unique. An example of valid values: **192.168.1.1,10.0.0.0/24**. Examples of invalid values: **0.0.0.0**, **xx.xx.xx.xx/0**, **172.16.0.0-172.16.255.255**, non-standard formats (e.g., **192.168.1**), and duplicate values. • If no whitelist is configured, all public IP addresses can access the cluster's Kibana console. However, this can be a security risk and should be avoided.

Submit

1. In the **Configuration Summary** panel, review the cluster configuration details.

   Any mandatory fields that are missing are displayed in red, and you have to set them in the configuration area.
2. Click **Create Now**.

   Cluster creation time depends on the number of nodes. Clusters with a large number of nodes typically take longer to create.
3. Return to the cluster list and check the newly created cluster.

   If the cluster is created successfully, **Cluster Status** changes to **Available**.

2.4 Creating a Cluster

This topic describes how to create an Elasticsearch cluster. You can create an Elasticsearch cluster in either of the following ways:

- Method 1: Create an Elasticsearch cluster on the CSS management console.
- Method 2: Create an Elasticsearch cluster using CSS APIs. For details, see **Creating a Cluster**.

Scenarios

Table 2-28 lists key parameters that differentiate between different types of clusters.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-28 Parameters that differentiate between different types of clusters

Cluster Type	Security Mode	HTTPS Access	Internet Access	Kibana Public Network Access
Cluster in non-security mode	Disabled	N/A	Cannot be enabled	Cannot be enabled
Cluster in security mode + HTTP	Enabled	Disabled	Cannot be enabled	Can be enabled
Cluster in security mode + HTTPS	Enabled	Enabled	Can be enabled	Can be enabled

Prerequisites

You have planned the Elasticsearch cluster configuration by following the instructions in **Cluster Planning**.

Creating a Cluster

1. Log in to the **CSS management console**.
2. On the **Dashboard** page, click **Create Cluster** in the upper right corner. The **Create Cluster** page is displayed.

   Alternatively, choose **Clusters** > **Elasticsearch** in the navigation tree on the left. Click **Create Cluster** in the upper right corner. The **Create Cluster** page is displayed.
3. On the **Basic Configuration** page, configure basic information and resources for the Elasticsearch cluster.

   Table 2-29 Basic configuration of the Elasticsearch cluster

Parameter	Description
Region	Select the region where the cluster is located. Resources in different regions cannot communicate with each other over an internal network. For lower network latency and quicker resource access, select the nearest region.
AZ	Select one or more AZs associated with the cluster region. A maximum of three AZs can be configured. For details about the use of multiple AZs, see **Suggestions on Multi-AZ Deployment**.
Type	Select **Elasticsearch**.
Version	Select a cluster version from the drop-down list.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Name	User-defined cluster name. The cluster name must start with a letter and can contain 4 to 32 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
Cluster Description	Add a description for the cluster for easy recognition. The value can contain 0 to 128 characters.
Nodes	Number of data nodes in the cluster. Data nodes store the cluster's data. If there are no client nodes in the cluster, data nodes will need to additionally handle cluster access and data analytics requests; if there are no master nodes, data nodes will also need to provide cluster management. • If master nodes are configured, the number of data nodes ranges from 1 to 200. • If no master nodes are configured, the number of data nodes ranges from 1 to 32. Principles: • To ensure high service availability and prevent individual node overload, a single-AZ cluster **should** have at least three data nodes; a two-AZ cluster **must** have at least four data nodes (two in each AZ); and a three-AZ cluster **must** have at least three data nodes (one in each AZ). • For optimal performance, the number of data nodes should be an integer multiple of the number of AZs, for example, six data nodes for a three-AZ cluster. This helps prevent uneven data distribution and ensure optimal query and ingestion performance.
CPU Architecture	**x86**. The supported types depend on the actual regional environment.
Node Specifications	Data node flavor. You can select a node flavor based on your needs. Each cluster supports only one specification. For details, see **ECS Types**.
Node Storage Type	If you select EVS for node storage, you need to further select the EVS disk type for data nodes of the cluster. Supported node storage types: Common I/O, High I/O, Ultra-high I/O. **NOTE** If the type of storage in use is not supported, the storage type is not displayed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Node Storage Capacity	Data node storage capacity. Its value range varies with the node specifications. The node storage capacity must be a multiple of 20.
Master node	Master nodes manage cluster-wide operations, including metadata, indexes, and shard allocation. For large-scale deployments, using dedicated master nodes enhances cluster stability, service availability, and centralized control. After enabling master nodes, specify **Node Specifications**, **Nodes**, and **Node Storage Type**. The value of **Nodes** must be an odd number greater than or equal to 3. Up to nine nodes are supported. The value of **Node Storage Capacity** is fixed. You can select a storage type based on your needs.
Client node	Client nodes route and coordinate search and index requests, offloading processing from data nodes for enhanced query performance and cluster scalability when there are heavy loads. After enabling client nodes, specify **Node Specifications**, **Nodes** and **Node Storage Type**. The value of **Nodes** ranges from 1 to 64. The value of **Node Storage Capacity** is fixed. You can select a storage type based on your needs. To ensure high service availability and prevent individual node overload in a multi-AZ deployment, the number of client nodes should be greater than or equal to the number of AZs. (For a cluster spanning two AZs, this means a minimum of two client nodes; for a three-AZ deployment, a minimum of three client nodes.)

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Cold data node	Cold data nodes are used to store and query large quantities of latency-insensitive data. They offer an effective way to manage large datasets while cutting storage costs. After enabling cold data nodes, configure **Node Specifications**, **Nodes**, **Node Storage Type**, and **Node Storage Capacity**. The value of **Nodes** ranges from 1 to 32. Select **Node Storage Type** and **Node Storage Capacity** as required. • To ensure high service availability and prevent individual node overload, a two-AZ cluster **must** have at least four cold data nodes (two in each AZ); and a three-AZ cluster **must** have at least three cold data nodes (one in each AZ). • To prevent uneven data distribution and ensure optimal query and ingestion performance, the number of cold data nodes should be an integer multiple of the number of AZs, for example, six cold data nodes for a three-AZ cluster. When cold data nodes are enabled, you can switch between cold and hot data nodes. For details, see **Switching Between Hot and Cold Data**.
Enterprise Project	When creating a CSS cluster, you can bind an enterprise project to the cluster if you have enabled the enterprise project function. Select an enterprise project from the **Enterprise Project** drop-down list, or click **View Enterprise Project** to go to the **Enterprise Project Management Service** page and check existing enterprise projects.

4. Click **Next: Network**.
5. On the **Network** page, configure the network settings and security mode for the Elasticsearch cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-30 Network settings for the Elasticsearch cluster

Parameter	Description
VPC	Specify a VPC to isolate the cluster's network. Click **View VPC** to go to the VPC management console and check the created VPCs. If no VPC is available, contact the CSS administrator to create a new VPC. For details, see **Creating a VPC with a Subnet**. **NOTE** The 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, improving network security. Select a subnet needed by the cluster in the current VPC. If no existing subnets meet your requirements, click **Create Subnet** to create a new subnet. For details, see **Creating a VPC and Subnet**.
Security Group	A security group serves as a virtual firewall that provides access control policies for clusters. Select a security group for the cluster. Click **View Security Group** to go to the security group list, where you can view details about security groups. **NOTE** • If your cluster version is 7.6.2 or later, ensure that all the ports used for communication between nodes in the same security group are allowed. If such settings cannot be configured, ensure at least the access to port 9300 is allowed. • After the port 9300 is enabled, if the cluster disk usage is high, delete expired data to release the disk storage space. For details, see **How Do I Clear Elasticsearch Indexes in CSS?**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Security Mode	Whether to enable the security mode for the cluster. • The security mode is enabled by default. In security mode, a cluster's communication is encrypted and access to the cluster requires user authentication. This is why **Administrator Username** and **Administrator Password** must be configured for the cluster. – The default administrator username is **admin**. – Set and confirm the **Administrator Password**. This password will be required when you access this cluster. • If **Security Mode** is disabled, a cluster in non-security mode will be created. With such a cluster, access to the cluster will require no user authentication, and data will be transmitted in plaintext using HTTP. Make sure the cluster is deployed in a secure environment. Do not expose the cluster's network interface to the public network.
HTTPS Access	HTTPS access can be enabled only when security mode is enabled for the cluster. With HTTPS access enabled, communication will be encrypted when you access the cluster. **NOTE** Compared with a non-security mode cluster that uses HTTP, a security-mode cluster that uses HTTPS has lower read performance. The performance loss is estimated at around 20% under high concurrency. If you want fast read performance as well as the isolation and permission control (such as indexes, documents, and fields) enabled by the security mode, you can disable **HTTPS Access**. After **HTTPS Access** is disabled, HTTP protocol is used for cluster communication. In this case, data security cannot be guaranteed and public network access is disabled.

6. Click **Next: Advanced Settings**.
7. On the **Advanced Settings** page, configure a snapshot policy and other advanced settings for the Elasticsearch cluster.
   a. Set a cluster snapshot policy.

      Cluster snapshots are disabled by default. To enable them, toggle on **Cluster Snapshots**. To store automatically created snapshots in OBS, an agency will need to be created in order to access OBS. Fees will be incurred for using OBS storage (storage class: Standard).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-31 Basic configuration for a cluster snapshot policy

Parameter	Description
OBS Bucket	From the drop-down list, select an OBS bucket for storing snapshots. You can also click **Create Bucket** on the right to create an OBS bucket. For details, see **Creating a Bucket**. The created or existing OBS bucket must meet the following requirements: • **Storage Class**: **Standard**. • **Region** must be the same as that of the created cluster.
Backup Path	Snapshot storage path in the OBS bucket. The backup path cannot: • Contain the following characters: \:*?"<>|'{} • Start with a slash (/). • Start or end with a period (.). • Contain more than two consecutive slashes (/) or periods (.). • Exceed 512 characters. **CAUTION** Only standard OBS storage can be used to store snapshots. Do not apply OBS lifecycle rules to the backup path.
Maximum Backup Rate (per Second)	The parameter sets the maximum backup rate per node. When it is exceeded, flow control is triggered to prevent excessive resource usage and ensure system stability. **Value format**: number (0–9999) + unit (KB, MB, GB, TB, PB, or B) **Default value**: 40 MB The value 0MB means there is no limit on how fast data is backed up to snapshots. An overly high backup rate may lead to excessive resource usage, which may impact cluster stability. Therefore, set this parameter carefully. The actual backup rate may not reach the configured value, as it depends on factors such as OBS performance and disk I/O.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description

Maximum Recovery Rate (per Second)	The parameter sets the maximum recovery rate per node. When it is exceeded, flow control is triggered to prevent excessive resource usage and ensure system stability.
	Value format: number (0–9999) + unit (KB, MB, GB, TB, PB, or B)
	Default value: 40 MB for version 7.6.2 or earlier; and 0 MB (no limit) for versions later than 7.6.2.
	An overly high recovery rate may lead to excessive resource usage, which may impact cluster stability. Configure this parameter carefully to maintain optimal performance.
	For versions later than 7.6.2, the actual recovery rate is also affected by the value of **indices.recovery.max_bytes_per_sec**. Whichever is smaller between the two parameters will be used. Additionally, the actual recovery rate may not reach the configured value, as it depends on factors such as OBS performance and disk I/O.
	NOTE
	• To check the value of **indices.recovery.max_bytes_per_sec**, run the following command:
	`GET _cluster/settings`
	• To modify **indices.recovery.max_bytes_per_sec**, run the following command:
	`PUT _cluster/settings`
	`{`
	`  "transient": {`
	`  "indices.recovery.max_bytes_per_sec": "100mb"`
	`  }`
	`}`

IAM Agency	IAM agency authorized by the current account for CSS to access or maintain data stored in OBS. If no agency is available, contact the CSS administrator to create one. For details, see **Creating an Agency**. The selected IAM agency must meet the following requirements: • **Agency Type** must be **Cloud service**. • Set **Cloud Service** to **Elasticsearch** or **CSS**. • Mandatory policies: **Tenant Administrator** or **OBS Administrator**. **WARNING** The agency name can contain only letters (case-sensitive), digits, underscores (_), and hyphens (-). Otherwise, the backup will fail.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-32 Setting Automatic Snapshot Creation

Parameter	Description
Snapshot Name Prefix	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. A snapshot name consists of a snapshot name prefix and a timestamp, for example, **snapshot-1566921603720**.
Time Zone	Time zone for the backup time. Specify **Backup Started Time** based on the time zone.
Backup Start Time	Specify the start time of auto backup. Select a value from the drop-down list. The interval can be **Daily**, **Hourly**, or weekly (by selecting a specific day of the week), and the backup time can be set to any hour from 00:00 to 23:00 (full hours only).

b. Configure advanced settings for the cluster. Select **Default** or **Custom**.

- **Default**: **VPC Endpoint Service**, **Kibana Public Access**, and **Tags** are disabled by default. You can manually enable these settings after the cluster is created.
- **Custom**: You can enable **VPC Endpoint**, **Kibana Public Access**, and **Tags** as required.

VPC Endpoint

VPC Endpoint enables you to access resources across Virtual Private Clouds (VPCs) using a dedicated gateway, without exposing the network information of servers. When VPC Endpoint is enabled, a VPC endpoint will be created by default. You can select **Create Private Domain Name** if necessary. Afterwards, others will be able to access this cluster across VPCs through a VPC endpoint IP address or a private domain name. Accessing a cluster through a VPC endpoint IP address or private domain name bypasses the cluster's security group rules. In this case, you can configure a whitelist for access control.

NOTE

- If a shared VPC and a subnet within this shared VPC were selected earlier for the cluster on the **Network** page, **VPC Endpoint** cannot be enabled for the cluster.
- After the VPC endpoint service is enabled for a cluster, you will be billed per use for the service. For more information, see **VPC Endpoint Billing**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-33 Configuring VPC Endpoint

Parameter	Description
Create Private Domain Name	If **Create Private Domain Name** is selected, the system generates a node IP address and also automatically creates a private domain name, which enables users to access this cluster from within the same VPC. If it is not selected, only a node IP address is generated.
VPC Endpoint Whitelist	In **VPC Endpoint Whitelist**, you can add accounts that are allowed to access the cluster using a node IP address or private domain name. • Click **Add** to add accounts in **Account ID**. If no account is added or the account ID is set to *, all users are allowed to access the cluster. • Click **Delete** in the **Operation** column to delete accounts. **NOTE** To obtain your authorized account ID, point to your username in the upper right corner, and choose **My Credentials**. Copy the value of **Account ID**.

Kibana Public Network Access

This parameter is available only when security mode is enabled for the cluster. By enabling this option, you can obtain a public IP address for accessing Kibana.

Table 2-34 Configuring public network access for Kibana

Parameter	Description
Bandwidth	Bandwidth for accessing Kibana through a public IP address. Value range: 1 to 100. Unit: Mbit/s
Access Control	If you disable this function, all IP addresses can access Kibana through the public IP address. If you enable this function, only IP addresses or IP address ranges that are on the whitelist can access Kibana from the public network.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Whitelist	IP addresses or IP address ranges that are allowed to access the cluster from the public network. Use commas (,) to separate different values. This parameter can be configured only when **Access Control** is enabled. You are advised to enable the whitelist. **NOTE** The whitelist that controls Kibana public network access depends on whitelist support by the ELB service. After you update the whitelist, the new settings take effect immediately for new connections. For existing persistent connections using the IP addresses that have been removed from the whitelist, the new settings take effect in approximately 1 minute after these connections are disconnected.

Tags

Adding tags to clusters helps you identify and manage your cluster resources. You can configure your own tags or use tags predefined by the Tag Management Service (TMS). If you want to use the same tag to identify multiple cloud resources for better resource grouping, we recommend that you predefine tags in Tag Management Service (TMS). For details, see **Predefined Tags Overview**.

NOTE

If your organization has configured tag policies for CSS, add cluster tags based on these policies. Tags that do not comply with predefined tag policies will cause a cluster creation failure. Contact the administrator to learn about the tag policies.

8. Click **Next: Confirm Configuration**. Check the configuration and click **Next** to create a cluster.
9. Return to the cluster list and check the newly created cluster. If the cluster is created successfully, **Cluster Status** changes to **Available**. Cluster creation time depends on the number of nodes. Clusters with a large number of nodes typically take longer to create.

   If cluster creation fails, try creating the cluster again by rectifying the errors returned.

2.5 Network Settings

2.5.1 Overview

This topic introduces the four networking modes supported by CSS Elasticsearch clusters. Choose the configuration that best suits your connectivity needs, for example, whether you require intra-VPC, cross-VPC, or public network access.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Comparing Different Networking Modes

Table 2-35 Comparing different networking modes

Dimension	Private Network Access	Public Network Access	Dedicated Load Balancer	VPC Endpoint
Intra-VPC access	Supported	Supported	Supported	Supported
Cross-VPC access	VPC peering connections and cluster routes required	Supported (automatic traversal)	VPC peering connections and cluster routes required	Supported (automatic traversal)
Internet accessibility	No	Yes	Depends on whether a public IP address is allocated to the load balancer.	No
Security	VPC isolation	Public network access control policy	ELB access control policy	VPCEP access control policy
Data transfer costs	N/A	Public network bandwidth fees	ELB resource fees	VPCEP resource fees
Advantage	Low latency, low cost, and high security	All-domain Internet accessibility	Powerful load balancing, two-way authentication via custom certificates	Private connections, network topology decoupled
When to Use	Internal service systems on the cloud	Ad hoc testing, public network services	High availability + load balancing	Cross-account/ Cross-VPC services
Configuration Guide	(Optional) **Configuring Routes for a Cluster**	**Configuring Public Network Access**	• **Configuring a Dedicated Load Balancer** • (Optional) **Configuring Routes for a Cluster**	**Configuring VPC Endpoints**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Choosing a Networking Mode

1. Do clients access our service from a public network environment?
   - Yes: Use **Dedicated Load Balancer**, but the load balancers must be allocated public IP addresses.
   - No: Go to 2.
2. Since the clients are on the internal network, do they share the same VPC as our cluster?
   - Yes: This is intra-VPC access. Does our cluster need load balancers to enhance throughput and stability?
     - Yes: Use **Dedicated Load Balancer**.
     - No: Use **Private Network Access**.
   - No: This is cross-VPC access. Do the VPCs have overlapping CIDR blocks?
     - Yes: Use **VPC Endpoint**.
     - No: Use **Dedicated Load Balancer**. VPC peering connections or cluster routes are required.

2.5.2 Configuring Public Network Access

For enterprise applications, to make CSS Elasticsearch clusters securely accessible from the Internet, you can enable public network access for them. When public network access is enabled for a cluster, the cluster is automatically assigned a public IP address with dedicated dynamic BGP bandwidth, making it accessible from the Internet via HTTPS. You can configure public network access control by IP addresses or IP address ranges.

To enable public network access for Elasticsearch clusters, a shared load balancer is typically used for load balancing. If your workloads require quicker access, you are advised to use a dedicated load balancer to connect to your clusters. For details about its configuration, see **Configuring a Dedicated Load Balancer**.

Impact on Billing

When public network access is enabled for an Elasticsearch cluster, the cluster is automatically assigned an EIP and public network bandwidth, which will both incur expenses.

Constraints

- Public network access can be enabled for an Elasticsearch cluster only if the cluster version is 6.5.4 or later, and **Security Mode** and **HTTPS Access** are both enabled.
- Public network access and VPC Endpoint share the same shared load balancer. If you configure a whitelist for public network access control, this whitelist is deployed to the shared load balancer. As such, it will control access not only from the public network but also from private IP addresses through VPCEP. In this case, you need to add IP address **198.19.128.0/17** to the public access control whitelist to allow traffic through VPCEP.

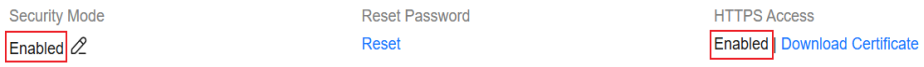
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Enabling Public Network Access

To enable public network access for an existing cluster, perform the following steps:

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, go to the **Network Information** area, and check whether **Security Mode** and **HTTPS Access** are enabled.
   - If they are enabled, go to the next step to enable public network access.
   - If either one is disabled, public network access cannot be enabled for the cluster.

   Figure 2-3 Checking whether public network access can be enabled for the cluster

5. Click **Enable** next to **Public Network Access**. In the displayed dialog box, configure the necessary settings.

   Table 2-36 Enabling public network access

Parameter	Description
Bandwidth	Cluster bandwidth for public network access. Value range: 1 Mbit/s to 200 Mbit/s
Configure Whitelist	Control public network access to the cluster using a whitelist. • If a whitelist is configured, only IP addresses that are on this whitelist can access the cluster over the public network. Click **+Add**. In the displayed text box, enter IP addresses or CIDR blocks that are allowed to access the cluster from the public network. Separate them using commas (,). Each value must be unique. An example of valid values: **192.168.1.1,10.0.0.0/24**. Examples of invalid values: **0.0.0.0**, **xx.xx.xx.xx/0**, **172.16.0.0-172.16.255.255**, non-standard formats (e.g., **192.168.1**), and duplicate values. • If no whitelist is configured, all public IP addresses can access the cluster. However, this can be a security risk and should be avoided.

6. Click **OK** to enable public network access.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

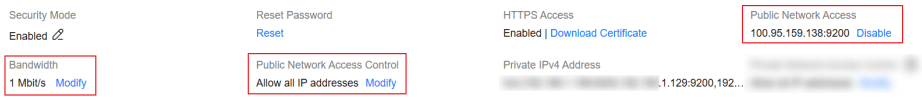
After public network access is enabled, the **Network Information** area shows the public IP address, public network access control, and bandwidth information.

Managing Public Network Access

When public network access is enabled, you can check the public IP address, and modify the bandwidth and access control settings.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, manage public network access settings in the **Network Information** area.

Figure 2-4 Managing public network access



- **Checking the public IP address**

  Record the IP address and port under **Public Network Access** for later use.

- **Modifying public network access control settings**

  Click **Modify** under **Public Network Access Control**. In the displayed dialog box, add or remove IP addresses or CIDR blocks to or from the whitelist. Click **OK** to save the change.

- **Modifying public network bandwidth**

  Click **Modify** under **Bandwidth**. In the displayed dialog box, change the bandwidth for optimal network performance. Click **OK** to save the change.

Disabling Public Network Access

If public network access is no longer required for a cluster, disable it to release resources.

WARNING

After the public IP address is disassociated, the cluster can no longer be accessed from the Internet through this IP address. If you disable public network access for a cluster and then re-enable it, the public IP address for accessing the cluster may change. Exercise caution.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, find **Public Network Access** in the **Network Information** area, and click **Disable** under it. In the displayed dialog box, enter **CONFIRM** and click **OK**.

   After public network access is disabled, the **Network Information** area no longer shows the public IP address, public network access control, or bandwidth information.

Accessing a Cluster Through the Public IP Address

After public network access is enabled, the cluster is assigned a public IP address. External users can securely access this cluster through this IP address plus a port number.

For example, if the public IP address is **10.62.xxx.xxx** and the port number is **9200**, run the following cURL command to view indexes in the cluster.

```
curl -u username:password -k 'https://10.62.xxx.xxx:9200/_cat/indices'
```

where, **username** and **password** indicate the username and password of the HTTPS-enabled security-mode cluster.

2.5.3 Configuring a Dedicated Load Balancer

In big data processing scenarios, load balancers are essential for distributing traffic and improving the availability and performance of Elasticsearch clusters. However, default shared load balancers have functional and performance limitations, making them inadequate for complex, high-concurrency data processing. Dedicated load balancers, with their richer features and superior performance, offer a more robust solution. This topic describes how to configure dedicated load balancers for more efficient and secure cluster access.

Overview

Advantages of connecting to a cluster through a dedicated load balancer:

- A non-security cluster can also utilize the capabilities of the Elastic Load Balance (ELB) service.
- You can use custom certificates for HTTPS two-way authentication.
- Seven-layer traffic monitoring and alarm configuration are supported, allowing you to keep close track of the cluster status.

There are eight different ELB service forms for clusters in different security modes to connect to a dedicated load balancer. **Table 2-37** describes the ELB capabilities for different cluster configurations. **Table 2-38** describes the configurations for different ELB service forms.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-37 ELB capabilities for different clusters

Security Mode	Service Form Provided by ELB for External Systems	ELB Load Balancing	ELB Traffic Monitoring	ELB Two-way Authentication
Non-security	No authentication	Supported	Supported	Not supported
	One-way authentication Two-way authentication	Supported	Supported	Supported
Security mode + HTTP	Password authentication	Supported	Supported	Not supported
	One-way authentication + Password authentication Two-way authentication + Password authentication	Supported	Supported	Supported
Security mode + HTTPS	One-way authentication + Password authentication Two-way authentication + Password authentication	Supported	Supported	Supported

Table 2-38 Configurations for different ELB service forms depending on the cluster

Security Mode	Service Form Provided by ELB for External Systems	ELB Listener	ELB Listener	ELB Listener	Backend Server Group	Backend Server Group	Backend Server Group
		Frontend Protocol	Frontend Port	SSL Authentication	Backend Protocol	Health Check Port	Health Check Path
Non-security	No authentication	HTTP	9200	No authentication	HTTP	9200	/

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Security Mode	Service Form Provided by ELB for External Systems	ELB Listener	ELB Listener	ELB Listener	Backend Server Group	Backend Server Group	Backend Server Group
		Frontend Protocol	Frontend Port	SSL Authentication	Backend Protocol	Health Check Port	Health Check Path
	One-way authentication	HTTPS	9200	One-way authentication	HTTP	9200	
	Two-way authentication	HTTPS	9200	Two-way authentication	HTTP	9200	
Security mode + HTTP	Password authentication	HTTP	9200	No authentication	HTTP	9200	/_opendistro/_security/health
	One-way authentication + Password authentication	HTTPS	9200	One-way authentication	HTTP	9200	
	Two-way authentication + Password authentication	HTTPS	9200	Two-way authentication	HTTP	9200	
Security mode + HTTPS	One-way authentication + Password authentication	HTTPS	9200	One-way authentication	HTTPS	9200	
	Two-way authentication + Password authentication	HTTPS	9200	Two-way authentication	HTTPS	9200	

Constraints

- You are not advised to connect a load balancer that has been associated with a public IP address to a non-security mode cluster. Allowing public network access through such a load balancer may cause security risks because a non-

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

security mode cluster can be accessed via HTTP without security authentication.

- HTTPS-enabled security-mode clusters do not support HTTP-based frontend authentication. If the frontend uses HTTP, disable security mode for the clusters first. For details, see **Changing the Security Mode**. Before changing the security mode, disable load balancing first. After the security mode is changed, enable load balancing again.

Prerequisites

- A dedicated load balancer has been created. For details, see **Creating a Dedicated Load Balancer**. This load balancer must meet the following requirements:
  - Its VPC is the same as that of the CSS cluster. The network between the two is connected.
  - **IP as a Backend** is enabled. This is necessary to connect a dedicated load balancer to a CSS cluster.
  - Determine whether to configure an EIP based on service needs. A public IP address is displayed for the load balancer connecting the CSS cluster only if an EIP is configured. This will enable public network access to the cluster through this load balancer.
- If the ELB listener uses HTTPS, upload a server certificate or CA certificate to the ELB console. For details, see **Adding a Certificate**.
  - If one-way authentication is used, upload a server certificate.
  - If two-way authentication is used, upload a server certificate and a CA certificate.

Connecting a Cluster to a Load Balancer

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Cluster Access** tab, and then click the **Load Balancing** tab. On the **Elasticsearch** tab, toggle on **Load Balancing**. In the displayed dialog box, set the parameters.

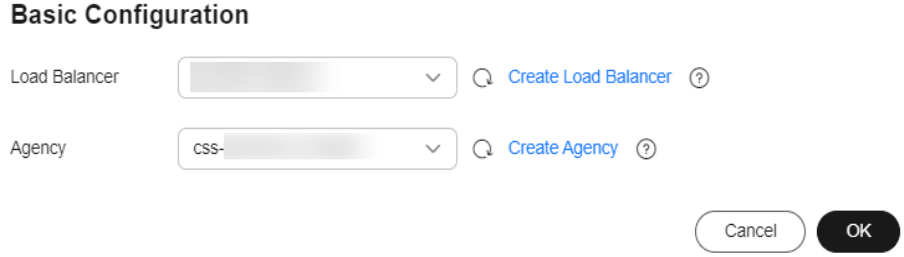
Table 2-39 Configuring load balancing

Parameter	Description
Load Balancer	Select the dedicated load balancer you have created earlier. To create a dedicated load balancer, see **Creating a Dedicated Load Balancer**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Agency	To configure a load balancer, you must have the permission to access ELB resources. By configuring an IAM agency, you can authorize CSS to access its ELB resources through an associated account. • If you are configuring an agency for the first time, click **Automatically Create IAM Agency** to create **css-elb-agency**. • If there is an IAM agency automatically created earlier, you can click **One-click authorization** to have the permissions associated with the **ELB Administrator** role or the **ELB FullAccess** system policy deleted automatically, and have the following custom policies added automatically instead to implement more refined permissions control. "elb:loadbalancers:list", "elb:loadbalancers:get", "elb:certificates:list", "elb:healthmonitors:*", "elb:members:*", "elb:pools:*", "elb:listeners:*" • To use **Automatically Create IAM Agency** and **One-click authorization**, the following minimum permissions are required: "iam:agencies:listAgencies", "iam:roles:listRoles", "iam:agencies:getAgency", "iam:agencies:createAgency", "iam:permissions:listRolesForAgency", "iam:permissions:grantRoleToAgency", "iam:permissions:listRolesForAgencyOnProject", "iam:permissions:revokeRoleFromAgency", "iam:roles:createRole" • To use an IAM agency, the following minimum permissions are required: "iam:agencies:listAgencies", "iam:agencies:getAgency", "iam:permissions:listRolesForAgencyOnProject", "iam:permissions:listRolesForAgency"

Figure 2-5 Enabling load balancing



5. Click **OK** to enable load balancing.
   Load balancer information is displayed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

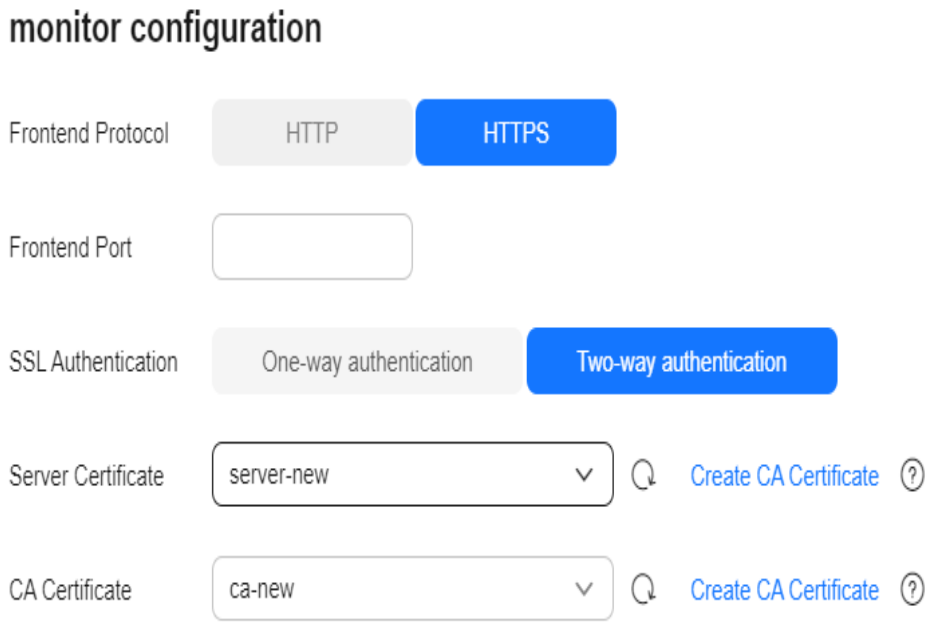
6. In the **Listener** area, click to configure a listener to ensure proper accessibility of the CSS cluster.

Table 2-40 Listener configuration

Parameter	Description
Frontend Protocol	Protocol used by the client and listener to distribute traffic. Select **HTTP** or **HTTPS**. Select this protocol based on your connectivity needs.
Frontend Port	Port used by the client and listener to distribute traffic. Set this parameter based on site requirements.
SSL Authentication	Authentication mode for the client to access the server. Set this parameter only when **Frontend Protocol** is set to **HTTPS**. Both one-way and two-way authentication are supported. Select an authentication mode that suits your needs.
Server Certificate	The server certificate is used for SSL handshake. The certificate content and private key must be provided. It is required only when **Frontend Protocol** is set to **HTTPS**. Select the server certificate created on ELB.
CA Certificate	Also called client CA public key certificate. It is used to verify the issuer of a client certificate. It is required only when **SSL Authentication** is set to **Two-way authentication**. Select the CA certificate created on ELB. When HTTPS two-way authentication is enabled, an HTTPS connection can be established only when the client can provide the certificate issued by a trusted CA.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 2-6 Listener configuration



7. (Optional) In the **Listener** area, click **Configure** next to **Access Control** to go to the listener list of the load balancer. Click Configure in the Access Control column of a listener to configure access control for that listener. You can specify IP addresses that are allowed to access your cluster for enhanced security. For more information, see **What Is Access Control?**.

   WARNING

   Without access control policies, all IP addresses are allowed to access the CSS cluster through this load balancer, which may create security risks.

8. In the **Health Check** area, you can check the health check results for each node IP address, ensuring that each node in the cluster is functioning properly.

   Table 2-41 Health check result description

Health Check Result	Description
Normal	The node IP address is connected.
Abnormal	The node IP address is disconnected.

9. If the cluster no longer needs a dedicated load balancer, disassociate it to release resources.

   Choose **Load Balancing** > **Elasticsearch**, toggle off **Load Balancing**. In the displayed dialog box, click **OK**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

> **CAUTION**
>
> After the load balancer is disassociated, any listener or backend server group configurations will be permanently deleted.

Accessing a Cluster Through a Load Balancer by Executing cURL Commands

1. In the navigation pane on the left, choose **Clusters**.
2. Log in to the **CSS management console**.
3. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
4. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
5. Click the **Cluster Access** tab, and then click the **Load Balancing** tab. On the **Elasticsearch** tab, record the private or public IP address or IPv6 address of the load balancer, as well as the frontend protocol/port of the listener.

> **CAUTION**
>
> You are not advised to connect a load balancer that has been associated with a public IP address to a non-security mode cluster. Allowing public network access through such a load balancer may cause security risks because a non-security mode cluster can be accessed via HTTP without security authentication.

6. Run the following cURL commands on an ECS to check whether the dedicated load balancer can connect to the cluster.

Table 2-42 Commands for accessing different types of clusters

Security Mode	Service Form Provided by ELB for External Systems	cURL Command for Accessing a Cluster
Non-security	No authentication	`curl http://IP:port`
	One-way authentication	`curl --cacert ./ca.crt https://IP:port`
	Two-way authentication	`curl --cacert ./ca.crt --cert ./client.crt --key ./client.key https://IP:port`

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Security Mode	Service Form Provided by ELB for External Systems	cURL Command for Accessing a Cluster
Security mode + HTTP	Password authentication	curl http://IP:port -u *user.pwd*
	One-way authentication + Password authentication	curl --cacert ./ca.crt https://IP:port -u *user.pwd*
	Two-way authentication + Password authentication	curl --cacert ./ca.crt --cert ./client.crt --key ./client.key https://IP:port -u *user.pwd*
Security mode + HTTPS	One-way authentication + Password authentication	curl --cacert ./ca.crt https://IP:port -u *user.pwd*
	Two-way authentication + Password authentication	curl --cacert ./ca.crt --cert ./client.crt --key ./client.key https://IP:port -u *user.pwd*

Table 2-43 Variables

Variable	Description
IP	IP address of a load balancer instance.
port	Frontend protocol and port configured for the listener.
user	Username of the cluster. This parameter is required only for a security-mode cluster.
pwd	Password of the username above. This parameter is required only for a security-mode cluster.

If cluster information is returned, the connection is successful.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

See also: **Sample Code for Accessing a Cluster Using HTTPS Two-way Authentication Through a Load Balancer (Java)**.

Sample Code for Accessing a Cluster Using HTTPS Two-way Authentication Through a Load Balancer (Java)

This section provides sample code that can be used on a Java client to access a cluster using HTTPS two-way authentication through a load balancer.

Sample Code for ESSecuredClientWithCerDemo

```
import org.apache.commons.io.IOUtils;
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 java.io.FileInputStream;
import java.io.IOException;
import java.security.KeyStore;
import java.security.SecureRandom;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.KeyManagerFactory;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManagerFactory;
public class ESSecuredClientWithCerDemo {
    private static final String KEY_STORE_PWD = "";
    private static final String TRUST_KEY_STORE_PWD = "";
    private static final String CA_JKS_PATH = "ca.jks";
    private static final String CLIENT_JKS_PATH = "client.jks";
    private static final String ELB_ADDRESS = "127.0.0.1";
    private static final int ELB_PORT = 9200;
    private static final String CSS_USERNAME = "user";
    private static final String CSS_PWD = "";
    public static void main(String[] args) {
      // Create a client.
       RestHighLevelClient client = initESClient(ELB_ADDRESS, CSS_USERNAME, CSS_PWD);
       try {
          // Search by using match_all, which is equivalent to {\"query\": {\"match_all\": {}}}.
          SearchRequest searchRequest = new SearchRequest();
          SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();
          searchSourceBuilder.query(QueryBuilders.matchAllQuery());
          searchRequest.source(searchSourceBuilder);
          // query
          SearchResponse searchResponse = client.search(searchRequest, RequestOptions.DEFAULT);
          System.out.println("query result: " + searchResponse.toString());
          SearchHits hits = searchResponse.getHits();
          for (SearchHit hit : hits) {
             System.out.println(hit.getSourceAsString());
          }
          System.out.println("query success");
          Thread.sleep(2000L);
       } catch (InterruptedException | IOException e) {
          e.printStackTrace();
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        } finally {
            IOUtils.closeQuietly(client);
        }
    }
    private static RestHighLevelClient initESClient(String clusterAddress, String userName, String password) {
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials(userName,
password));
        SSLContext ctx = null;
        try {
            KeyStore ks = getKeyStore(CLIENT_JKS_PATH, KEY_STORE_PWD, "JKS");
            KeyManagerFactory kmf = KeyManagerFactory.getInstance("SunX509");
            kmf.init(ks, KEY_STORE_PWD.toCharArray());
            KeyStore tks = getKeyStore(CA_JKS_PATH, TRUST_KEY_STORE_PWD, "JKS");
            TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509");
            tmf.init(tks);
            ctx = SSLContext.getInstance("SSL", "SunJSSE");
            ctx.init(kmf.getKeyManagers(), tmf.getTrustManagers(), new SecureRandom());
        } catch (Exception e) {
            e.printStackTrace();
        }
        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(ctx, new HostnameVerifier() {
            @Override
            public boolean verify(String arg0, SSLSession arg1) {
                return true;
            }
        });
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
            credentialsProvider);
        RestClientBuilder builder = RestClient.builder(new HttpHost(clusterAddress, ELB_PORT, "https"))
            .setHttpClientConfigCallback(httpClientConfigCallback);
        RestHighLevelClient client = new RestHighLevelClient(builder);
        return client;
    }
    private static KeyStore getKeyStore(String path, String pwd, String type) {
        KeyStore keyStore = null;
        FileInputStream is = null;
        try {
            is = new FileInputStream(path);
            keyStore = KeyStore.getInstance(type);
            keyStore.load(is, pwd.toCharArray());
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            IOUtils.closeQuietly(is);
        }
        return keyStore;
    }
}
```

Sample Code for SecuredHttpClientConfigCallback

```
import org.apache.http.client.CredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.common.Nullable;
import java.util.Objects;
class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
    @Nullable
    private final CredentialsProvider credentialsProvider;
    /**
     * The {@link SSLIOSessionStrategy} for all requests to enable SSL / TLS encryption.
     */
    private final SSLIOSessionStrategy sslStrategy;
    /**
     * Create a new {@link SecuredHttpClientConfigCallback}.
     *
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
     * @param credentialsProvider The credential provider, if a username/password have been supplied
     * @param sslStrategy         The SSL strategy, if SSL / TLS have been supplied
     * @throws NullPointerException if {@code sslStrategy} is {@code null}
     */
    SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
        @Nullable final CredentialsProvider credentialsProvider) {
        this.sslStrategy = Objects.requireNonNull(sslStrategy);
        this.credentialsProvider = credentialsProvider;
    }
    /**
     * Get the {@link CredentialsProvider} that will be added to the HTTP client.
     *
     * @return Can be {@code null}.
     */
    @Nullable
    CredentialsProvider getCredentialsProvider() {
        return credentialsProvider;
    }
    /**
     * Get the {@link SSLIOSessionStrategy} that will be added to the HTTP client.
     *
     * @return Never {@code null}.
     */
    SSLIOSessionStrategy getSSLStrategy() {
        return sslStrategy;
    }
    /**
     * Sets the {@linkplain HttpAsyncClientBuilder#setDefaultCredentialsProvider(CredentialsProvider)
credential provider},
     *
     * @param httpClientBuilder The client to configure.
     * @return Always {@code httpClientBuilder}.
     */
    @Override
    public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder httpClientBuilder) {
        // enable SSL / TLS
        httpClientBuilder.setSSLStrategy(sslStrategy);
        // enable user authentication
        if (credentialsProvider != null) {
            httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
        }
        return httpClientBuilder;
    }
}
```

pom.xml Sample Code

```
<?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>

    <properties>
        <maven.compiler.source>8</maven.compiler.source>
        <maven.compiler.target>8</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <elasticsearch.version>7.10.2</elasticsearch.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.elasticsearch.client</groupId>
            <artifactId>transport</artifactId>
            <version>${elasticsearch.version}</version>
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        </dependency>
        <dependency>
            <groupId>org.elasticsearch</groupId>
            <artifactId>elasticsearch</artifactId>
            <version>${elasticsearch.version}</version>
        </dependency>
        <dependency>
            <groupId>org.elasticsearch.client</groupId>
            <artifactId>elasticsearch-rest-high-level-client</artifactId>
            <version>${elasticsearch.version}</version>
        </dependency>
        <dependency>
            <groupId>commons-io</groupId>
            <artifactId>commons-io</artifactId>
            <version>2.11.0</version>
        </dependency>
    </dependencies>
</project>
```

2.5.4 Configuring VPC Endpoints

In CSS, you can configure VPC endpoints as dedicated gateways to enable cross-VPC resource access via an internal network without exposing server details. This ensures more secure and reliable internal connectivity for enterprise applications. A VPC endpoint can be accessed via an IPv4 address or private domain name.

- An IPv4 address is automatically allocated when VPC Endpoint is enabled.
- A private domain name is allocated only when you enable private domain names.

VPC Endpoint uses a shared load balancer for internal network access. If your workloads require faster access, we recommend that you use a dedicated load balancer to handle access to your cluster. For details, see **Configuring a Dedicated Load Balancer**.

Impact on Billing

VPC endpoints, if created for the cluster, will incur extra fees, depending on the resource usage. For details, see **VPC Endpoint Billing**.

Constraints

- You need specific permissions to create VPC endpoints. For details, see **VPCEP Permissions**.
- After VPCEP is enabled, access to CSS through a VPCEP IP address or private domain name from within the internal network is not controlled by any cluster security group rules. Rather, you need to configure a VPCEP whitelist to implement access control. For details, see section "Configuring Access Control for an Interface VPC Endpoint" in *VPC Endpoint User Guide*.
- Public network access and VPC Endpoint share the same shared load balancer. If you configure a whitelist for public network access control, this whitelist is deployed to the shared load balancer. As such, it will control access not only from the public network but also from private IP addresses through VPCEP. In this case, you need to add IP address **198.19.128.0/17** to the public access control whitelist to allow traffic through VPCEP.
- When you access a cluster through the IP address or private domain name of a VPC endpoint, you are not subject to the cluster's security group rules, and you can configure a whitelist for access control.

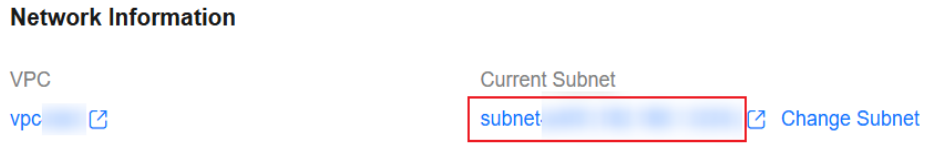
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Enabling VPC Endpoint

If VPC Endpoint was not enabled when the cluster was created, you can enable it as follows:

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, check the cluster's subnet information in the **Network Information** area. When you add a VPC endpoint, an IP address that belongs to the current subnet of the cluster is automatically assigned to it.

   Figure 2-7 Checking the current subnet

5. Choose **Cluster Access** > **VPC Endpoint**.
6. Toggle on **VPC Endpoint**. In the displayed dialog box, select relevant options.

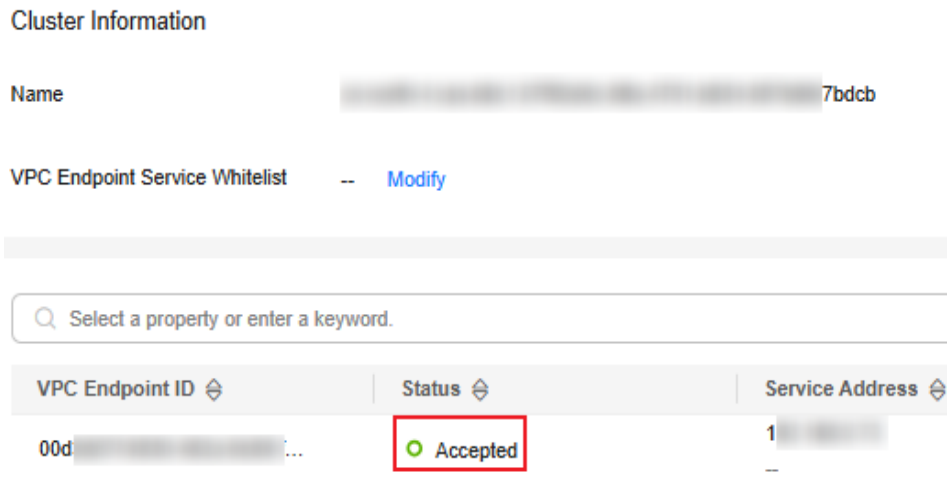
   Table 2-44 Options for enabling VPC Endpoint

Option	Description
Create Private Domain Name	Whether to create a private domain name for the VPC endpoint. • Enable: The system automatically assigns a private domain name to the VPC endpoint. After cluster creation, you can check this private domain name on the **VPC Endpoint** tab of the cluster details page. • Disable: No private domain name will be configured for the VPC endpoint. The cluster can only be accessed through an IP address assigned to the VPC endpoint.

7. Click **OK** to enable VPC Endpoint. After the VPC endpoint is enabled, its information is displayed below. When its status changes to **Accepted**, the current cluster can be accessed through the VPC endpoint.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 2-8 Checking the VPC endpoint status



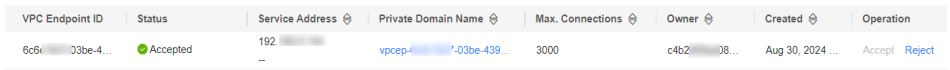
Managing VPC Endpoints

After VPC Endpoint is enabled for a cluster, you can set access control, check VPC endpoint information, and deny access from specific VPC endpoints.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Cluster Access** > **VPC Endpoint**.
5. **Configure access control for a VPC endpoint.**
   a. Click **Modify** on the right of **VPC Endpoint Whitelist**. In the displayed dialog box, add accounts that are allowed to access the cluster through the VPC endpoint. If no account is added or the account ID is set to *, all users are allowed to access the cluster.
      - Click **Add** to add accounts in **Account ID**. To obtain your authorized account ID, point to your username in the upper right corner, and choose **My Credentials**. Copy the value of **Account ID**.
      - Click **Delete** in the **Operation** column to delete an authorized account.

   b. Click **OK**.
6. **Check VPC endpoint information.**
   The VPC endpoint list shows the VPC endpoints created for the current cluster. You can obtain their **Status**, **Service Address**, and **Private Domain Name**.

Figure 2-9 Managing VPC endpoints



7. **Modify VPC endpoint status** to make the cluster accessible or inaccessible through specific endpoints.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- In the VPC endpoint list, select the target endpoint and click **Reject** in the **Operation** column. If the endpoint status changes to **Rejected**, it means the cluster is no longer accessible through this endpoint.
- Select an endpoint whose status is **Rejected**. Click **Accept** in the **Operation** column. If the endpoint status changes to **Accepted**, it means the cluster is accessible again through this endpoint.

Disabling VPC Endpoint

If the cluster no longer requires cross-VPC access via VPC endpoints, disable VPC Endpoint to release resources.

WARNING

After VPC Endpoint is disabled, the cluster is no longer accessible through a VPCEP IP address or private domain name. If you disable VPC Endpoint and then re-enable it, the VPCEP IP address or private domain name for accessing the cluster may change. When it happens, you may need to update the client connection. If you just need to temporarily disable VPC Endpoint (rather than permanently releasing its resources), do so by rejecting specific VPC endpoints. For details, see 7.

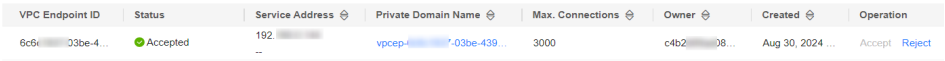
1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Cluster Access** > **VPC Endpoint**.
5. Toggle off **VPC Endpoint**. In the displayed dialog box, enter **CONFIRM** and click **OK**.

Accessing a Cluster Through a VPC Endpoint

1. Obtain the private domain name or IP address of a VPC endpoint.

   In the VPC endpoint list, check **Service Address** or **Private Domain Name**.

   Figure 2-10 Checking the service address or private domain name of a VPC endpoint

2. On the client (e.g., an ECS), run the curl command to access the cluster.

   For example, run the following command to check the cluster's index information:

   - For a cluster with the security mode disabled:
     ```
     curl "http://<host>:9200/_cat/indices"
     ```
   - For a security-mode cluster that uses HTTP:
     ```
     curl -u <user>:<password> "http://<host>:9200/_cat/indices"
     ```
   - For a security-mode cluster that uses HTTPS:
     ```
     curl -u <user>:<password> -k "https://<host>:9200/_cat/indices"
     ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

where, **user** and **password** are the username and password used to access the cluster, and **host** indicates the VPC endpoint's private domain name or IP address.

2.5.5 Configuring Routes for a Cluster

By default, a CSS Elasticsearch cluster cannot access a target service—such as a client program or a Lightweight Directory Access Protocol (LDAP) service—that is located in a different VPC. To enable it, you need to configure a route for the Elasticsearch cluster.

1. Establish a VPC peering connection to connect two VPCs.
2. On CSS, configure a route for the Elasticsearch cluster that points to the client.
3. Verify network connectivity after starting services.

Configuring routes to enable cluster communication offers the following advantages:

- Improved cross-VPC communication performance: A route table specifies the traffic forwarding paths, so that traffic does not have to pass through the public network.
- Secure communication: Private networks can communicate through VPC peering connections, which is more secure than transmitting data over the public network.
- Flexible access control policies: You can customize access permissions for specific subnets.
- Support for a hybrid cloud architecture: Services can communicate across VPCs in different environments.

Constraints

- By default, routes can be configured only for Elasticsearch clusters whose image version is no earlier than x.x.x_25.3.0_x.x.x.
- Adding cluster routes will alter network connections, possibly causing disruptions in the network. Exercise caution.

Prerequisites

- The Elasticsearch cluster is Available, and port 9200 is allowed in the inbound direction by the cluster's security group.
- The client is running properly, and the server's security group allows port 9200 in the outbound direction.

Establishing a VPC Peering Connection

Before configuring routes for a CSS cluster, establish a connection between the CSS cluster's VPC and that of the client, that is, a VPC peering connection.

- If the CSS cluster and the client belong to the same account, see section "Creating a VPC Peering Connection to Connect Two VPCs in the Same Account" in *Virtual Private Cloud User Guide*.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- If the CSS cluster and the client belong to different accounts, see section "Creating a VPC Peering Connection to Connect Two VPCs in Different Accounts" in *Virtual Private Cloud User Guide*.

Set the peer VPC to the VPC where the CSS cluster is located and the local VPC to the VPC where the client is located.

When adding routes for a VPC peering connection, configure the destination addresses based on service requirements.

If the CSS cluster and the client share the same security group, there is no need to configure security group rules for the instances in the VPC at either end of the VPC peering connection.

Configuring a Route for the Elasticsearch Cluster

Configure a route for the Elasticsearch cluster that points to the client, enabling the cluster to access the client.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, click **Add Route** under **Cluster Route** in the **Network Information** area.
5. In the displayed dialog box, configure the route information.

   Table 2-45 Adding a route

Parameter	Description
IP Address	Enter the IP address (or a CIDR block) of the client or server that the cluster needs to access, for example, **10.10.1.0**.
Subnet Mask	Enter the subnet mask used with the IP address. The subnet mask must align with the network part of the IP address. That is: • The network bits (the 1s) of the subnet mask must encompass those of the IP address. • The host bits (the 0s) of the subnet mask must accommodate the IP address's host bits. The subnet mask should not cover too wide a range, avoiding unnecessary IP address exposure or waste of the address space. 255.255.255.0 (that is, /24) is recommended, which applies to most internal networking scenarios.

6. Click **OK** to add the route.

   After the route is added, click **View Route** to check the route information. You can click **Add Route** to add another route; or click **Delete Route** to delete

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

one—In the **Delete Route** dialog box, select a route, enter **DELETE** manually, and then click **OK**.

Testing Connectivity

Connectivity between the Elasticsearch cluster and client cannot be verified right away. After the route is configured, start services. If the search service can be accessed, the route is configured successfully.

2.6 Security Settings

2.6.1 Creating Cluster Users and Granting Permissions

An Elasticsearch cluster with security mode enabled uses access control. Initially, only the administrator account can log in to Kibana and access cluster data. In real-world business scenarios, different users (such as developers, O&M personnel, and data analysts) need to log in to the cluster using their own identities and perform only authorized operations on authorized indexes and features. To meet this requirement, CSS provides user and permission management based on the role-based access control (RBAC) model. Administrators can create users, configure roles and permissions, and assign roles to users through role mappings in the **Security** module of Kibana. This enables fine-grained access control at the cluster, index, document, and field levels.

How the Feature Works

CSS security-mode clusters implement fine-grained access control based on the opendistro_security plugin. With the RBAC model at its core, this plugin implements authorization through the chain of users → role mappings → roles → permissions.

Figure 2-11 RBAC authorization model



Table 2-46 Core concepts

Concept	Description
User	The entity that sends requests to the cluster. A user has a username and password and can be associated with backend roles and custom attributes.
Role mapping	Associates users (or backend roles) with specific roles, granting them all permissions associated with those roles. A role can be mapped to multiple users (or backend roles). For example, if the **readall** role is mapped to user **Bob**, **Bob** will have all permissions of the **readall** role after authentication.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Concept	Description
Role	A collection of permissions or action groups that defines the operations allowed on clusters, indexes, documents, and fields.
Permission	A specific action, such as **indices:admin/create** (creating an index).
Action group	A collection of permissions. For example, the predefined **search** action group contains search-related permissions such as **search** and **msearch**.

RBAC answers the question "Who can do what in the cluster?", while Tenant addresses the question "What Kibana objects can users see and edit?" RBAC and Tenant work together to provide fine-grained access control. RBAC determines the operations that users can perform on cluster resources, while Tenant controls the visibility and sharing of Kibana resources, such as dashboards and visualizations. Three tenant types are available to define the visibility of Kibana resources.

Table 2-47 Tenant types

Type	Sharing Scope	Applicable Scenario
Global (global tenant)	Accessible to all users.	Shared dashboards, public monitoring views, and other common resources
Private (private tenant)	Accessible only to the resource creator. Even the administrator cannot directly view private tenant content.	Private reports used by individuals for development and debugging
Custom (custom tenant)	Administrators can precisely specify a group of users or roles through role mappings and further assign Tenant Read (read-only) or Tenant Write (write) permissions.	Team collaboration and isolation by department

Constraints

- Usernames, role names, and tenant names must be unique.
- You are advised not to use the same name for indexes and users. This helps avoid permission management confusion.
- This topic uses Kibana 7.10.2 as an example. The user interface may vary slightly depending on the version.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Prerequisites

- The security mode has been enabled for your cluster. Only security-mode Elasticsearch clusters support user permission management.
- You have obtained the administrator account. The username is **admin** and the password is the administrator password configured when the cluster was created.

Logging In to the Security Page of Kibana

Log in to the **Security** page of Kibana using the administrator account to manage cluster permissions.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to Kibana using the **admin** account.
4. After successful login, choose **Security** from the navigation pane on the left. The permission management page is displayed.

Creating a User

Create a new cluster user. The user can then be mapped to roles to be granted the corresponding permissions.

1. On the **Security** page, choose **Internal users** from the navigation pane on the left.
2. On the **Internal Users** page, click **Create internal user**.
3. On the **Create internal user** page, configure the required parameters.

Table 2-48 User configuration

Parameter	Mandatory (Yes/No)	Description
Username	Yes	Username. In this example, enter **test**.
Password/Re-enter password	Yes	Enter and confirm the login password. The two passwords must be identical.
Backend roles	No	Backend role names from an external authentication system (such as LDAP or SAML). Backend roles are used to map external users to security roles. For details, click **Learn more** on the page.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory (Yes/No)	Description
Attributes	No	Custom user attributes. They can be used as dynamic variables in document-level security (DLS) queries to implement fine-grained data access control based on user attributes. For details, click **Learn more** on the page.

4. Click **Create**.

   After the user is created, it is displayed in the **Internal Users** list.

Creating a Role and Defining Its Permissions

Define a role and specify the operations that the role is allowed to perform on clusters and indexes, as well as the scope of these operations. The permissions defined in the role can then be granted to users by mapping the role to them.

NOTE

The system provides multiple predefined roles (such as **kibana_user** and **readall**). If the predefined roles meet your requirements, you are advised to use them directly instead of creating new ones.

1. On the **Security** page, choose **Roles** from the navigation pane on the left.
2. On the **Roles** page, click **Create Role**.
3. On the **Create Role** page, configure the required parameters.

Table 2-49 Role configuration

Parameter	Mandatory (Yes/No)	Description
Name	Yes	Role name. In this example, enter **role_test**.
Cluster permissions	No	Cluster-level permissions. • Select a predefined action group, such as **cluster_monitor** (read-only cluster monitoring) or cluster_manage (**cluster management**). In this example, select **cluster_monitor**. The **cluster_monitor** action group allows users to view cluster status in read-only mode, including cluster health, node information, and statistics, but does not allow modification operations. • Leaving this parameter blank means not to grant any cluster-level permissions.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory (Yes/No)	Description
Index Permissions	No	Index-level permissions. Set the following parameters: • **Index**: Mandatory. The index name or wildcard pattern to which permissions are granted, such as **my_store** or **logs-***. • **Index permissions**: Mandatory. Operations allowed on the index, such as **read** and **write**. • **Document Level Security**: Optional. Sets document-level filters so that users with this role can only access documents that meet the specified conditions. • **Field Level Security**: Optional. Sets field allowlists or blocklists to control which fields are visible to users. • **Anonymization**: Optional. Specifies fields that need to be anonymized to protect data security.
Tenant Permissions	No	Tenant permissions. • **Tenant patterns**: Specifies tenant spaces that users can access, such as global_tenant or custom tenant names. • Permissions: Tenant space permissions, which can be **Read only** or **Read and Write**.

4. Click **Create**.

   After the role is created, it is displayed in the **Roles** list.

Mapping Roles to Users

Map roles to users to grant users all permissions associated with the mapped roles.

1. On the **Security > Roles** page, click a role name, such as **role_test**. The role details page is displayed.
2. Click the **Mapped users** tab and then click **Manage mapping**.
3. On the **Map user** page, click the drop-down list box next to **Users** and select the target user, for example, **test**.

   NOTE

   A role can be mapped to multiple users. You can also use **Backend roles** to batch-map user groups from external authentication systems.
4. Click **Map** to complete the role mapping.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

After the mapping is successful, the target user **test** is displayed in the **Mapped users** list.

Verifying User Permissions

Log in to the cluster as the new user and perform operations to verify that the user's permission configuration has taken effect as expected.

1. Log out of the current **admin** account and log in to Kibana again using the username **test** and the configured password.
2. On Kibana, expand the menu in the upper-left corner, and choose **Dev Tools**.
3. Run the following two commands to verify that user **test** has the permissions granted by role **role_test**, that is, only the **cluster_monitor** permission.
   - Query cluster health status.
     ```
     GET /_cluster/health?pretty
     ```
     Expected result: HTTP status code 200 is returned and cluster health information is displayed, indicating that the **cluster_monitor** permission has taken effect.
   - Create an index.
     ```
     PUT /my_test
     ```
     Expected result: HTTP status code 403 Forbidden is returned, indicating that user **test** does not have permission to create indexes and permission isolation is working.

     NOTE

     If you need to grant user **test** the permission to create indexes, add the corresponding permission (for example, **indices:admin/create**) to **Index Permissions** of **role_test** based on the permission name in the 403 error message. You do not need to create a new user.

Common Role Permission Configuration Examples

In real-world scenarios, users in different roles require different levels of permissions. This section provides role permission configurations for some common scenarios. You can refer to these examples when configuring roles.

Table 2-50 Role permission configuration examples

Scenario	Applicable Roles	Core Permissions
Read-only access to index data	Data analysts, report query users	Read permissions on specified indexes (without cluster permissions)
Read/write access to index data	Application developers, data ingestion service accounts	Read and write permissions on specified indexes (without cluster permissions)
Read-only cluster monitoring	O&M personnel, inspection accounts	Cluster monitoring permissions (without index data permissions)

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Scenario	Applicable Roles	Core Permissions
Index and document management	Application service accounts, data pipeline ingestion accounts	Full index and document management permissions (without cluster permissions)
Dashboard editing (read/write)	O&M engineers, BI application developers	Kibana dashboard permissions + Tenant read/write permissions

- **Read-only access to index data**

  Scenario: Common users need to query data in specified indexes through Kibana but are not allowed to perform any write, update, or delete operations. They are also not allowed to perform cluster-level operations, preventing accidental operations that may affect production data.

  User permission configuration: Map the user to the role **role_index_readonly**.

  Table 2-51 Permission configuration for read-only access to index data

Configuration Item	Sub-Configuration Item	Description
Name	-	Role name: **role_index_readonly**
Cluster permissions	-	Not configured. No cluster-level permissions are required.
Index Permissions	Index	Target index name or wildcard.
	Index permissions	Select the predefined action group **read** (document read), which includes the following permissions: indices:data/read/search (_search) indices:data/read/get (_get) indices:data/read/mget (_mget) indices:data/read/scroll (_scroll) indices:admin/mappings/fields/get* (view field mappings)

  Permission verification:

```
# Expected success (search for documents)
GET /my_store/_search
{ "query": { "match_all": {} } }

# Expected failure (no write permission)
POST /my_store/_doc
{ "name": "test" }

# Expected failure (no cluster-level permissions)
GET /_cluster/health?pretty
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- **Read/write access to index data**

  Scenario: Common users need to write, update, and query data in specified indexes through Kibana but are not allowed to create or delete indexes or modify index mappings and other structural configurations. This ensures that index structures are managed by dedicated personnel.

  User permission configuration: Map the user to the role **role_index_readwrite**.

  Table 2-52 Permission configuration for read/write access to index data

Configuration Item	Sub-Configuration Item	Description
Name	-	Role name: **role_index_readwrite**
Cluster permissions	-	Not configured. No cluster-level permissions are required.
Index Permissions	Index	Target index name or wildcard.
	Index permissions	Select the predefined action groups **read** (document read) and **write** (document write). • The **read** action group includes the following permissions: indices:data/read/search (_search) indices:data/read/get (_get) indices:data/read/mget (_mget) indices:data/read/scroll (_scroll) indices:admin/mappings/fields/get* (view field mappings) • The **write** action group includes the following permissions: indices:data/write/index (create/replace documents) indices:data/write/update (update documents) indices:data/write/delete (delete documents) indices:data/write/bulk (batch write)

Permission verification:

```
# Expected success (query documents)
GET /my_store/_search
{ "query": { "match_all": {} } }

# Expected success (write documents)
POST /my_store/_doc
{ "product": "laptop", "price": 5999 }

# Expected success (delete a document)
DELETE /my_store/_doc/1

# Expected failure (no index creation permission)
PUT /my_store_v2

# Expected failure (no index deletion permission)
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
DELETE /my_store

# Expected failure (no cluster-level permissions)
GET /_cluster/health?pretty
```

- **Read-only cluster monitoring**

  Scenario: O&M personnel or automated inspection scripts need to periodically collect monitoring data such as cluster health status, node loads, and shard distribution. However, they are not allowed to modify cluster status or perform read/write operations on index data, preventing inspection accounts from being misused.

  User permission configuration: Map the user to the role **role_index_monitor**.

  Table 2-53 Permission configuration for read-only cluster monitoring

Configuration Item	Sub-Configuration Item	Description
Name	-	Role name: **role_index_monitor**
Cluster permissions	-	Select the predefined action group **cluster_monitor** (read-only cluster monitoring), which includes the following permissions: cluster:monitor/health (cluster health status) cluster:monitor/stats (cluster statistics) cluster:monitor/nodes/info (node information) cluster:monitor/nodes/stats (node statistics) cluster:monitor/task/get (task information) cluster:monitor/pending_tasks (pending tasks)
Index Permissions	Index	Set to * to view the status of all indexes.
	Index permissions	Select the predefined action group **indices_monitor** (read-only index monitoring), which includes the following permissions: indices:monitor/stats (index statistics) indices:monitor/segments (segment information) indices:monitor/recovery (recovery information) indices:monitor/shard_stores (shard storage information)

  Permission verification:

```
# Expected success (cluster health status)
GET /_cluster/health?pretty

# Expected success (index statistics)
GET /my_store/_stats

# Expected failure (no document query permissions)
GET /my_store/_search
{ "query": { "match_all": {} } }

# Expected failure (no permission to modify the cluster configuration)
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
PUT /_cluster/settings
{ "persistent": { "cluster.routing.allocation.enable": "none" } }
```

- **Index and document management**

  Scenario: Microservice applications or data pipeline service accounts need to use REST APIs to perform full index lifecycle management in Elasticsearch, including index creation, index deletion, index settings updates, and mapping updates, as well as document CRUD operations. Because these accounts are used by applications rather than human users, it is recommended that you clearly define the data scope to reduce potential data security risks.

  User permission configuration: Map the user to the role **role_api_service**.

  Table 2-54 Permission configuration for index and document management

Configuration Item	Sub-Configuration Item	Description
Name	-	Role name: **role_api_service**
Cluster permissions	-	Not configured. No cluster-level permissions are required for pure data operations.
Index Permissions	Index	Use wildcards to specify the scope of indexes, for example, **log-***.
	Index permissions	Select the following predefined action groups/permissions: • **read** (document read) • **write** (document write) • **indices:admin/create** (create indexes) • **indices:admin/delete** (delete indexes) • **indices:admin/settings/update** (modify index settings) • **indices:admin/mapping/put** (add or modify index mappings)

  Permission verification:

```
# Expected success (create an index)
PUT /log-v1
{
  "settings": { "number_of_shards": 3, "number_of_replicas": 1 }
}

# Expected success (modify index settings)
PUT /log-v1/_settings
{
  "index": { "number_of_replicas": 2 }
}

# Expected success (modify index mappings)
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
PUT /log-v1/_mapping
{
  "properties": { "new_field": { "type": "keyword" } }
}

# Expected success (write documents)
POST /log-v1/_doc
{ "user_id": "u001", "action": "login" }

# Expected success (query documents)
GET /log-v1/_search
{ "query": { "match_all": {} } }

# Expected success (delete a document)
DELETE /log-v1/_doc/1

# Expected success (delete an index)
DELETE /log-v1

# Expected failure (no cluster-level permissions)
GET /_cluster/health?pretty
```

- **Dashboard editing (read/write)**

  Scenario: BI developers or O&M engineers need to create and modify dashboards, visualizations, and index patterns in Kibana.

  User permission configuration: Map the user to roles **kibana_user** and **role_dashboard_editor**.

  Table 2-55 Permission configuration for dashboard editing

Configuration Item	Sub-Configuration Item	Description
Name	-	Role name: **role_dashboard_editor**
Cluster permissions	-	Not configured. No cluster-level permissions are required.
Index Permissions	Index	Target index name or wildcard, for example, **log-***.
	Index permissions	Select the predefined action groups **read** (document read) and **write** (document write).
Tenant Permissions	Tenant patterns	Leave this blank or specify a custom tenant name. By default, the role **kibana_user** has the **global_tenant** permission, so a custom role does not have to be configured here. To isolate tenant spaces, specify a custom tenant name, for example, **test_tenant**.
	Permissions	If a custom tenant is specified, select **Read and Write** to grant read/write permissions for dashboards in the tenant.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Permission verification:

Log in to Kibana using an account assigned these roles and switch the tenant to **Global**.

- You can view, create, and edit dashboards and visualizations.
- You can create new index patterns.
- You cannot access the **Security** page for permission management. (The **all_access** role is required to access this page.)

FAQ: What predefined roles are available in the system?

- **kibana_user**: Allows users to create and view dashboards in Kibana. This role has the **global_tenant** permissions by default.
- **readall**: Grants read-only access to all indexes. It is suitable for data analysts.
- **all_access**: Grants administrator permissions. Assign this role with caution.

On the **Roles** page, click **Learn more** to view descriptions of all predefined roles.

2.6.2 Configuring LDAP Authentication

In enterprise applications, user authentication is essential for ensuring data security and proper access control. However, traditional authentication methods often require repeated configuration for each application, which can be complex and error-prone. Using the Security plugin for Open Distro, CSS adds Active Directory as an authentication backend for clusters, connecting them seamlessly to the LDAP service. The Lightweight Directory Access Protocol (LDAP) is a lightweight version of the directory access protocol based on the X.500 standard, providing user authentication and authorization capabilities. This topic will describe how to enable LDAP authentication for a CSS Elasticsearch cluster to grant access to LDAP users of specific roles, and how to connect to a cluster using LDAP.

Constraints

Only Elasticsearch 7.10.2 security-mode clusters can be accessed through LDAP.

Preparations

- A security-mode Elasticsearch cluster has been created in CSS and its status is available.
- The LDAP service that is in the same VPC as the Elasticsearch cluster and the necessary user data have been prepared. For details, see **the OpenLDAP document: A Quick-Start Guide**.

Configuring LDAP Authentication

Step 1 Install an LDAP service on an ECS. If the LDAP service and user data have already been prepared, skip this step.

1. Create an ECS. The ECS must run a Windows OS and must be in the same VPC and security group as the security-mode Elasticsearch cluster of CSS. The Windows Server running on the ECS provides the built-in Active Directory service that supports the LDAP protocol.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

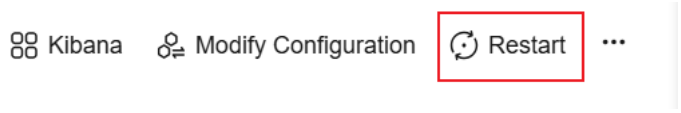
2. Log in to the ECS, and enable the Active Directory service. Create a domain, administrator, users, and user groups.

Step 2 Modify the parameter settings of the security-mode Elasticsearch cluster on CSS. Configure a static parameter in **elasticsearch.yml** to connect the cluster to the LDAP service.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Cluster Settings** > **Parameter Settings** > **Elasticsearch**, and click **Edit**. Expand **Custom**, and add the following parameter and value.
   - **Parameter**: opendistro_security.unsupported.restapi.allow_securityconfig_modification
   - **Value**: true
5. Click **Submit** above. In the displayed **Submit Configuration** dialog box, select the box that says "I understand that the modification will take effect after the cluster is restarted." and click **Yes**.

   If **Job Status** is **Succeeded** in the parameter settings task list below, the change has been saved.
6. After saving the change, click **Restart** in the upper-right corner of the page to restart the cluster, thus making the change take effect.

   Figure 2-12 Restarting a cluster

   - You need to restart the cluster after the change, or **Configuration not updated** will be displayed in the **Task Status** column in the cluster list.
   - If you restart the cluster after the change, and **Task Status** displays **Configuration error** in the cluster list, the parameter configuration file has failed to be modified.

Step 3 Configure a route for an Elasticsearch cluster on the CSS console to connect the cluster to the LDAP service.

1. On the cluster information page, click the **Overview** tab.
2. In the **Network Information** area, click **Add Route** under **Cluster Route**.
3. In the displayed dialog box, configure the route information.

   Table 2-56 Adding a route

Parameter	Description
IP Address	Enter the IP address of the LDAP server. If the LDAP service on the ECS is used, enter the IP address of the ECS.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Subnet Mask	Enter the subnet mask of the LDAP server. If the LDAP service on the ECS is used, enter the subnet mask of the ECS.

4. Click OK.

Step 4 Configure LDAP authentication for a security-mode Elasticsearch cluster.

1. On the cluster information page, click **Kibana** in the upper-right corner to log in to Kibana.
2. In the navigation pane on the left, choose **Dev Tools**.
3. Run the following commands to configure LDAP authentication.

   NOTE

   - Concepts used in an X.500 directory access protocol (including LDAP):
     - CN = Common Name
     - OU = Organizational Unit
     - DC = Domain Component
     - DN = Distinguished Name

     The CN, OU, and DC must be provided in the correct order. Otherwise, authentication will fail.
   - The configuration file consists of two parts: **authc** and **authz**.
     - **authc** (authentication): verifies whether a user is truly who they claim they are (password verification).
     - **authz** (authorization): verifies what the current user has access to.

```
PUT _opendistro/_security/api/securityconfig/config
{
   "dynamic": {
      "authc": {
         "basic_internal_auth_domain": {
            "description": "Authenticate via HTTP Basic against internal users database",
            "http_enabled": true,
            "transport_enabled": true,
            "order": 1,
            "http_authenticator": {
               "type": "basic",
               "challenge": true
            },
            "authentication_backend": {
               "type": "intern"
            }
         },
         "ldap": {
            "description": "Authenticate via LDAP or Active Directory",
            "http_enabled": true,
            "transport_enabled": true,
            "order": 2,
            "http_authenticator": {
               "type": "basic",
               "challenge": false
            },
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            "authentication_backend": {
               "type": "ldap",
               "config": {
                  "enable_ssl": false,
                  "enable_start_tls": false,
                  "enable_ssl_client_auth": false,
                  "verify_hostnames": true,
                  "hosts": ["10.0.XXX.XXX:389"],
                  "bind_dn": "CN=adminAD,DC=test,DC=ldap,DC=com",
                  "password": "<password>",
                  "userbase": "OU=ITDepartment,DC=test,DC=ldap,DC=com",
                  "usersearch": "(sAMAccountName={0})",
                  "username_attribute": "uid"
               }
            }
         }
      },
      "authz": {
         "roles_from_myldap": {
            "description": "Authorize via LDAP or Active Directory",
            "http_enabled": true,
            "transport_enabled": true,
            "authorization_backend": {
               "type": "ldap",
               "config": {
                  "enable_ssl": false,
                  "enable_start_tls": false,
                  "enable_ssl_client_auth": false,
                  "verify_hostnames": true,
                  "hosts": ["10.0.XXX.XXX:389"],
                  "bind_dn": "CN=adminAD,DC=test,DC=ldap,DC=com",
                  "password": "<password>",
                  "rolebase": "OU=groups,DC=test,DC=ldap,DC=com",
                  "rolesearch": "(member={0})",
                  "userroleattribute": null,
                  "userrolename": "disabled",
                  "rolename": "CN",
                  "resolve_nested_roles": true,
                  "userbase": "OU=ITDepartment,DC=test,DC=ldap,DC=com",
                  "usersearch": "(uid={0})"
               }
            }
         }
      }
   }
}
```

The parameters in **Table 2-57** need to be modified based on the actual environment.

Table 2-57 Parameter description

Parameter	Description
hosts	Address of the LDAP service. The port number is 389. If the LDAP service on the ECS is used, enter the IP address of the ECS.
bind_dn	It is similar to the LDAP user name (CN - OU - DC) and is used to access the LDAP server. Select a user name from the user data of the LDAP service.
password	Password of the LDAP user configured using **bind_dn**.

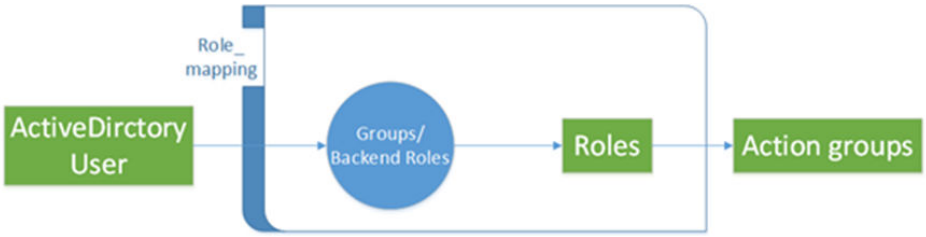
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
userbase	After the LDAP service is connected, the DN that the user belongs to is obtained. In this example, all user information in the **ITDepartment** directory is synchronized.
rolebase	The collection of permissions that can be configured for the **userbase** user group of the LDAP service.

Step 5 Configure the mapping between LDAP user permissions and Elasticsearch permissions in the Elasticsearch security-mode cluster to enable fine-grained access control.

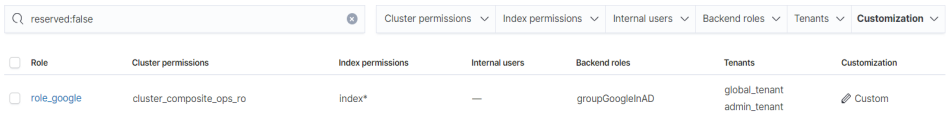
The rolebase permissions group of the LDAP server must be mapped to the roles in the Elasticsearch cluster. **Figure 2-13** illustrates the mapping. For details about the configuration, see **Creating Cluster Users and Granting Permissions**.

Figure 2-13 Permissions mapping



1. On the Kibana console, expand the menu in the upper-left corner, and choose **Security**. The **Security** page is displayed.
2. Click **Roles** to go to the Open Distro Security Roles page. Click **Create Role**, set **Name**, **Cluster Permissions**, **Index Permissions**, and **Tenant Permissions**. Then click **Save Role Definition** to save the role settings. The parameters are as follows:
   - Name (name of the role)
   - Cluster permissions
   - Index permissions
   - Tenant permissions
3. Click the newly created role, select **Mapped users**, enter a permissions group of the LDAP service in **Backend roles**, and click **Map**.
4. Check the configuration result.

Figure 2-14 Permissions mapping



5. Repeat **Step 5.2** to **Step 5.4** to map other permissions groups.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Step 6 Log in to Kibana using the LDAP user to verify the configuration.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column. Use the LDAP user to log in to the Kibana console.

   If the login is successful, the configuration is successful, and users can access the Elasticsearch cluster through LDAP. The specific permissions authorized are controlled by role permissions configured in Elasticsearch.

----End

2.7 Connecting to Clusters

2.7.1 Cluster Connection Methods

To ensure secure and efficient access to deployed Elasticsearch clusters, CSS provides multiple connection options tailored to different roles, programming languages, and network environments. At the application layer, CSS supports open-source RESTful APIs and SDKs for popular languages. It also includes built-in visualization and management tools such as Kibana and Cerebro. By following the instructions in this topic, you can quickly establish connections to your Elasticsearch clusters to access search and analytics capabilities both securely and efficiently.

Selecting the Appropriate Tools and Clients

Select an appropriate connection method based on your role and use scenario.

Table 2-58 Comparing different cluster connection methods

Purpose	Recommended Tool	When to Use	Advantage	Details
Visualization and management	Kibana	Data exploration, dashboards, cluster monitoring, development and debugging	Official visualization tool, comprehensive features, no coding required	**Connecting to a Cluster Through Kibana**
Cluster O&M	Cerebro	Cluster monitoring, index management, shard migration	Official visualization tool for cluster monitoring and operations	**Connecting to a Cluster Through Cerebro**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Purpose	Recommended Tool	When to Use	Advantage	Details
Application development	Java SDK	Connecting from a Java-based system	High performance; connection pool and high-availability retry supported	**Connecting to a Cluster Using the High Level REST Client** **Connecting to a Cluster Using the Low Level REST Client** **Connecting to a Cluster Through the Transport Client** **Connecting to a Cluster Through Spring Boot** **Connecting to a Cluster Through the Elasticsearch Java API Client**
	Python SDK	Connecting from a Python-based system		**Connecting to a Cluster Using Python**
	Go SDK	Connecting from a Go-based system		**Connecting to a Cluster Using Go**
	PHP SDK	Connecting from a PHP-based system		**Connecting to a Cluster Using PHP**
Data processing	MRS Hive (ES-Hadoop)	Offline big data analytics, exchange of Hive tables and Elasticsearch indexes	Seamless interconnection with the Hadoop ecosystem, suitable for mass data migration	**Connecting MRS Hive to a Cluster**
Lightweight debugging	cURL	Automated scripts, ad-hoc connectivity testing	Cross-platform access, no client installation	**Connecting to a Cluster Using cURL Commands**

Obtaining the Cluster Access Address

After the network is configured for a cluster (), to connect to it via a client, you must first obtain its access address. **Table 2-59** describes how to obtain the cluster address under different networking modes.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-59 Obtaining the cluster access address

Networking Mode	Operation Guide	Configuration Guide
Private network access	Connect to your cluster through the private network address of a directly connected node. This allows for low-cost, high-performance private network access. To obtain the private network address of one or more nodes in your cluster, perform the following steps: 1. Log in to the **CSS management console**. 2. In the navigation pane on the left, choose **Clusters > Elasticsearch**. 3. In the cluster list, obtain the target cluster's internal network address from the **Internal Network Address** column. Typical address format: *<host>:<port>* or *<host>:<port>,<host>:<port>*. Example: **10.62.179.32:9200,10.62.179.33:9200**. If the cluster has client nodes, only the IP addresses and ports of all the client nodes are displayed. Otherwise, the IP addresses and ports of all data nodes and cold data nodes are displayed.	Default
Public network access	Enable public network access for a security-mode cluster that uses HTTPS, and configure a whitelist for access control. To obtain the cluster's public network address, perform the following steps: 1. Log in to the **CSS management console**. 2. In the navigation pane on the left, choose **Clusters > Elasticsearch**. 3. In the cluster list, click the name of the target cluster. The cluster information page is displayed. 4. Click the **Overview** tab. In the **Network Information** area, find and record the IP address used for public network access.	**Configuring Public Network Access**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Networking Mode	Operation Guide	Configuration Guide
Dedicated load balancer	Bind a dedicated load balancer to your cluster for reliable load balancing. This mode supports two-way authentication via a custom certificate. To obtain the load balancer's private or public IP address, perform the following steps: 1. Log in to the **CSS management console**. 2. In the navigation pane on the left, choose **Clusters > Elasticsearch**. 3. In the cluster list, click the name of the target cluster. The cluster information page is displayed. 4. Choose **Cluster Access > Load Balancing**. On the **Elasticsearch** tab, find and record the private or public IP address of the target load balancer.	**Configuring a Dedicated Load Balancer**

2.7.2 Connecting to a Cluster Through Kibana

Kibana is the official data visualization and exploration platform for Elasticsearch. It enables in-depth data analysis and interactive visualization. In CSS, Kibana is pre-built for each Elasticsearch cluster. You can start Kibana in one click, without installing anything. Kibana offers a comprehensive suite of dashboard features and visualization tools. It seamlessly integrates Elasticsearch's analytical capabilities, supporting the full analytics process from data exploration to actionable business insights.

CSS's Kibana supports multiple access methods. Steps needed to log in to an Elasticsearch cluster vary depending on the access method you choose. See **Table 2-60**.

Table 2-60 Methods for logging in to an Elasticsearch cluster through Kibana

Kibana Access Method	Constraints	Details
One-click access to Kibana from the service console	N/A	**Logging In to an Elasticsearch Cluster by Accessing Kibana Through the Console**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Kibana Access Method	Constraints	Details
Accessing Kibana using a public IP address	• Only clusters in security mode support Kibana access through a public IP address. • Kibana public network access cannot be enabled for Elasticsearch security-mode clusters created before June 2020, that is, when the feature first became available.	**Logging In to an Elasticsearch Cluster by Accessing Kibana Using a Public IP Address**
Accessing Kibana using a private network address	The client that accesses Kibana must be able to reach Kibana's private IP address.	**Logging In to an Elasticsearch Cluster by Accessing Kibana Using a Private Network Address**

Constraints on Kibana Usage

- You can customize the username, role name, and tenant name in Kibana.

Logging In to an Elasticsearch Cluster by Accessing Kibana Through the Console

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters** > **Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
   - Non-security cluster: The Kibana console is displayed without asking for a username and password.
   - Security cluster: Enter the username and password on the login page and click **Log In** to log in to the Kibana console. The default username is **admin** and the password is the one specified during cluster creation.
4. After the login is successful, you can access the cluster through Kibana.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Logging In to an Elasticsearch Cluster by Accessing Kibana Using a Public IP Address

WARNING

- Only clusters in security mode support Kibana access through a public IP address.
- Kibana public network access cannot be enabled for Elasticsearch security-mode clusters created before June 2020, that is, when the feature first became available.
- The whitelist that controls Kibana public network access depends on whitelist support by the ELB service. After you update the whitelist, the new settings take effect immediately for new connections. For existing persistent connections using the IP addresses that have been removed from the whitelist, the new settings take effect in approximately 1 minute after these connections are disconnected.
- If you disable Kibana public network access and then re-enable it, the public IP address for accessing Kibana may change. Exercise caution.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Cluster Access** > **Kibana Public Network Access** to check whether Kibana public network access is enabled for the cluster.
   - Yes: Go to **6**.
   - No: Go to the next step.
5. Enable Kibana public network access for the Elasticsearch cluster.
   a. On the **Kibana Public Network Access** page, toggle on the **Kibana Public Network Access** button.
   b. In the displayed **Kibana Public Network Access** dialog box, set the parameters.

   Table 2-61 Configuring public network access for Kibana

Parameter	Description
Bandwidth	Bandwidth for accessing Kibana from the public network Value range: 1 Mbit/s to 200 Mbit/s

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

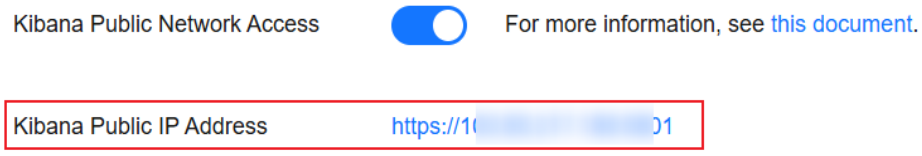
Parameter	Description
Configure Whitelist	Control Kibana public network access using a whitelist. • If a whitelist is configured, only IP addresses that are on this whitelist can access the cluster's Kibana console over the public network. Click **+ Add**. In the displayed text box, enter IP addresses or CIDR blocks that are allowed to access the cluster's Kibana console from the public network. Separate them using commas (,). Each value must be unique. An example of valid values: **192.168.1.1,10.0.0.0/24**. Examples of invalid values: **0.0.0.0**, **xx.xx.xx.xx/0**, **172.16.0.0-172.16.255.255**, non-standard formats (e.g., **192.168.1**), and duplicate values. • If no whitelist is configured, all public IP addresses can access the cluster's Kibana console. However, this can be a security risk and should be avoided.

c. Click **OK** to confirm the settings.

If Kibana public network access is already enabled, you can modify relevant settings.

6. After Kibana public network access is enabled, obtain the Kibana public IP address on the **Kibana Public Network Access** page.

Figure 2-15 Obtaining the Kibana public IP address



7. Enter the public IP address for Kibana in the browser address box to go to the Kibana login page.

   Enter the username and password on the login page and click **Log In** to log in to the Kibana console. The default username is **admin** and the password is the one specified during cluster creation.

8. After the login is successful, you can access the Elasticsearch cluster through Kibana.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Logging In to an Elasticsearch Cluster by Accessing Kibana Using a Private Network Address

CAUTION

The client that accesses Kibana must be able to reach Kibana's private IP address.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Select the **Overview** tab. In the **Network Information** area, record **Private IPv4 Address**.
5. Obtain Kibana's private IP address.

   By changing the port number in the cluster's private IP address from **9200** to **5601**, you get Kibana's private IP address. For example, if the cluster's private IPv4 address is **xx.xx.xx.xx:9200**, the private IP address of Kibana is **xx.xx.xx.xx:5601**.
6. (Optional) Configure a return route. To enable a client to access the cluster's Kibana console across different VPCs, configure a route for the Elasticsearch cluster.

   a. Connect the client and Kibana through a Direct Connect or VPC peering connection.

   b. Configure the route connecting the Elasticsearch cluster and the client.

      On the **Overview** tab, find **Cluster Route** in the **Network Information** area, and click **Add Route** under it. In the displayed dialog box, set **IP Address** and **Subnet Mask**.

   For details, see **Configuring Routes for a Cluster**.
7. Access Kibana.
   - If the security mode is enabled for an Elasticsearch cluster, the Kibana access address is **https://{Kibana private network address}**, for example, **https://xx.xx.xx.xx:5601**.
   - If the security mode is disabled, the Kibana access address is **http://{Kibana private network address}**, for example, **http://xx.xx.xx.xx:5601**.

Related Documents

- For details about how to use in-house developed Kibana to access Elasticsearch clusters in CSS, see **How Do I Connect In-house Developed Kibana to an Elasticsearch Cluster in CSS?**
- If Kibana cannot be launched, see **Failed to Open Kibana**.
- For routine O&M tasks (such as shard adjustment, index management, and performance monitoring), you are advised to use Cerebro for cluster login. For details, see **Connecting to a Cluster Through Cerebro**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2.7.3 Connecting to a Cluster Through Cerebro

Cerebro is an open-source Elasticsearch web admin tool, designed for real-time monitoring and efficient cluster operations. In CSS, Cerebro is pre-built for each Elasticsearch cluster. You can start Cerebro with one click, without installing anything. Cerebro allows you to quickly check a cluster's health status, node distribution, and index details. For routine O&M tasks, including shard adjustment, index management, and performance monitoring, we recommend using Cerebro. Its intuitive cluster topology views and comprehensive management features streamline O&M workflows, reducing complexity while improving efficiency. Cerebro is suitable for different roles, including developers, O&M teams, and data analysts.

Accessing a Cluster Through Cerebro on the CSS Console

The Elasticsearch clusters created in CSS provide a built-in Cerebro component, using which you can quickly access these clusters.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Cerebro** in the **Operation** column. The Cerebro login page is displayed.
   - Non-security cluster: Click the cluster name on the Cerebro login page to go to the Cerebro console.
   - Security cluster: Click the cluster name on the Cerebro login page, enter the username and password, and click **Authenticate** to go to the Cerebro console. The default username is **admin** and the password is the one specified during cluster creation.
4. After the login is successful, you can access clusters through Cerebro.

Accessing a Cluster Using an In-House Built Cerebro

CAUTION

When using in-house built Cerebro to access a cluster, make sure the network between them is connected.

1. Obtain the cluster access address. For details about how to obtain the cluster access address for different network configurations, see **Obtaining the Cluster Access Address**.
2. Start Cerebro, and enter the cluster access address.
   - For a cluster in security mode, enter https://*access address*:9200.

     To log in to a cluster in security mode, you need to provide a username and password.
   - For a cluster in non-security mode, enter http://*access address*:9200.

2.7.4 Connecting to a Cluster Using Java

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2.7.4.1 Connecting to a Cluster Using the High Level REST Client

Working directly with Elasticsearch's diverse APIs can be complex and inefficient. CSS simplifies this by providing a High Level REST Client, offering a more efficient way for querying and managing your data. The High Level REST Client encapsulates Elasticsearch APIs. You only need to construct the required request structures to access an Elasticsearch cluster. This simplifies the process of working with Elasticsearch clusters, as well as the development process. For details about how to use the REST Client, see **Java High Level REST Client**.

Prerequisites

- The target Elasticsearch cluster is available.
- The server that runs the Java code can communicate with the Elasticsearch cluster.
- Depending on the network configuration method used, obtain the cluster access address. For details, see **Obtaining the Cluster Access Address**.
- Java has been installed on the server and the JDK version is 1.8 or later. Download JDK 1.8 from **Java Downloads**.
- The High Level REST Client version has been confirmed. CSS allows you to connect to an Elasticsearch cluster using a Java client that has a later version. To ensure better compatibility, however, you are advised to use a Java client that has the same version as the target Elasticsearch cluster.

  If your High Level REST Client version is later than the Elasticsearch cluster version and there are incompatibility issues with some requests, you can use **RestHighLevelClient.getLowLevelClient()** to obtain the Low Level REST Client and customize Elasticsearch requests. For details, see **Connecting to a Cluster Using the Low Level REST Client**.

Introducing Dependencies

Import Java dependencies on the server that runs the Java code using either of the following ways:

- Maven:

  Replace 7.10.2 with the actual Java client version.

  ```
  <dependency>
      <groupId>org.elasticsearch.client</groupId>
      <artifactId>elasticsearch-rest-high-level-client</artifactId>
      <version>7.10.2</version>
  </dependency>
  <dependency>
      <groupId>org.elasticsearch</groupId>
      <artifactId>elasticsearch</artifactId>
      <version>7.10.2</version>
  </dependency>
  ```

- Gradle:

  Replace 7.10.2 with the actual Java client version.

  ```
  compile group: 'org.elasticsearch.client', name: 'elasticsearch-rest-high-level-client', version: '7.10.2'
  ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Connecting to a Cluster

The sample code varies depending on the security mode settings of the target Elasticsearch cluster. Select the right reference document based on your service scenario.

Table 2-62 Cluster access scenarios

Elasticsearch Cluster Security-Mode Settings	Whether to Load a Security Certificate	Details
Non-security mode	-	**Connecting to a Non-Security Mode Cluster Using the High Level REST Client**
Security mode + HTTP Security mode + HTTPS	No	**Connecting to a Security-Mode Cluster Using the High Level REST Client (Without a Certificate)**
Security mode + HTTPS	Yes	**Connecting to a Security-Mode Cluster Using the High Level REST Client (With a Certificate)**

Connecting to a Non-Security Mode Cluster Using the High Level REST Client

Use the High Level REST Client to connect to an Elasticsearch cluster for which the security mode is disabled, and query whether the **test** index exists. The sample code is as follows:

```
import org.apache.http.HttpHost;
import org.elasticsearch.client.RequestOptions;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.client.indices.GetIndexRequest;

import java.io.IOException;
import java.util.Arrays;
import java.util.List;

/**
 * Use the High Level REST Client to connect to a non-security mode cluster.
 */
public class Main {
    public static void main(String[] args) throws IOException {
    List<String> host = Arrays.asList("{Cluster access address}");
        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, 9200, "http"));
        final RestHighLevelClient client = new RestHighLevelClient(builder);
        GetIndexRequest indexRequest = new GetIndexRequest("test");
        boolean exists = client.indices().exists(indexRequest, RequestOptions.DEFAULT);
        System.out.println(exists);
        client.close();
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }
}
```

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the High Level REST Client (Without a Certificate)

Use the High Level REST Client to connect to a security-mode Elasticsearch cluster (HTTP or HTTPS) without loading a security certificate, and query whether the **test** index exists. The sample code is as follows:

```
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthRequest;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthResponse;
import org.elasticsearch.client.RequestOptions;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.client.indices.GetIndexRequest;

import java.io.IOException;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.concurrent.TimeUnit;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;

/**
 * Use the High Level REST Client to connect to a security-mode cluster (without a certificate).
 */
public class Main {

    private static final Logger logger = LogManager.getLogger(Main.class);

    /**
     * Create a class for the client. Define the create function.
     */
    public static RestHighLevelClient create(List<String> host, int port, String protocol, int connectTimeout,
        int connectionRequestTimeout, int socketTimeout, String username, String password) throws
IOException {

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientBuilder -> {
```

```
            // enable user authentication
            final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
            credentialsProvider.setCredentials(AuthScope.ANY, new
UsernamePasswordCredentials(username, password));
            httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
            // set keepalive
            httpClientBuilder.setKeepAliveStrategy(((httpResponse, httpContext) ->
TimeUnit.MINUTES.toMinutes(10)));
            // enable SSL / TLS
            SSLContext sc = null;
            try {
                sc = SSLContext.getInstance("SSL");
                sc.init(null, trustAllCerts, new SecureRandom());
            } catch (KeyManagementException | NoSuchAlgorithmException e) {
                e.printStackTrace();
            }
            SSLIOSessionStrategy sslStrategy = new SSLIOSessionStrategy(sc, new NoopHostnameVerifier());
            httpClientBuilder.setSSLStrategy(sslStrategy);

            return httpClientBuilder;
        });
    final RestHighLevelClient client = new RestHighLevelClient(builder);
    logger.info("es rest client build success {} ", client);

    ClusterHealthRequest request = new ClusterHealthRequest();
    ClusterHealthResponse response = client.cluster().health(request, RequestOptions.DEFAULT);
    System.out.println("es rest client health response {} " + response);
    return client;
  }

  /**
   * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
   */
  public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
    return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
  }

  /**
   * Configure trustAllCerts to ignore the certificate configuration.
   */
  public 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;
      }
    }
  };

  /**
* The following is an example of the main function. Call the create function to create a client and check
whether the test index exists.
   */
  public static void main(String[] args) throws IOException {
    RestHighLevelClient client = create(Arrays.asList("{host}"), 9200, "https", 1000, 1000, 1000,
"username", "password");
    GetIndexRequest indexRequest = new GetIndexRequest("test");
    boolean exists = client.indices().exists(indexRequest, RequestOptions.DEFAULT);
    System.out.println(exists);
```

```
        client.close();
    }
}
```

Table 2-63 Variables

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol, which can be **http** or **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the High Level REST Client (With a Certificate)

Use the High Level REST Client to connect to a security-mode Elasticsearch cluster that uses HTTPS with a security certificate loaded, and query whether the **test** index exists. The sample code is as follows:

CAUTION

For how to obtain and upload a security certificate, see **Obtaining and Uploading a Security Certificate**.

```
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthRequest;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import org.elasticsearch.action.admin.cluster.health.ClusterHealthResponse;
import org.elasticsearch.client.RequestOptions;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.client.indices.GetIndexRequest;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.concurrent.TimeUnit;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;

/**
 * Use the Hive Level REST Client to connect to a security-mode cluster (with an HTTPS certificate).
 */
public class Main {
    public static RestHighLevelClient create(List<String> host, int port, String protocol, int connectTimeout,
        int connectionRequestTimeout, int socketTimeout, String username, String password, String
certFilePath,
        String certPassword) throws IOException {

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientBuilder -> {
                // enable user authentication
                final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
                credentialsProvider.setCredentials(AuthScope.ANY, new
UsernamePasswordCredentials(username, password));
                httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
                // set keepalive
                httpClientBuilder.setKeepAliveStrategy(((httpResponse, httpContext) ->
TimeUnit.MINUTES.toMinutes(10)));
                // enable SSL / TLS
                SSLContext sc = null;
                try {
                    TrustManager[] tm = {new MyX509TrustManager(certFilePath, certPassword)};
                    sc = SSLContext.getInstance("SSL", "SunJSSE");
                   //You can also use SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
                    sc.init(null, tm, new SecureRandom());
                } catch (Exception e) {
                    e.printStackTrace();
                }
                SSLIOSessionStrategy sslStrategy = new SSLIOSessionStrategy(sc, new NoopHostnameVerifier());
                httpClientBuilder.setSSLStrategy(sslStrategy);

                return httpClientBuilder;
            });
        final RestHighLevelClient client = new RestHighLevelClient(builder);
        logger.info("es rest client build success {} ", client);

        ClusterHealthRequest request = new ClusterHealthRequest();
        ClusterHealthResponse response = client.cluster().health(request, RequestOptions.DEFAULT);
        logger.info("es rest client health response {} ", response);
        return client;
    }
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }

    private static final Logger logger = LogManager.getLogger(Main.class);

    public static class MyX509TrustManager implements X509TrustManager {
        X509TrustManager sunJSSEX509TrustManager;

        MyX509TrustManager(String certFilePath, String certPassword) throws Exception {
            File file = new File(certFilePath);
            if (!file.isFile()) {
                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, certPassword.toCharArray());
            TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
            tmf.init(ks);
            TrustManager[] tms = tmf.getTrustManagers();
            for (TrustManager tm : tms) {
                if (tm instanceof X509TrustManager) {
                    sunJSSEX509TrustManager = (X509TrustManager) tm;
                    return;
                }
            }
            throw new Exception("Couldn't initialize");
        }

        @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 new X509Certificate[0];
        }
    }

    /**
* The following is an example of the main function. Call the create function to create a client and check whether the test index exists.
     */
    public static void main(String[] args) throws IOException {
        RestHighLevelClient client = create(Arrays.asList({host}), 9200, "https", 1000, 1000, 1000, "username", "password", "certFilePath", "certPassword");
        GetIndexRequest indexRequest = new GetIndexRequest("test");
        boolean exists = client.indices().exists(indexRequest, RequestOptions.DEFAULT);
        System.out.println(exists);
        client.close();
    }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-64 Function parameters

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol. Set this parameter to **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.
certFilePath	Path for storing the security certificate.
certPassword	Password of the security certificate.

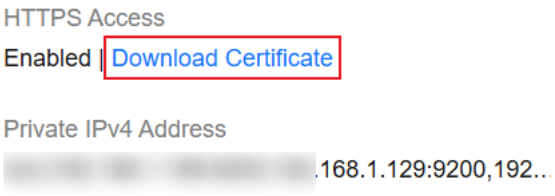
This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

Obtaining and Uploading a Security Certificate

To access a security-mode Elasticsearch cluster that uses HTTPS, perform the following steps to obtain the security certificate if it is required, and upload it to the client.

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

   Figure 2-16 Downloading a security certificate

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. Convert the security certificate **CloudSearchService.cer**. Upload the downloaded security certificate to the client and use keytool to convert the **.cer** certificate into a **.jks** certificate that can be read by Java.
   - In Linux, run the following command to convert the certificate:
     ```
     keytool -import -alias newname -keystore ./truststore.jks -file ./CloudSearchService.cer
     ```
   - In Windows, run the following command to convert the certificate:
     ```
     keytool -import -alias newname -keystore .\truststore.jks -file .\CloudSearchService.cer
     ```

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

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

2.7.4.2 Connecting to a Cluster Using the Low Level REST Client

When querying and managing data in Elasticsearch, sometimes you may find the High Level REST Client inadequate. For instance, due to inherent limitations, the High Level REST Client may not be able to execute complex custom requests. In this case, you may consider using the Low Level REST Client, which encapsulates Elasticsearch APIs. You only need to construct the required request structures to access an Elasticsearch cluster. This simplifies the process of working with Elasticsearch clusters. The Low Level REST Client allows you to customize the request structure, which is more flexible and supports all the request formats of Elasticsearch, such as GET, POST, DELETE, and HEAD.

You can use the Low Level REST Client to access an Elasticsearch cluster in either of the following ways:

- Create the Low Level Rest Client directly.
- Create the High Level Rest Client and then call getLowLevelClient() to obtain the Low Level Rest Client. That is, use **RestHighLevelClient.getLowLevelClient()** to obtain the Low Level Rest Client.

How do I determine when to use which method? If you need to execute highly customized requests, create the Low Level Rest Client directly. If you are already using the High Level Rest Client, you can call the getLowLevelClient() method to obtain the Low Level Rest Client. This simplifies your code.

Prerequisites

- The target Elasticsearch cluster is available.
- The server that runs the Java code can communicate with the Elasticsearch cluster.
- Depending on the network configuration method used, obtain the cluster access address. For details, see **Obtaining the Cluster Access Address**.
- Java has been installed on the server and the JDK version is 1.8 or later. Download JDK 1.8 from **Java Downloads**.
- The Low Level REST Client version has been confirmed. CSS allows you to connect to an Elasticsearch cluster using a Java client that has a later version. To ensure better compatibility, however, you are advised to use a Java client that has the same version as the target Elasticsearch cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Introducing Dependencies

Introduce the required Java dependencies on the server where you run Java code. Declare the Apache version in Maven mode.

Replace 7.10.2 with the actual Java client version.

```
<dependency>
    <groupId>org.elasticsearch.client</groupId>
    <artifactId>elasticsearch-rest-high-level-client</artifactId>
    <version>7.10.2</version>
</dependency>
<dependency>
    <groupId>org.elasticsearch</groupId>
    <artifactId>elasticsearch</artifactId>
    <version>7.10.2</version>
</dependency>
```

Connecting to a Cluster

The sample code varies depending on the security mode settings of the target Elasticsearch cluster. Select the right reference document based on your service scenario.

Table 2-65 Cluster access scenarios

How Low Level REST Client Is Created	Elasticsearch Cluster Security-Mode Settings	Whether to Load a Security Certificate	Details
Directly create the Low Level REST Client	Non-security mode	-	**Connecting to a Non-Security Mode Cluster Using the Low Level REST Client**
	Security mode + HTTP Security mode + HTTPS	No	**Connecting to a Security-Mode Cluster Using the Low Level REST Client (Without a Certificate)**
	Security mode + HTTPS	Yes	**Connecting to a Security-Mode Cluster Using the Low Level REST Client (With a Certificate)**
Create the High Level REST Client first and then call getLowLevelClient() to obtain the Low Level REST Client	Non-security mode	-	**Connecting to a Non-Security Mode Cluster Using the High Level REST Client**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

How Low Level REST Client Is Created	Elasticsearch Cluster Security-Mode Settings	Whether to Load a Security Certificate	Details
	Security mode + HTTP Security mode + HTTPS	No	**Connecting to a Security-Mode Cluster Using the High Level REST Client (Without a Certificate)**
	Security mode + HTTPS	Yes	**Connecting to a Security-Mode Cluster Using the High Level REST Client (With a Certificate)**

Connecting to a Non-Security Mode Cluster Using the Low Level REST Client

Use the Low Level REST Client to connect to an Elasticsearch cluster for which the security mode is disabled, and query whether the **test** index exists. The sample code is as follows:

```
import org.apache.http.HttpHost;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;

import java.io.IOException;
import java.util.Arrays;
import java.util.List;

public class Main {

    public static void main(String[] args) throws IOException {
    List<String> host = Arrays.asList("{Cluster access address}");
        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, 9200, "http"));
        /**
         *Create the Low Level Rest Client.
         */
        RestClient lowLevelClient = builder.build();
        /**
         * Check whether the test index exists. If the index exists, 200 is returned. If the index does not exist, 404 is returned.
         */
        Request request = new Request("HEAD", "/test");
        Response response = lowLevelClient.performRequest(request);
        System.out.println(response.getStatusLine().getStatusCode());
        lowLevelClient.close();
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
    }
}
```

This piece of code checks whether the **test** index exists in the cluster. If **200** (the index exists) or **404** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the Low Level REST Client (Without a Certificate)

Use the Low Level REST Client to connect to a security-mode Elasticsearch cluster (HTTP or HTTPS) without loading a security certificate, and query whether the **test** index exists. The sample code is as follows:

```
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;

import java.io.IOException;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.concurrent.TimeUnit;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;

public class Main {

    private static final Logger logger = LogManager.getLogger(Main.class);

    /**
     * Create a class for the client. Define the create function.
     */
    public static RestClient create(List<String> host, int port, String protocol, int connectTimeout,
        int connectionRequestTimeout, int socketTimeout, String username, String password) throws
IOException {

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientBuilder -> {
                // enable user authentication
                if (username != null && password != null) {
                    final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
                    credentialsProvider.setCredentials(AuthScope.ANY,
                        new UsernamePasswordCredentials(username, password));
                    httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
                }
                // set keepalive
                httpClientBuilder.setKeepAliveStrategy(((httpResponse, httpContext) ->
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
TimeUnit.MINUTES.toMinutes(10)));
                // enable SSL / TLS
                SSLContext sc = null;
                try {
                    sc = SSLContext.getInstance("SSL");
                    sc.init(null, trustAllCerts, new SecureRandom());
                } catch (KeyManagementException | NoSuchAlgorithmException e) {
                    e.printStackTrace();
                }
                SSLIOSessionStrategy sslStrategy = new SSLIOSessionStrategy(sc, new NoopHostnameVerifier());
                httpClientBuilder.setSSLStrategy(sslStrategy);
                return httpClientBuilder;
            });
        final RestClient client = builder.build();
        logger.info("es rest client build success {} ", client);
        return client;
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }

    /**
     * Configure trustAllCerts to ignore the certificate configuration.
     */
    public 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;
            }
        }
    };

    /**
     * The following is an example of the main function. Call the create function to create the Low Level
REST Client and check whether the test index exists.
     */
    public static void main(String[] args) throws IOException {
        RestClient lowLevelClient = create(Arrays.asList("{Cluster access address}"), 9200, "http", 1000, 1000,
1000, "username",
            "password");
        Request request = new Request("HEAD", "/test");
        Response response = lowLevelClient.performRequest(request);
        System.out.println(response.getStatusLine().getStatusCode());
        lowLevelClient.close();
    }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-66 Variables

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol, which can be **http** or **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.

This piece of code checks whether the **test** index exists in the cluster. If **200** (the index exists) or **404** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the Low Level REST Client (With a Certificate)

Use the Low Level REST Client to connect to a security-mode Elasticsearch cluster that uses HTTPS with a security certificate loaded, and query whether the **test** index exists. The sample code is as follows:

CAUTION

For how to obtain and upload a security certificate, see **Obtaining and Uploading a Security Certificate**.

```
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;

import java.io.File;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.concurrent.TimeUnit;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;

public class Main {

    private static final Logger logger = LogManager.getLogger(Main.class);

    /**
     * Create a class for the client. Define the create function.
     */
    public static RestClient create(List<String> host, int port, String protocol, int connectTimeout,
        int connectionRequestTimeout, int socketTimeout, String username, String password, String
certFilePath,
        String certPassword) throws IOException {
        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientBuilder -> {
                // enable user authentication
                if (username != null && password != null) {
                    final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
                    credentialsProvider.setCredentials(AuthScope.ANY,
                        new UsernamePasswordCredentials(username, password));
                    httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
                }
                // set keepalive
                httpClientBuilder.setKeepAliveStrategy(((httpResponse, httpContext) ->
TimeUnit.MINUTES.toMinutes(10)));
                // enable SSL / TLS
                SSLContext sc = null;
                try {
                    TrustManager[] tm = {new MyX509TrustManager(certFilePath, certPassword)};
                    sc = SSLContext.getInstance("SSL", "SunJSSE");
                   //You can also use SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
                    sc.init(null, tm, new SecureRandom());
                } catch (Exception e) {
                    e.printStackTrace();
                }
                SSLIOSessionStrategy sslStrategy = new SSLIOSessionStrategy(sc, new NoopHostnameVerifier());
                httpClientBuilder.setSSLStrategy(sslStrategy);
                return httpClientBuilder;
            });
        final RestClient client = builder.build();
        logger.info("es rest client build success {} ", client);
        return client;
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }

    public static class MyX509TrustManager implements X509TrustManager {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
    X509TrustManager sunJSSEX509TrustManager;

    MyX509TrustManager(String certFilePath, String certPassword) throws Exception {
        File file = new File(certFilePath);
        if (!file.isFile()) {
            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, certPassword.toCharArray());
        TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
        tmf.init(ks);
        TrustManager[] tms = tmf.getTrustManagers();
        for (TrustManager tm : tms) {
            if (tm instanceof X509TrustManager) {
                sunJSSEX509TrustManager = (X509TrustManager) tm;
                return;
            }
        }
        throw new Exception("Couldn't initialize");
    }

    @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 new X509Certificate[0];
    }
  }

  /**
   * The following is an example of the main function. Call the create function to create the Low Level
REST Client and check whether the test index exists.
   */
  public static void main(String[] args) throws IOException {
      RestClient lowLevelClient = create(Arrays.asList("{Cluster access address}"), 9200, "https", 1000, 1000,
1000, "username",
          "password", "certFilePath", "certPassword");
      Request request = new Request("HEAD", "test");
      Response response = lowLevelClient.performRequest(request);
      System.out.println(response.getStatusLine().getStatusCode());
      lowLevelClient.close();
  }
}
```

Table 2-67 Function parameters

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
protocol	Connection protocol. Set this parameter to **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.
certFilePath	Path for storing the security certificate.
certPassword	Password of the security certificate.

This piece of code checks whether the **test** index exists in the cluster. If **200** (the index exists) or **404** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Non-Security Mode Cluster Using the High Level REST Client

Use the High Level REST Client to obtain the Low Level REST Client by calling **getLowLevelClient()**, use the low-level client to connect to an Elasticsearch cluster for which the security mode is disabled, and query whether the **test** index exists. The sample code is as follows:

```
import org.apache.http.HttpHost;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;

import java.io.IOException;
import java.util.Arrays;
import java.util.List;

public class Main {

    public static void main(String[] args) throws IOException {
    List<String> host = Arrays.asList("{Cluster access address}");
        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, 9200, "http"));
        final RestHighLevelClient restHighLevelClient = new RestHighLevelClient(builder);
        /**
         * Create a High Level Rest Client and then call getLowLevelClient() to obtain the Low Level Rest Client. The code differs from the client creation code only in the following line:
         */
        final RestClient lowLevelClient = restHighLevelClient.getLowLevelClient();
        /**
         * Check whether the test index exists. If the index exists, 200 is returned. If the index does not exist, 404 is returned.
         */
        Request request = new Request("HEAD", "/test");
        Response response = lowLevelClient.performRequest(request);
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        System.out.println(response.getStatusLine().getStatusCode());
        lowLevelClient.close();
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }
}
```

This piece of code checks whether the **test** index exists in the cluster. If **200** (the index exists) or **404** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the High Level REST Client (Without a Certificate)

Use the High Level REST Client to obtain the Low Level REST Client by calling **getLowLevelClient()**, use the low-level client to connect to a security-mode Elasticsearch cluster that uses HTTP or HTTPS without loading a security certificate, and query whether the **test** index exists. The sample code is as follows:

```
import org.apache.http.HttpHost;
import org.apache.http.HttpResponse;
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.impl.client.DefaultConnectionKeepAliveStrategy;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.http.protocol.HttpContext;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.common.Nullable;

import java.io.IOException;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;
import java.util.concurrent.TimeUnit;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;import javax.net.ssl.X509TrustManager;

import org.elasticsearch.client.RestHighLevelClient;

public class Main13 {

    /**
     * Create a class for the client. Define the create function.
     */
    public static RestHighLevelClient create(List<String> host, int port, String protocol, int connectTimeout,
```

```
int connectionRequestTimeout, int socketTimeout,  String username, String password) throws IOException {

        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 | NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new NullHostNameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
            credentialsProvider);

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientConfigCallback);
        final RestHighLevelClient client = new RestHighLevelClient(builder);
        logger.info("es rest client build success {} ", client);
        return client;
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }

    /**
     * Configure trustAllCerts to ignore the certificate configuration.
     */
    public 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;
            }
        }
    };

    /**
     * The CustomConnectionKeepAliveStrategy function is used to set the connection keepalive when there
are a large number of short connections or when there are not many data requests.
     */
    public static class CustomConnectionKeepAliveStrategy extends DefaultConnectionKeepAliveStrategy {
        public static final CustomConnectionKeepAliveStrategy INSTANCE = new
CustomConnectionKeepAliveStrategy();

        private CustomConnectionKeepAliveStrategy() {
            super();
        }

        /**
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        * Maximum keepalive time (in minutes)
        * The default value is 10 minutes. You can set it based on the number of TCP connections in
TIME_WAIT state. If there are too many TCP connections, you can increase this value.
        */
       private final long MAX_KEEP_ALIVE_MINUTES = 10;

       @Override
       public long getKeepAliveDuration(HttpResponse response, HttpContext context) {
          long keepAliveDuration = super.getKeepAliveDuration(response, context);
          // <0 indicates an unlimited keepalive period.
          // Change the period from unlimited to a default period.
          if (keepAliveDuration < 0) {
             return TimeUnit.MINUTES.toMillis(MAX_KEEP_ALIVE_MINUTES);
          }
          return keepAliveDuration;
       }
    }

    private static final Logger logger = LogManager.getLogger(Main.class);

    static class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
       @Nullable
       private final CredentialsProvider credentialsProvider;
       /**
        * The {@link SSLIOSessionStrategy} for all requests to enable SSL / TLS encryption.
        */
       private final SSLIOSessionStrategy sslStrategy;
       /**
        * Create a new {@link SecuredHttpClientConfigCallback}.
        *
        * @param credentialsProvider The credential provider, if a username/password have been supplied
        * @param sslStrategy         The SSL strategy, if SSL / TLS have been supplied
        * @throws NullPointerException if {@code sslStrategy} is {@code null}
        */
       SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
          @Nullable final CredentialsProvider credentialsProvider) {
          this.sslStrategy = Objects.requireNonNull(sslStrategy);
          this.credentialsProvider = credentialsProvider;
       }
       /**
        * Get the {@link CredentialsProvider} that will be added to the HTTP client.
        *
        * @return Can be {@code null}.
        */
       @Nullable
       CredentialsProvider getCredentialsProvider() {
          return credentialsProvider;
       }
       /**
        * Get the {@link SSLIOSessionStrategy} that will be added to the HTTP client.
        *
        * @return Never {@code null}.
        */
       SSLIOSessionStrategy getSSLStrategy() {
          return sslStrategy;
       }
       /**
        * Sets the {@linkplain HttpAsyncClientBuilder#setDefaultCredentialsProvider(CredentialsProvider)
credential provider},
        *
        * @param httpClientBuilder The client to configure.
        * @return Always {@code httpClientBuilder}.
        */
       @Override
       public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder httpClientBuilder) {
          // enable SSL / TLS
          httpClientBuilder.setSSLStrategy(sslStrategy);
          // enable user authentication
          if (credentialsProvider != null) {
```

```
                httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
            }
            return httpClientBuilder;
        }
    }

    public static class NullHostNameVerifier implements HostnameVerifier {
        @Override
        public boolean verify(String arg0, SSLSession arg1) {
            return true;
        }
    }

    /**
* The following is an example of the main function. Call the create function to create a high-level client, call the getLowLevelClient() function to obtain a low-level client, and check whether the test index exists.
     */
    public static void main(String[] args) throws IOException {
        RestHighLevelClient client = create(Arrays.asList("{Cluster access address}") 9200, "http", 1000, 1000, 1000, "username", "password");
        RestClient lowLevelClient = client.getLowLevelClient();
        Request request = new Request("HEAD", "test");
        Response response = lowLevelClient.performRequest(request);
        System.out.println(response.getStatusLine().getStatusCode());
        lowLevelClient.close();
    }
}
```

Table 2-68 Variables

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol, which can be **http** or **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.

This piece of code checks whether the **test** index exists in the cluster. If **200** (the index exists) or **404** (the index does not exist) is returned, it indicates that the cluster is connected.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Connecting to a Security-Mode Cluster Using the High Level REST Client (With a Certificate)

Use the High Level REST Client to obtain the Low Level REST Client by calling **getLowLevelClient()**, use the low-level client to connect to a security-mode Elasticsearch cluster that uses HTTPS with a security certificate loaded, and query whether the **test** index exists. The sample code is as follows:

CAUTION

For how to obtain and upload a security certificate, see **Obtaining and Uploading a Security Certificate**.

```
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthRequest;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthResponse;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.RequestOptions;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.common.Nullable;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;

import javax.net.ssl.SSLContext;import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;

public class Main {

   private static final Logger logger = LogManager.getLogger(Main.class);

   /**
    * Create a class for the client. Define the create function.
    */
   public static RestHighLevelClient create(List<String> host, int port, String protocol, int connectTimeout,
int connectionRequestTimeout, int socketTimeout, String username, String password, String certFilePath,
String certPassword) throws IOException {
       final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
       credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials(username,
password));
       SSLContext sc = null;
       try {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            TrustManager[] tm = {new MyX509TrustManager(certFilePath, certPassword)};
            sc = SSLContext.getInstance("SSL", "SunJSSE");
            //You can also use SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
            sc.init(null, tm, new SecureRandom());
        } catch (Exception e) {
            e.printStackTrace();
        }

        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new NoopHostnameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
            credentialsProvider);

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientConfigCallback);
        final RestHighLevelClient client = new RestHighLevelClient(builder);
        logger.info("es rest client build success {} ", client);

        ClusterHealthRequest request = new ClusterHealthRequest();
        ClusterHealthResponse response = client.cluster().health(request, RequestOptions.DEFAULT);
        logger.info("es rest client health response {} ", response);
        return client;
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);}

    static class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
        @Nullable
        private final CredentialsProvider credentialsProvider;

        private final SSLIOSessionStrategy sslStrategy;

        SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
            @Nullable final CredentialsProvider credentialsProvider) {
            this.sslStrategy = Objects.requireNonNull(sslStrategy);
            this.credentialsProvider = credentialsProvider;
        }

        @Nullable
        CredentialsProvider getCredentialsProvider() {
            return credentialsProvider;
        }

        SSLIOSessionStrategy getSSLStrategy() {
            return sslStrategy;
        }

        @Override
        public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder httpClientBuilder) {
            httpClientBuilder.setSSLStrategy(sslStrategy);
            if (credentialsProvider != null) {
                httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
            }
            return httpClientBuilder;
        }}

    public static class MyX509TrustManager implements X509TrustManager {
        X509TrustManager sunJSSEX509TrustManager;

        MyX509TrustManager(String certFilePath, String certPassword) throws Exception {
            File file = new File(certFilePath);
            if (!file.isFile()) {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
                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, certPassword.toCharArray());
            TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
            tmf.init(ks);
            TrustManager[] tms = tmf.getTrustManagers();
            for (TrustManager tm : tms) {
                if (tm instanceof X509TrustManager) {
                    sunJSSEX509TrustManager = (X509TrustManager) tm;
                    return;
                }
            }
            throw new Exception("Couldn't initialize");
        }

        @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 new X509Certificate[0];
        }
    }

    /**
* The following is an example of the main function. Call the create function to create a high-level client, call the getLowLevelClient() function to obtain a low-level client, and check whether the test index exists.
     */
    public static void main(String[] args) throws IOException {
        RestHighLevelClient client = create(Arrays.asList("{Cluster access address}", 9200, "https", 1000, 1000, 1000, "username", "password", "certFilePath", "certPassword");
        RestClient lowLevelClient = client.getLowLevelClient();
        Request request = new Request("HEAD", "test");
        Response response = lowLevelClient.performRequest(request);
        System.out.println(response.getStatusLine().getStatusCode());
        lowLevelClient.close();
    }
}
```

Table 2-69 Function parameters

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol. Set this parameter to **https**.
connectTimeout	Connection timeout (in ms).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.
certFilePath	Path for storing the security certificate.
certPassword	Password of the security certificate.

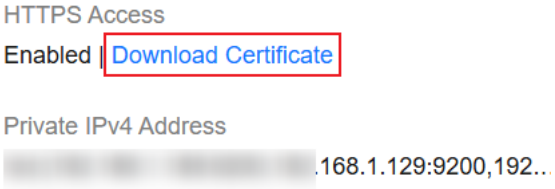
This piece of code checks whether the **test** index exists in the cluster. If **200** (the index exists) or **404** (the index does not exist) is returned, it indicates that the cluster is connected.

Obtaining and Uploading a Security Certificate

To access a security-mode Elasticsearch cluster that uses HTTPS, perform the following steps to obtain the security certificate if it is required, and upload it to the client.

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

   Figure 2-17 Downloading a security certificate

2. Convert the security certificate **CloudSearchService.cer**. Upload the downloaded security certificate to the client and use keytool to convert the **.cer** certificate into a **.jks** certificate that can be read by Java.
   - In Linux, run the following command to convert the certificate:
     ```
     keytool -import -alias newname -keystore ./truststore.jks -file ./CloudSearchService.cer
     ```
   - In Windows, run the following command to convert the certificate:
     ```
     keytool -import -alias newname -keystore .\truststore.jks -file .\CloudSearchService.cer
     ```

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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

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

2.7.4.3 Connecting to a Cluster Through the Transport Client

When using Elasticsearch for data query and management, many users rely on the Java-based Transport Client that simplifies interaction with Elasticsearch clusters by encapsulating native APIs. You can work with CSS Elasticsearch clusters through the Transport Client, for which you only need to construct the required structures.

CAUTION

The Transport Client has been marked deprecated in Elasticsearch 7.0. To ensure better compatibility and feature support, you are advised to use the Java High Level REST Client.

Prerequisites

- The target Elasticsearch cluster is available and the security mode is disabled for it. (The Transport Client can only be used to access non-security mode clusters.)
- The server that runs the Java code can communicate with the Elasticsearch cluster.
- The cluster's private network address has been obtained. For details, see **Obtaining the Cluster Access Address**. Given that the Transport Client can only be used to access non-security mode clusters, it can only do so through a private network address.
- Java has been installed on the server and the JDK version is 1.8 or later. Download JDK 1.8 from **Java Downloads**.
- The Transport Client version has been confirmed. To ensure better compatibility, use a Transport Client that has the same version as the target Elasticsearch cluster.

Introducing Dependencies

Introduce the required Java dependencies on the server where you run Java code. Declare the Apache version in Maven mode.

Replace 7.10.2 with the actual Java client version.

```
<dependency>
   <groupId>org.elasticsearch.client</groupId>
   <artifactId>transport</artifactId>
   <version>7.10.2</version>
</dependency>
<dependency>
   <groupId>org.elasticsearch</groupId>
   <artifactId>elasticsearch</artifactId>
   <version>7.10.2</version>
</dependency>
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Connecting to a Cluster

Use the Transport Client to connect to an Elasticsearch cluster for which the security mode is disabled, and query whether the **test** index exists. The sample code is as follows:

```
import org.elasticsearch.action.ActionFuture;
import org.elasticsearch.action.admin.indices.exists.indices.IndicesExistsRequest;
import org.elasticsearch.action.admin.indices.exists.indices.IndicesExistsResponse;
import org.elasticsearch.client.transport.TransportClient;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.transport.TransportAddress;
import org.elasticsearch.transport.client.PreBuiltTransportClient;

import java.net.InetAddress;
import java.net.UnknownHostException;
import java.util.concurrent.ExecutionException;

public class Main {
    public static void main(String[] args) throws ExecutionException, InterruptedException,
UnknownHostException {
        String cluster_name = "xxx";
        String host1 = "xx.xx.xx.xx";
        String host2 = "yy.yy.yy.yy";
        Settings settings = Settings.builder()
            .put("client.transport.sniff",false)
            .put("cluster.name", cluster_name)
            .build();
        TransportClient client = new PreBuiltTransportClient(settings)
            .addTransportAddress(new TransportAddress(InetAddress.getByName(host1), 9300))
            .addTransportAddress(new TransportAddress(InetAddress.getByName(host2), 9300));
        IndicesExistsRequest indicesExistsRequest = new IndicesExistsRequest("test");
        ActionFuture<IndicesExistsResponse> exists = client.admin().indices().exists(indicesExistsRequest);
        System.out.println(exists.get().isExists());
    }
}
```

In the information above, *cluster_name* indicates the cluster name, and *host1* and *host2* indicate the IP addresses of the cluster nodes. You can run the **GET _cat/nodes** command to obtain the IP addresses of the nodes.

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

2.7.4.4 Connecting to a Cluster Through Spring Boot

Working directly with Elasticsearch's underlying APIs can increase development complexity, such as additional code maintenance tasks. CSS Elasticsearch clusters support data query and management through Spring Data Elasticsearch (an integrated Elasticsearch component in the Spring Boot ecosystem). This component encapsulates official Elasticsearch Java APIs. Developers can use the Spring repository API or native query DSL to efficiently access clusters without working with underlying APIs. For details about how to use Spring Boot, see **Spring Boot**.

Prerequisites

- The target Elasticsearch cluster is available.
- The server that runs the Java code can communicate with the Elasticsearch cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- Depending on the network configuration method used, obtain the cluster access address. For details, see **Obtaining the Cluster Access Address**.
- Java has been installed on the server and the JDK version is 1.8 or later. Download JDK 1.8 from **Java Downloads**.
- The Spring Boot version has been confirmed. To ensure better compatibility, use a Java client that has the same version as the target Elasticsearch cluster.

  In this document, Spring Boot 2.5.5 is used as an example. The corresponding Spring Data Elasticsearch version is 4.2.x, and the target Elasticsearch cluster version is 7.10.2.

Preparations

1. Check that the Spring Boot version you're using meets the compatibility requirements. For details, see the **official compatibility list**.

   In this document, Spring Boot 2.5.5 is used as an example. The corresponding Spring Data Elasticsearch version is 4.2.x, and the target Elasticsearch cluster version is 7.10.2.
2. Create a Spring Boot project.
3. Declare Java dependencies. Declare the Apache version in Maven mode.

```
<parent>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-parent</artifactId>
   <version>2.5.5</version>
</parent>
<dependencies>
   <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
   </dependency>
   <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-elasticsearch</artifactId>
   </dependency>
   <dependency>
      <groupId>org.elasticsearch.client</groupId>
      <artifactId>elasticsearch-rest-high-level-client</artifactId>
      <version>7.10.2</version>
   </dependency>
</dependencies>
```

Connecting to a Cluster

The sample code varies depending on the security mode settings of the target Elasticsearch cluster. Select the right reference document based on your service scenario.

Table 2-70 Cluster access scenarios

Elasticsearch Cluster Security-Mode Settings	Whether to Load a Security Certificate	Details
Non-security mode Security mode + HTTP	-	**Connecting to a Cluster That Uses HTTP Through Spring Boot**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Elasticsearch Cluster Security-Mode Settings	Whether to Load a Security Certificate	Details
Security mode + HTTPS	No	**Connecting to a Cluster That Uses HTTPS via Spring Boot (Without a Certificate)**
Security mode + HTTPS	Yes	**Connecting to a Cluster That Uses HTTPS via Spring Boot (With a Certificate)**

Connecting to a Cluster That Uses HTTP Through Spring Boot

The following are steps for using Spring Boot to connect to a non-security mode Elasticsearch cluster; or a security-mode Elasticsearch cluster that uses HTTP instead of HTTPS.

1. Configure the **application.properties** file.

```
elasticsearch.url=host1:9200,host2:9200
// You do not need to configure the following two lines for a non-security cluster.
elasticsearch.username=username
elasticsearch.password=password
```

Table 2-71 Configuration parameters

Parameter	Description
host	Address for accessing the cluster.
username	Username for accessing the cluster.
password	Password of the user.

2. Configure the client code.

```
package com.xxx.configuration;

import org.elasticsearch.client.RestHighLevelClient;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.elasticsearch.client.ClientConfiguration;
import org.springframework.data.elasticsearch.client.RestClients;
import org.springframework.data.elasticsearch.config.AbstractElasticsearchConfiguration;
import org.springframework.data.elasticsearch.repository.config.EnableElasticsearchRepositories;

@Configuration
// com.xxx.repository is the repository directory, which is defined by extends org.springframework.data.elasticsearch.repository.ElasticsearchRepository.
@EnableElasticsearchRepositories(basePackages = "com.xxx.repository")
// com.xxx indicates the project directory, for example, com.company.project.
@ComponentScan(basePackages = "com.xxx")
public class Config extends AbstractElasticsearchConfiguration {

    @Value("${elasticsearch.url}")
    public String elasticsearchUrl;

    // There is no need to set the following two parameters for a non-security cluster.
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
    @Value("${elasticsearch.username}")
    public String elasticsearchUsername;

    @Value("${elasticsearch.password}")
    public String elasticsearchPassword;

    @Override
    @Bean
    public RestHighLevelClient elasticsearchClient() {
        final ClientConfiguration clientConfiguration = ClientConfiguration.builder()
            .connectedTo(StringHostParse(elasticsearchUrl))
            // For a non-security cluster, there is no need to configure withBasicAuth.
            .withBasicAuth(elasticsearchUsername, elasticsearchPassword)
            .build();

        return RestClients.create(clientConfiguration).rest();
    }

    private String[] StringHostParse(String hostAndPorts) {
        return hostAndPorts.split(",");
    }
}
```

3. If Spring Boot starts properly, the cluster is connected.

Connecting to a Cluster That Uses HTTPS via Spring Boot (Without a Certificate)

The following are steps for using Spring Boot to connect to a security-mode + HTTPS Elasticsearch cluster without loading a security certificate.

1. Configure the **application.properties** file.

```
elasticsearch.url=host1:9200,host2:9200
elasticsearch.username=username
elasticsearch.password=password
```

Table 2-72 Configuration parameters

Parameter	Description
host	Address for accessing the cluster.
username	Username for accessing the cluster.
password	Password of the user.

2. Configure the client code.

```
package com.xxx.configuration;
import org.elasticsearch.client.RestHighLevelClient;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.elasticsearch.client.ClientConfiguration;
import org.springframework.data.elasticsearch.client.RestClients;
import org.springframework.data.elasticsearch.config.AbstractElasticsearchConfiguration;
import org.springframework.data.elasticsearch.repository.config.EnableElasticsearchRepositories;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLContext;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
@Configuration
// com.xxx.repository is the repository directory, which is defined by extends
org.springframework.data.elasticsearch.repository.ElasticsearchRepository.
@EnableElasticsearchRepositories(basePackages = "com.xxx.repository")
// com.xxx indicates the project directory, for example, com.company.project.
@ComponentScan(basePackages = "com.xxx")
public class Config extends AbstractElasticsearchConfiguration {
    @Value("${elasticsearch.url}")
    public String elasticsearchUrl;
    @Value("${elasticsearch.username}")
    public String elasticsearchUsername;
    @Value("${elasticsearch.password}")
    public String elasticsearchPassword;
    @Override
    @Bean
    public RestHighLevelClient elasticsearchClient() {
        SSLContext sc = null;
        try {
            sc = SSLContext.getInstance("SSL");
            sc.init(null, trustAllCerts, new SecureRandom());
        } catch (KeyManagementException | NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        final ClientConfiguration clientConfiguration = ClientConfiguration.builder()
            .connectedTo(StringHostParse(elasticsearchUrl))
            .usingSsl(sc, new NullHostNameVerifier())
            .withBasicAuth(elasticsearchUsername, elasticsearchPassword)
            .build();
        return RestClients.create(clientConfiguration).rest();
    }
    private String[] StringHostParse(String hostAndPorts) {
        return hostAndPorts.split(",");
    }
    public 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;
        }
    }
}
```

3. If Spring Boot starts properly, the cluster is connected.

Connecting to a Cluster That Uses HTTPS via Spring Boot (With a Certificate)

The following are steps for using Spring Boot to connect to a security-mode + HTTPS Elasticsearch cluster while loading a security certificate.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

1. **Obtaining and Uploading a Security Certificate**.
2. Configure the **application.properties** file.

```
elasticsearch.url=host1:9200,host2:9200
elasticsearch.username=username
elasticsearch.password=password
```

Table 2-73 Configuration parameters

Parameter	Description
host	Address for accessing the cluster.
username	Username for accessing the cluster.
password	Password of the user.

3. Configure the client code.

NOTE

- **com.xxx** indicates the project directory, for example, **com.company.project**.
- **com.xxx.repository** is the repository directory, which is defined by **extends org.springframework.data.elasticsearch.repository.ElasticsearchRepository**.

```
package com.xxx.configuration;
import org.elasticsearch.client.RestHighLevelClient;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.elasticsearch.client.ClientConfiguration;
import org.springframework.data.elasticsearch.client.RestClients;
import org.springframework.data.elasticsearch.config.AbstractElasticsearchConfiguration;
import org.springframework.data.elasticsearch.repository.config.EnableElasticsearchRepositories;
import java.io.File;
import java.io.FileInputStream;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;
@Configuration
// com.xxx.repository is the repository directory, which is defined by extends
org.springframework.data.elasticsearch.repository.ElasticsearchRepository.
@EnableElasticsearchRepositories(basePackages = "com.xxx.repository")
// com.xxx indicates the project directory, for example, com.company.project.
@ComponentScan(basePackages = "com.xxx")
public class Config extends AbstractElasticsearchConfiguration {
    @Value("${elasticsearch.url}")
    public String elasticsearchUrl;
    @Value("${elasticsearch.username}")
    public String elasticsearchUsername;
    @Value("${elasticsearch.password}")
    public String elasticsearchPassword;
    @Override
    @Bean
    public RestHighLevelClient elasticsearchClient() {
        SSLContext sc = null;
        try {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            // certFilePath and certPassword are the path and password of the security certificate.
            TrustManager[] tm = {new MyX509TrustManager(certFilePath, certPassword)};
            sc = SSLContext.getInstance("SSL", "SunJSSE");
            sc.init(null, tm, new SecureRandom());
        } catch (Exception e) {
            e.printStackTrace();
        }
        final ClientConfiguration clientConfiguration = ClientConfiguration.builder()
            .connectedTo(StringHostParse(elasticsearchUrl))
            .usingSsl(sc, new NullHostNameVerifier())
            .withBasicAuth(elasticsearchUsername, elasticsearchPassword)
            .build();
        return RestClients.create(clientConfiguration).rest();
    }

    private String[] StringHostParse(String hostAndPorts) {
        return hostAndPorts.split(",");
    }

    public static class MyX509TrustManager implements X509TrustManager {
        X509TrustManager sunJSSEX509TrustManager;
        MyX509TrustManager(String certFilePath, String certPassword) throws Exception {
            File file = new File(certFilePath);
            if (!file.isFile()) {
                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, certPassword.toCharArray());
            TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
            tmf.init(ks);
            TrustManager[] tms = tmf.getTrustManagers();
            for (TrustManager tm : tms) {
                if (tm instanceof X509TrustManager) {
                    sunJSSEX509TrustManager = (X509TrustManager) tm;
                    return;
                }
            }
            throw new Exception("Couldn't initialize");
        }
        @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 new X509Certificate[0];
        }
    }
    public static class NullHostNameVerifier implements HostnameVerifier {
        @Override
        public boolean verify(String arg0, SSLSession arg1) {
            return true;
        }
    }
}
```

4. If Spring Boot starts properly, the cluster is connected.

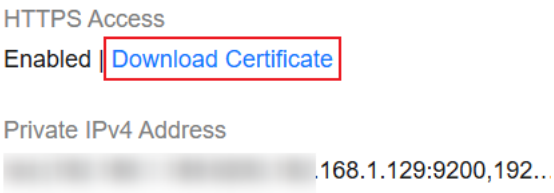
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Obtaining and Uploading a Security Certificate

To access a security-mode Elasticsearch cluster that uses HTTPS, perform the following steps to obtain the security certificate if it is required, and upload it to the client.

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

   Figure 2-18 Downloading a security certificate

2. Convert the security certificate **CloudSearchService.cer**. Upload the downloaded security certificate to the client and use keytool to convert the **.cer** certificate into a **.jks** certificate that can be read by Java.
   - In Linux, run the following command to convert the certificate:
     ```
     keytool -import -alias newname -keystore ./truststore.jks -file ./CloudSearchService.cer
     ```
   - In Windows, run the following command to convert the certificate:
     ```
     keytool -import -alias newname -keystore .\truststore.jks -file .\CloudSearchService.cer
     ```

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

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

2.7.4.5 Connecting to a Cluster Through the Elasticsearch Java API Client

You can query and manage data in a CSS Elasticsearch cluster using the Elasticsearch Java API Client. As the official Java client supported by Elasticsearch 8.x, the Elasticsearch Java API Client encapsulates native Elasticsearch APIs. You only need to construct the required request structure to access an Elasticsearch cluster. For details about how to use the Elasticsearch Java API Client, see **Java API Client**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

> **CAUTION**
>
> For Elasticsearch 7.x, the High Level REST Client is recommended. For Elasticsearch 8.x and later, the Java API Client is used. Although CSS's Elasticsearch 7.10.2 clusters can be connected using the Elasticsearch Java API Client, some features of the client may still be missing if later client versions are used. To ensure compatibility, confirm that the target API has been implemented and verified in CSS.

Prerequisites

- The target Elasticsearch cluster is available, the cluster version is 7.10.2, and the image version is 7.10.2_25.3.0_xxx or later.
- The server that runs the Java code can communicate with the Elasticsearch cluster.
- Depending on the network configuration method used, obtain the cluster access address. For details, see **Obtaining the Cluster Access Address**.
- Java has been installed on the server and the JDK version is 1.8 or later. Download JDK 1.8 from **Java Downloads**.

Introducing Dependencies

Import Java API Client dependencies on the server that runs the Java code using either of the following ways:

- Maven:

  Replace 8.19.0 with the actual Java client version.

  ```
  <dependencies>
      <dependency>
          <groupId>co.elastic.clients</groupId>
          <artifactId>elasticsearch-java</artifactId>
          <version>8.19.0</version>
      </dependency>
      <dependency>
          <groupId>com.fasterxml.jackson.core</groupId>
          <artifactId>jackson-databind</artifactId>
          <version>2.17.0</version>
      </dependency>
  </dependencies>
  ```

- Gradle:

  Replace 8.19.0 with the actual Java client version.

  ```
  dependencies {
      implementation 'co.elastic.clients:elasticsearch-java:8.19.0'
      implementation 'com.fasterxml.jackson.core:jackson-databind:2.17.0'
  }
  ```

Connecting to a Cluster

The sample code varies depending on the security mode settings of the target Elasticsearch cluster. Select the right reference document based on your service scenario.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-74 Cluster access scenarios

Elasticsearch Cluster Security-Mode Settings	Whether to Load a Security Certificate	Details
Non-security mode	-	**Connecting to a Non-Security Mode Cluster Using the Elasticsearch Java API Client**
Security mode + HTTP Security mode + HTTPS	No	**Connecting to a Security-Mode Cluster Using the Elasticsearch Java API Client (Without a Certificate)**
Security mode + HTTPS	Yes	**Connecting to a Security-Mode Cluster Using the Elasticsearch Java API Client (With a Certificate)**

Connecting to a Non-Security Mode Cluster Using the Elasticsearch Java API Client

Use the Elasticsearch Java API Client to connect to an Elasticsearch cluster for which the security mode is disabled, and query whether the **test** index exists. The sample code is as follows:

```
import co.elastic.clients.elasticsearch.ElasticsearchClient;
import co.elastic.clients.elasticsearch.indices.CreateIndexResponse;
import co.elastic.clients.elasticsearch.indices.DeleteIndexResponse;
import co.elastic.clients.elasticsearch.indices.GetIndexResponse;
import co.elastic.clients.json.jackson.JacksonJsonpMapper;
import co.elastic.clients.transport.ElasticsearchTransport;
import co.elastic.clients.transport.rest_client.RestClientTransport;

import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.elasticsearch.client.RestClient;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.KeyManagementException;
import java.security.KeyStore;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.concurrent.TimeUnit;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import javax.net.ssl.X509TrustManager;

public class Main {

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }

    private static ElasticsearchClient create(List<String> host, int port, String protocol, int connectTimeout,
        int connectionRequestTimeout, int socketTimeout, String username, String password, String
certFilePath,
        String certPassword) throws IOException {
        // Create the low-level client
        RestClient restClient = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientBuilder -> {
                // enable user authentication
                if (username != null && password != null) {
                    final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
                    credentialsProvider.setCredentials(AuthScope.ANY,
                        new UsernamePasswordCredentials(username, password));
                    httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
                }
                // set keepalive
                httpClientBuilder.setKeepAliveStrategy(((httpResponse, httpContext) ->
TimeUnit.MINUTES.toMinutes(10)));
                // enable SSL / TLS
                SSLContext sc = (certFilePath != null && certPassword != null) ?
                    createContextFromCaCert(certFilePath, certPassword) : createTrustAllCertsContext();

                SSLIOSessionStrategy sslStrategy = new SSLIOSessionStrategy(sc, new NoopHostnameVerifier());
                httpClientBuilder.setSSLStrategy(sslStrategy);

                httpClientBuilder.setMaxConnTotal(500);
                httpClientBuilder.setMaxConnPerRoute(300);
                return httpClientBuilder;
            })
            .build();

        // Create the transport with a Jackson mapper
        ElasticsearchTransport transport = new RestClientTransport(restClient, new JacksonJsonpMapper());

        // And create the API client
        return new ElasticsearchClient(transport);
    }

    private static SSLContext createTrustAllCertsContext() {
        try {
            SSLContext sslContext = SSLContext.getInstance("SSL");
            sslContext.init(null, trustAllCerts, new SecureRandom());
            return sslContext;
        } catch (NoSuchAlgorithmException | KeyManagementException e) {
            throw new RuntimeException("Can not create the SSLContext", e);
        }
    }

    private static SSLContext createContextFromCaCert(String certFilePath, String certPassword) {
        try {
            // enable SSL / TLS
            TrustManager[] tm = {new MyX509TrustManager(certFilePath, certPassword)};
            SSLContext sc = SSLContext.getInstance("SSL", "SunJSSE");
            //You can also use SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
            sc.init(null, tm, new SecureRandom());
            return sc;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }

    /**
     * Configure trustAllCerts to ignore the certificate configuration.
     */
    private 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;
            }
        }
    };

    /**
     * Configure certificate authentication
     */
    private static class MyX509TrustManager implements X509TrustManager {
        X509TrustManager sunJSSEX509TrustManager;

        MyX509TrustManager(String certFilePath, String certPassword) throws Exception {
            File file = new File(certFilePath);
            if (!file.isFile()) {
                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, certPassword.toCharArray());
            TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
            tmf.init(ks);
            TrustManager[] tms = tmf.getTrustManagers();
            for (TrustManager tm : tms) {
                if (tm instanceof X509TrustManager) {
                    sunJSSEX509TrustManager = (X509TrustManager) tm;
                    return;
                }
            }
            throw new Exception("Couldn't initialize");
        }

        @Override
        public void checkClientTrusted(X509Certificate[] chain, String authType) {

        }

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

        }

        @Override
        public X509Certificate[] getAcceptedIssuers() {
            return new X509Certificate[0];
        }
    }
```

```
    /**
* The following is an example of the main function. Call the create function to create a client and check whether the test index exists.
     */
    public static void main(String[] args) throws IOException {
        ElasticsearchClient esClient = create(Arrays.asList("{IP address of the target cluster}"), 9200, "http", 1000, 1000, 1000, null, null, null, null);

        CreateIndexResponse indexRequest = esClient.indices()
            .create(createIndexBuilder -> createIndexBuilder.index("test"));
        boolean acknowledged = indexRequest.acknowledged();
        System.out.println("Create index successfully! " + acknowledged);

        GetIndexResponse getIndexResponse = esClient.indices().get(getIndexRequest -> getIndexRequest.index("test"));
        System.out.println("Query index successfully! \n" + getIndexResponse.toString());

        DeleteIndexResponse deleteResponse = esClient.indices()
            .delete(createIndexBuilder -> createIndexBuilder.index("test"));
        System.out.println("Delete index successfully! \n" + deleteResponse.toString());
    }
}
```

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the Elasticsearch Java API Client (Without a Certificate)

Use the Elasticsearch Java API Client to connect to a security-mode Elasticsearch cluster (HTTP or HTTPS) without loading a security certificate, and query whether the **test** index exists. The sample code is as follows:

```
import co.elastic.clients.elasticsearch.ElasticsearchClient;
import co.elastic.clients.elasticsearch.indices.CreateIndexResponse;
import co.elastic.clients.elasticsearch.indices.DeleteIndexResponse;
import co.elastic.clients.elasticsearch.indices.GetIndexResponse;
import co.elastic.clients.json.jackson.JacksonJsonpMapper;
import co.elastic.clients.transport.ElasticsearchTransport;
import co.elastic.clients.transport.rest_client.RestClientTransport;

import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.elasticsearch.client.RestClient;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.KeyManagementException;
import java.security.KeyStore;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.concurrent.TimeUnit;

import javax.net.ssl.SSLContext;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;

public class Main {

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }

    private static ElasticsearchClient create(List<String> host, int port, String protocol, int connectTimeout,
        int connectionRequestTimeout, int socketTimeout, String username, String password, String
certFilePath,
        String certPassword) throws IOException {
        // Create the low-level client
        RestClient restClient = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientBuilder -> {
                // enable user authentication
                if (username != null && password != null) {
                    final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
                    credentialsProvider.setCredentials(AuthScope.ANY,
                        new UsernamePasswordCredentials(username, password));
                    httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
                }
                // set keepalive
                httpClientBuilder.setKeepAliveStrategy(((httpResponse, httpContext) ->
TimeUnit.MINUTES.toMinutes(10)));
                // enable SSL / TLS
                SSLContext sc = (certFilePath != null && certPassword != null) ?
                    createContextFromCaCert(certFilePath, certPassword) : createTrustAllCertsContext();

                SSLIOSessionStrategy sslStrategy = new SSLIOSessionStrategy(sc, new NoopHostnameVerifier());
                httpClientBuilder.setSSLStrategy(sslStrategy);

                httpClientBuilder.setMaxConnTotal(500);
                httpClientBuilder.setMaxConnPerRoute(300);
                return httpClientBuilder;
            })
            .build();

        // Create the transport with a Jackson mapper
        ElasticsearchTransport transport = new RestClientTransport(restClient, new JacksonJsonpMapper());

        // And create the API client
        return new ElasticsearchClient(transport);
    }

    private static SSLContext createTrustAllCertsContext() {
        try {
            SSLContext sslContext = SSLContext.getInstance("SSL");
            sslContext.init(null, trustAllCerts, new SecureRandom());
            return sslContext;
        } catch (NoSuchAlgorithmException | KeyManagementException e) {
            throw new RuntimeException("Can not create the SSLContext", e);
        }
    }

    private static SSLContext createContextFromCaCert(String certFilePath, String certPassword) {
        try {
            // enable SSL / TLS
            TrustManager[] tm = {new MyX509TrustManager(certFilePath, certPassword)};
            SSLContext sc = SSLContext.getInstance("SSL", "SunJSSE");
            //You can also use SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            sc.init(null, tm, new SecureRandom());
            return sc;
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }

    /**
     * Configure trustAllCerts to ignore the certificate configuration.
     */
    private 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;
            }
        }
    };

    /**
     * Configure certificate authentication
     */
    private static class MyX509TrustManager implements X509TrustManager {
        X509TrustManager sunJSSEX509TrustManager;

        MyX509TrustManager(String certFilePath, String certPassword) throws Exception {
            File file = new File(certFilePath);
            if (!file.isFile()) {
                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, certPassword.toCharArray());
            TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
            tmf.init(ks);
            TrustManager[] tms = tmf.getTrustManagers();
            for (TrustManager tm : tms) {
                if (tm instanceof X509TrustManager) {
                    sunJSSEX509TrustManager = (X509TrustManager) tm;
                    return;
                }
            }
            throw new Exception("Couldn't initialize");
        }

        @Override
        public void checkClientTrusted(X509Certificate[] chain, String authType) {

        }

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

        }

        @Override
        public X509Certificate[] getAcceptedIssuers() {
            return new X509Certificate[0];
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        }
    }

    /**
* The following is an example of the main function. Call the create function to create a client and check whether the test index exists.
     */
    public static void main(String[] args) throws IOException {
        ElasticsearchClient esClient = create(Arrays.asList("{IP address of the target cluster}"), 9200, "https", 1000, 1000, 1000, "{Username}", "{Password}", null, null);

        CreateIndexResponse indexRequest = esClient.indices()
            .create(createIndexBuilder -> createIndexBuilder.index("test"));
        boolean acknowledged = indexRequest.acknowledged();
        System.out.println("Create index successfully! " + acknowledged);

        GetIndexResponse getIndexResponse = esClient.indices().get(getIndexRequest -> getIndexRequest.index("test"));
        System.out.println("Query index successfully! \n" + getIndexResponse.toString());

        DeleteIndexResponse deleteResponse = esClient.indices()
            .delete(createIndexBuilder -> createIndexBuilder.index("test"));
        System.out.println("Delete index successfully! \n" + deleteResponse.toString());
    }
}
```

Table 2-75 Variables

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol, which can be **http** or **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Connecting to a Security-Mode Cluster Using the Elasticsearch Java API Client (With a Certificate)

Use the Elasticsearch Java API Client to connect to a security-mode Elasticsearch cluster that uses HTTPS with a security certificate loaded, and query whether the **test** index exists. The sample code is as follows:

CAUTION

For how to obtain and upload a security certificate, see **Obtaining and Uploading a Security Certificate**.

```
import co.elastic.clients.elasticsearch.ElasticsearchClient;
import co.elastic.clients.elasticsearch.indices.CreateIndexResponse;
import co.elastic.clients.elasticsearch.indices.DeleteIndexResponse;
import co.elastic.clients.elasticsearch.indices.GetIndexResponse;
import co.elastic.clients.json.jackson.JacksonJsonpMapper;
import co.elastic.clients.transport.ElasticsearchTransport;
import co.elastic.clients.transport.rest_client.RestClientTransport;

import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.elasticsearch.client.RestClient;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.KeyManagementException;
import java.security.KeyStore;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.concurrent.TimeUnit;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;

public class Main {

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }

    private static ElasticsearchClient create(List<String> host, int port, String protocol, int connectTimeout,
        int connectionRequestTimeout, int socketTimeout, String username, String password, String
certFilePath,
        String certPassword) throws IOException {
        // Create the low-level client
        RestClient restClient = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientBuilder -> {
                // enable user authentication
                if (username != null && password != null) {
                    final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
                    credentialsProvider.setCredentials(AuthScope.ANY,
                        new UsernamePasswordCredentials(username, password));
                    httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
                }
                // set keepalive
                httpClientBuilder.setKeepAliveStrategy(((httpResponse, httpContext) ->
TimeUnit.MINUTES.toMinutes(10)));
                // enable SSL / TLS
                SSLContext sc = (certFilePath != null && certPassword != null) ?
                    createContextFromCaCert(certFilePath, certPassword) : createTrustAllCertsContext();

                SSLIOSessionStrategy sslStrategy = new SSLIOSessionStrategy(sc, new NoopHostnameVerifier());
                httpClientBuilder.setSSLStrategy(sslStrategy);

                httpClientBuilder.setMaxConnTotal(500);
                httpClientBuilder.setMaxConnPerRoute(300);
                return httpClientBuilder;
            })
            .build();

        // Create the transport with a Jackson mapper
        ElasticsearchTransport transport = new RestClientTransport(restClient, new JacksonJsonpMapper());

        // And create the API client
        return new ElasticsearchClient(transport);
    }

    private static SSLContext createTrustAllCertsContext() {
        try {
            SSLContext sslContext = SSLContext.getInstance("SSL");
            sslContext.init(null, trustAllCerts, new SecureRandom());
            return sslContext;
        } catch (NoSuchAlgorithmException | KeyManagementException e) {
            throw new RuntimeException("Can not create the SSLContext", e);
        }
    }

    private static SSLContext createContextFromCaCert(String certFilePath, String certPassword) {
        try {
            // enable SSL / TLS
            TrustManager[] tm = {new MyX509TrustManager(certFilePath, certPassword)};
            SSLContext sc = SSLContext.getInstance("SSL", "SunJSSE");
            //You can also use SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
            sc.init(null, tm, new SecureRandom());
            return sc;
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }

    /**
     * Configure trustAllCerts to ignore the certificate configuration.
     */
    private 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;
        }
    }
};

/**
 * Configure certificate authentication
 */
private static class MyX509TrustManager implements X509TrustManager {
    X509TrustManager sunJSSEX509TrustManager;

    MyX509TrustManager(String certFilePath, String certPassword) throws Exception {
        File file = new File(certFilePath);
        if (!file.isFile()) {
            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, certPassword.toCharArray());
        TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
        tmf.init(ks);
        TrustManager[] tms = tmf.getTrustManagers();
        for (TrustManager tm : tms) {
            if (tm instanceof X509TrustManager) {
                sunJSSEX509TrustManager = (X509TrustManager) tm;
                return;
            }
        }
        throw new Exception("Couldn't initialize");
    }

    @Override
    public void checkClientTrusted(X509Certificate[] chain, String authType) {

    }

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

    }

    @Override
    public X509Certificate[] getAcceptedIssuers() {
        return new X509Certificate[0];
    }
}

/**
* The following is an example of the main function. Call the create function to create a client and check whether the test index exists.
 */
public static void main(String[] args) throws IOException {
    ElasticsearchClient esClient = create(Arrays.asList("{IP address of the target cluster}"), 9200, "https", 1000, 1000, 1000, "{Username}", "{Password}", "{Certificate path}", "{Certificate password}")

    CreateIndexResponse indexRequest = esClient.indices()
        .create(createIndexBuilder -> createIndexBuilder.index("test"));
    boolean acknowledged = indexRequest.acknowledged();
    System.out.println("Create index successfully! " + acknowledged);

    GetIndexResponse getIndexResponse = esClient.indices().get(getIndexRequest -> getIndexRequest.index("test"));
    System.out.println("Query index successfully! \n" + getIndexResponse.toString());
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        DeleteIndexResponse deleteResponse = esClient.indices()
            .delete(createIndexBuilder -> createIndexBuilder.index("test"));
        System.out.println("Delete index successfully! \n" + deleteResponse.toString());
    }
}
```

Table 2-76 Function parameters

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol. Set this parameter to **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.
certFilePath	Path for storing the security certificate.
certPassword	Password of the security certificate.

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

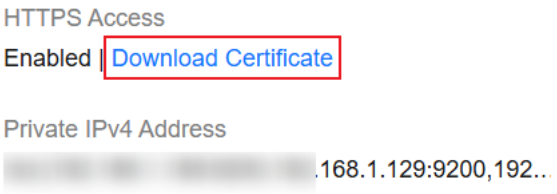
Obtaining and Uploading a Security Certificate

To access a security-mode Elasticsearch cluster that uses HTTPS, perform the following steps to obtain the security certificate if it is required, and upload it to the client.

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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 2-19 Downloading a security certificate



2. Convert the security certificate **CloudSearchService.cer**. Upload the downloaded security certificate to the client and use keytool to convert the **.cer** certificate into a **.jks** certificate that can be read by Java.
   - In Linux, run the following command to convert the certificate:
     ```
     keytool -import -alias newname -keystore ./truststore.jks -file ./CloudSearchService.cer
     ```
   - In Windows, run the following command to convert the certificate:
     ```
     keytool -import -alias newname -keystore .\truststore.jks -file .\CloudSearchService.cer
     ```

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

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

FAQ: How Do I Increase the Number of Client Connections to Handle High Concurrency?

When using the Elasticsearch Java API Client, the default maximum number of client connections may become insufficient under high concurrency, leading to request delays or failures.

To address this, you can increase the connection limit by customizing HttpClientConfigCallback during client initialization. This involves injecting an HttpClientBuilder to increase the maximum number of client connections supported, thereby improving the cluster's ability to handle concurrent requests.

Configuration example:

```
// Set the maximum number of total connections to 500. (Set this parameter based on the typical concurrency the cluster handles.)
httpClientBuilder.setMaxConnTotal(500);
//Set the maximum number of connections per route to 300. You are advised to set this value to approximately 60% of the total client connection limit.
httpClientBuilder.setMaxConnPerRoute(300);
```

2.7.5 Connecting to a Cluster Using Python

When using CSS's Elasticsearch clusters for data query and management, developers traditionally rely on programming languages. However, directly working with Elasticsearch's HTTP APIs often involves complex and error-prone coding. To simplify development, CSS Elasticsearch supports the official Elasticsearch Python client. The client library (elasticsearch-py) provides comprehensive encapsulation of Elasticsearch APIs, enabling developers to perform index management, CRUD (Create, Read, Update, Delete) operations, and document searches through native Python APIs. For further information, see **Elasticsearch Python Client**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Prerequisites

- The target Elasticsearch cluster is available.
- The server that runs the Python code can communicate with the Elasticsearch cluster.
- Depending on the network configuration method used, obtain the cluster access address. For details, see **Obtaining the Cluster Access Address**.
- Python 3 has been installed on the server.

Installing the Python Client Library

Install the Python client library on the server that runs the Python code. To ensure better compatibility, use a Python client library that has the same version as the target Elasticsearch cluster.

Replace 7.10 with the actual Python client version.

```
pip install Elasticsearch==7.10
```

Connecting to a Cluster

The sample code varies depending on the security mode settings of the target Elasticsearch cluster. Select the right reference document based on your service scenario.

Table 2-77 Cluster access scenarios

Elasticsearch Cluster Security-Mode Settings	Details
Non-security mode	**Connecting to a Non-Security Mode Cluster with the Elasticsearch Python Client**
Security mode + HTTP	**Connecting to a Security-Mode+HTTP Cluster with the Elasticsearch Python Client**
Security mode + HTTPS	**Connecting to a Security-Mode+HTTPS Cluster with the Elasticsearch Python Client**

Connecting to a Non-Security Mode Cluster with the Elasticsearch Python Client

Use the Elasticsearch Python client to connect to an Elasticsearch cluster for which the security mode is disabled, and query whether the **test** index exists.

The following is an example of the code for creating a client instance using the cluster connection information:

```
from elasticsearch import Elasticsearch

class ElasticFactory(object):

    def __init__(self, host: list, port: str, username: str, password: str):
        self.port = port
        self.host = host
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        self.username = username
        self.password = password

    def create(self) -> Elasticsearch:
        addrs = []
        for host in self.host:
            addr = {'host': host, 'port': self.port}
            addrs.append(addr)

        if self.username and self.password:
            elasticsearch = Elasticsearch(addrs, http_auth=(self.username, self.password))
        else:
            elasticsearch = Elasticsearch(addrs)
        return elasticsearch

es = ElasticFactory(["{cluster address}"], "9200", None, None).create()
print(es.indices.exists(index='test'))
```

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode+HTTP Cluster with the Elasticsearch Python Client

Use the Elasticsearch Python client to connect to a security-mode Elasticsearch cluster that uses HTTP, and query whether the **test** index exists.

The following is an example of the code for creating a client instance using the cluster connection information:

```
from elasticsearch import Elasticsearch

class ElasticFactory(object):

    def __init__(self, host: list, port: str, username: str, password: str):
        self.port = port
        self.host = host
        self.username = username
        self.password = password

    def create(self) -> Elasticsearch:
        addrs = []
        for host in self.host:
            addr = {'host': host, 'port': self.port}
            addrs.append(addr)

        if self.username and self.password:
            elasticsearch = Elasticsearch(addrs, http_auth=(self.username, self.password))
        else:
            elasticsearch = Elasticsearch(addrs)
        return elasticsearch

es = ElasticFactory(["xxx.xxx.xxx.xxx"], "9200", "username", "password").create()
print(es.indices.exists(index='test'))
```

Table 2-78 Variables

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
port	Access port of the cluster. Enter **9200**.
username	Username for accessing the cluster.
password	Password of the user.

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode+HTTPS Cluster with the Elasticsearch Python Client

Use the Elasticsearch Python client to connect to a security-mode Elasticsearch cluster that uses HTTPS, and query whether the **test** index exists.

The following is an example of the code for creating a client instance using the cluster connection information:

```
from elasticsearch import Elasticsearch
import ssl

class ElasticFactory(object):

    def __init__(self, host: list, port: str, username: str, password: str):
        self.port = port
        self.host = host
        self.username = username
        self.password = password

    def create(self) -> Elasticsearch:
        context = ssl._create_unverified_context()

        addrs = []
        for host in self.host:
            addr = {'host': host, 'port': self.port}
            addrs.append(addr)

        if self.username and self.password:
            elasticsearch = Elasticsearch(addrs, http_auth=(self.username, self.password), scheme="https",
ssl_context=context)
        else:
            elasticsearch = Elasticsearch(addrs)
        return elasticsearch

es = ElasticFactory(["xxx.xxx.xxx.xxx"], "9200", "username", "password").create()
print(es.indices.exists(index='test'))
```

Table 2-79 Variables

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. Enter **9200**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
username	Username for accessing the cluster.
password	Password of the user.

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

2.7.6 Connecting to a Cluster Using Go

When querying and managing data in CSS's Elasticsearch clusters, developers sometimes use the Go language. This section describes how to connect to a CSS Elasticsearch cluster using Go.

Prerequisites

- The target Elasticsearch cluster is available.
- The server that runs Go can communicate with the Elasticsearch cluster.
- Depending on the network configuration method used, obtain the cluster access address. For details, see **Obtaining the Cluster Access Address**.
- The Elasticsearch Go client has been installed on the server. For details, see **Elasticsearch Go Client**.

Connecting to a Non-Security Mode Cluster

Connect to a non-security mode cluster. The sample code is as follows:

```
package main

import (
    "github.com/elastic/go-elasticsearch/v7"
    "log"
)

func main() {
    cfg := elasticsearch.Config{
        Addresses: []string{
            "http://HOST:9200/",
        },
    }

    es, _ := elasticsearch.NewClient(cfg)
    log.Println(es.Info())
}
```

In the information above, **HOST** indicates the internal IP address of a cluster node.

Connecting to a Security-Mode Cluster That Uses HTTP

The sample code is as follows:

```
package main

import (
    "github.com/elastic/go-elasticsearch/v7"
    "log"
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
)

func main() {
    cfg := elasticsearch.Config{
        Addresses: []string{
            "http://HOST:9200/",
        },
        Username: "USERNAME",
        Password: "XXXXXXXX",
    }

    es, _ := elasticsearch.NewClient(cfg)
    log.Println(es.Info())
}
```

Table 2-80 Variables

Parameter	Description
HOST	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
USERNAME	Username for accessing the cluster.
PASSWORD	Password of the user.

Connecting to a Security-Mode Cluster That Uses HTTPS (Without a Certificate)

The sample code is as follows:

```
package main

import (
    "crypto/tls"
    "github.com/elastic/go-elasticsearch/v7"
    "log"
    "net/http"
)

func main() {
    cfg := elasticsearch.Config{
        Addresses: []string{
            "https://HOST:9200/",
        },
        Username: "USERNAME",
        Password: "PASSWORD",
        Transport: &http.Transport{
            TLSClientConfig: &tls.Config{
                InsecureSkipVerify: true,
            },
        },
    }

    es, _ := elasticsearch.NewClient(cfg)
    log.Println(es.Info())
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-81 Variables

Parameter	Description
HOST	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
USERNAME	Username for accessing the cluster.
PASSWORD	Password of the user.

Connecting to a Security-Mode Cluster That Uses HTTPS (With a Certificate)

The sample code is as follows:

```
package main

import (
    "crypto/tls"
    "crypto/x509"
    "flag"
    "github.com/elastic/go-elasticsearch/v7"
    "io/ioutil"
    "log"
    "net"
    "net/http"
    "time"
)

func main() {
    insecure := flag.Bool("insecure-ssl", false, "Accept/Ignore all server SSL certificates")
    flag.Parse()

    // Get the SystemCertPool, continue with an empty pool on error
    rootCAs, _ := x509.SystemCertPool()
    if rootCAs == nil {
        rootCAs = x509.NewCertPool()
    }

    // Read in the cert file
    certs, err := ioutil.ReadFile("/tmp/CloudSearchService.cer")
    if err != nil {
        log.Fatalf("Failed to append %q to RootCAs: %v", "xxx", err)
    }

    // Append our cert to the system pool
    if ok := rootCAs.AppendCertsFromPEM(certs); !ok {
        log.Println("No certs appended, using system certs only")
    }

    config := elasticsearch.Config{
        Addresses: []string{
            "https://HOST:9200/",
        },
        Username: "USERNAME",
        Password: "PASSWORD",
        Transport: &http.Transport{
            MaxIdleConnsPerHost:   10,
            ResponseHeaderTimeout: time.Second,
            DialContext: (&net.Dialer{
                Timeout:   30 * time.Second,
                KeepAlive: 30 * time.Second,
            }).DialContext,
            TLSClientConfig: &tls.Config{
                InsecureSkipVerify: *insecure,
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            RootCAs:            rootCAs,
        },
    },
}
es, _ := elasticsearch.NewClient(config)
log.Println(elasticsearch.Version)
log.Println(es.Info())
}
```

Table 2-82 Variables

Parameter	Description
HOST	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
USERNAME	Username for accessing the cluster.
PASSWORD	Password of the user.

Running Code

Write the code above to the **EsTest.go** file based on the cluster type and save the file to an independent directory. Run the following command in that directory to run the code:

```
go env -w GO111MODULE=on
go env -w GONOSUMDB=*

go mod init test
go mod tidy
go run EsTest.go
```

If the cluster information is returned, the cluster is connected.

2.7.7 Connecting to a Cluster Using PHP

In big data search and analytics scenarios, developers traditionally rely on programming languages to work with Elasticsearch clusters. However, the direct use of Elasticsearch's HTTP APIs can involve complex and error-prone coding. To simplify development, CSS Elasticsearch supports the official Elasticsearch PHP client. The client library (elasticsearch-php) provides comprehensive encapsulation of Elasticsearch APIs, enabling developers to perform index management, CRUD (Create, Read, Update, Delete) operations, and document searches through native PHP APIs.

Prerequisites

- The target Elasticsearch cluster is available.
- The server that runs the PHP code can communicate with the Elasticsearch cluster.
- Depending on the network configuration method used, obtain the cluster access address. For details, see **Obtaining the Cluster Access Address**.
- The Elasticsearch PHP client has been installed on the server. For details, see **Elasticsearch PHP Client**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

In this document, the installed PHP version is 8.0.30, the Composer version is 2.9.2, the Elasticsearch PHP client version is 7.17.2, and the connected Elasticsearch cluster version is 7.10.2.

Installing the PHP Client Library

Install the PHP client library on the server that runs the PHP code.

```
yum install -y php
```

Connecting to a Cluster

The sample code varies depending on the security mode settings of the target Elasticsearch cluster. Select the right reference document based on your service scenario.

Table 2-83 Cluster access scenarios

Elasticsearch Cluster Security-Mode Settings	Details
Non-security mode	**Connecting to a Non-Security Mode Cluster with the Elasticsearch PHP Client**
Security mode + HTTP	**Connecting to a Security-Mode+HTTP Cluster with the Elasticsearch PHP Client**
Security mode + HTTPS	**Connecting to a Security-Mode+HTTPS Cluster with the Elasticsearch PHP Client (Without a Certificate)**
Security mode + HTTPS	**Connecting to a Security-Mode+HTTPS Cluster with the Elasticsearch PHP Client (With a Certificate)**

Connecting to a Non-Security Mode Cluster with the Elasticsearch PHP Client

Use the Elasticsearch PHP client to connect to a non-security mode Elasticsearch cluster, and obtain the cluster health status.

The following is an example of the code for creating a client instance using the cluster connection information:

```
<?php
require 'vendor/autoload.php';

use Elasticsearch\ClientBuilder;

class ElasticsearchClient
{
    private $client;

    public function __construct($config)
    {
        $this->client = ClientBuilder::create()
            ->setHosts($config['hosts'])
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            ->setConnectionPool('\Elasticsearch\ConnectionPool\SimpleConnectionPool', [])
            ->setRetries($config['retries'] ?? 3)
            ->setConnectionParams([
                'client' => [
                    'timeout' => 10,
                    'connect_timeout' => 5
                ]
            ])
            ->build();
    }

    public function getClusterHealth()
    {
        return $this->client->cluster()->health();
    }
}

$config = [
    'hosts' => [
        'http://IP:9200',
        'http://IP:9200'
    ],
    'retries' => 2
];

$es = new ElasticsearchClient($config);
print_r($es->getClusterHealth());
?>
```

Table 2-84 Config parameters

Parameter	Description
hosts	Cluster access address, including the protocol (HTTP), cluster IP address, and port 9200, for example, http://xx.xx.xx.xx:9200. If there are multiple addresses, separate them with a comma (,).
retries	Number of retries upon a failure. The SimpleConnectionPool serves as the connection pool, enabling a retry mechanism upon connection failures.

The code obtains the cluster health status. If a result similar to the following is returned, the cluster is connected.

```
Array
(
    [cluster_name] => css-xxxx-php
    [status] => green
    ...
)
```

Connecting to a Security-Mode+HTTP Cluster with the Elasticsearch PHP Client

Use the Elasticsearch PHP client to connect to a security-mode Elasticsearch cluster that uses HTTP, and obtain the cluster health status.

The following is an example of the code for creating a client instance using the cluster connection information:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
<?php
require 'vendor/autoload.php';

use Elasticsearch\ClientBuilder;

class ElasticsearchClient
{
    private $client;

    public function __construct($config)
    {
        $this->client = ClientBuilder::create()
            ->setHosts($config['hosts'])
            ->setBasicAuthentication($config['username'], $config['password'])
            ->setConnectionPool('\Elasticsearch\ConnectionPool\SimpleConnectionPool', [])
            ->setRetries($config['retries'] ?? 3)
            ->setConnectionParams([
                'client' => [
                    'timeout' => 10,
                    'connect_timeout' => 5
                ]
            ])
            ->build();
    }

    public function getClusterHealth()
    {
        return $this->client->cluster()->health();
    }
}

$config = [
    'hosts' => [
        'http://IP:9200',
        'http://IP:9200'
    ],
    'username' => 'USERNAME',
    'password' => 'PASSWORD',
    'retries' => 2
];

$es = new ElasticsearchClient($config);
print_r($es->getClusterHealth());
?>
```

Table 2-85 Config parameters

Parameter	Description
hosts	Cluster access address, including the protocol (HTTP), cluster IP address, and port 9200, for example, http://xx.xx.xx.xx:9200. If there are multiple addresses, separate them with a comma (,).
username	Username for accessing the cluster.
password	Password of the user.
retries	Number of retries upon a failure. The SimpleConnectionPool serves as the connection pool, enabling a retry mechanism upon connection failures.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

The code obtains the cluster health status. If a result similar to the following is returned, the cluster is connected.

```
Array
(
    [cluster_name] => css-xxxx-php
    [status] => green
    ...
)
```

Connecting to a Security-Mode+HTTPS Cluster with the Elasticsearch PHP Client (Without a Certificate)

Use the Elasticsearch PHP client to connect to a security-mode Elasticsearch cluster that uses HTTPS without loading a security certificate, and obtain the cluster health status.

The following is an example of the code for creating a client instance using the cluster connection information:

```
<?php
require 'vendor/autoload.php';

use Elasticsearch\ClientBuilder;

class ElasticsearchClient
{
    private $client;

    public function __construct($config)
    {
        $this->client = ClientBuilder::create()
            ->setHosts($config['hosts'])
            ->setBasicAuthentication($config['username'], $config['password'])
            ->setSSLVerification($config['ssl_verify'] ?? false)
            ->setConnectionPool('\Elasticsearch\ConnectionPool\SimpleConnectionPool', [])
            ->setRetries($config['retries'] ?? 3)
            ->setConnectionParams([
                'client' => [
                    'timeout' => 10,
                    'connect_timeout' => 5
                ]
            ])
            ->build();
    }

    public function getClusterHealth()
    {
        return $this->client->cluster()->health();
    }
}

$config = [
    'hosts' => [
        'https://IP:9200',
        'https://IP:9200'
    ],
    'username' => 'USERNAME',
    'password' => 'PASSWORD',
    'ssl_verify' => false,
    'retries' => 2
];

$es = new ElasticsearchClient($config);
print_r($es->getClusterHealth());
?>
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-86 Config parameters

Parameter	Description
hosts	Cluster access address, including the protocol (HTTPS), cluster IP address, and port 9200, for example, https://xx.xx.xx.xx:9200. If there are multiple addresses, separate them with a comma (,).
username	Username for accessing the cluster.
password	Password of the user.
ssl_verify	Whether to load a security certificate.
retries	Number of retries upon a failure. The SimpleConnectionPool serves as the connection pool, enabling a retry mechanism upon connection failures.

The code obtains the cluster health status. If a result similar to the following is returned, the cluster is connected.

```
Array
(
    [cluster_name] => css-xxxx-php
    [status] => green
    ...
)
```

Connecting to a Security-Mode+HTTPS Cluster with the Elasticsearch PHP Client (With a Certificate)

Use the Elasticsearch PHP client to connect to a security-mode Elasticsearch cluster that uses HTTPS while loading a security certificate, and obtain the cluster health status.

CAUTION

For how to obtain and upload a security certificate, see **Obtaining and Uploading a Security Certificate**.

The following is an example of the code for creating a client instance using the cluster connection information:

```
<?php
require 'vendor/autoload.php';

use Elasticsearch\ClientBuilder;

class ElasticsearchClient
{
    private $client;

    public function __construct($config)
    {
        $this->client = ClientBuilder::create()
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        ->setHosts($config['hosts'])
        ->setBasicAuthentication($config['username'], $config['password'])
        ->setSSLVerification($config['ssl_verify'] ?? true)
        ->setConnectionPool('\Elasticsearch\ConnectionPool\SimpleConnectionPool', [])
        ->setRetries($config['retries'] ?? 3)
        ->setConnectionParams([
            'client' => [
                'timeout' => 10,
                'connect_timeout' => 5
            ]
        ])
        ->build();
    }

    public function getClusterHealth()
    {
        return $this->client->cluster()->health();
    }
}

$config = [
    'hosts' => [
        'https://IP:9200',
        'https://IP:9200'
    ],
    'username' => 'USERNAME',
    'password' => 'PASSWORD',
    'ssl_verify' => true,
    'retries' => 2
];

$es = new ElasticsearchClient($config);
print_r($es->getClusterHealth());
?>
```

Table 2-87 Config parameters

Parameter	Description
hosts	Cluster access address, including the protocol (HTTPS), cluster IP address, and port 9200, for example, https://xx.xx.xx.xx:9200. If there are multiple addresses, separate them with a comma (,).
username	Username for accessing the cluster.
password	Password of the user.
ssl_verify	Whether to load a security certificate.
retries	Number of retries upon a failure. The SimpleConnectionPool serves as the connection pool, enabling a retry mechanism upon connection failures.

The code obtains the cluster health status. If a result similar to the following is returned, the cluster is connected.

```
Array
(
    [cluster_name] => css-xxxx-php
    [status] => green
    ...
)
```

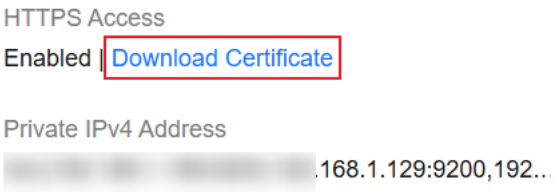
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Obtaining and Uploading a Security Certificate

To access a security-mode Elasticsearch cluster that uses HTTPS, perform the following steps to obtain the security certificate if it is required, and upload it to the client.

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

   Figure 2-20 Downloading a security certificate

2. Upload the downloaded certificate to the server.
3. Find the PHP installation directory on the server and modify the **php.ini** file.
   curl.cainfo="/tmp/CloudSearchService" // Replace it with the full absolute path of the security certificate.

2.7.8 Connecting MRS Hive to a Cluster

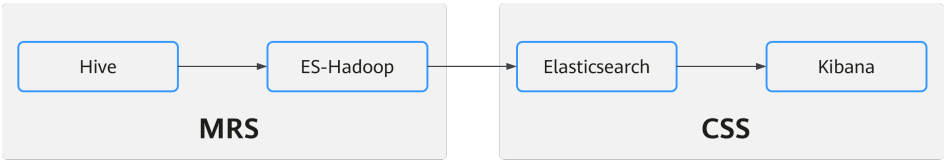
In big data analytics scenarios, Hive excels at batch processing massive datasets, while Elasticsearch enables real-time search and analytics. Integrating the two is key to bridging the gap between large-scale batch processing and low-latency search, effectively breaking down data silos. Using the ES-Hadoop connector (Elasticsearch for Apache Hadoop), you can integrate MRS Hive with CSS Elasticsearch by allowing Hive to directly read and write Elasticsearch data in the form of foreign tables. This approach preserves familiar Hadoop ecosystem workflows while delivering second-level search latency

Solution

ES-Hadoop is a connector library that integrates Elasticsearch with the Hadoop ecosystem, allowing MapReduce, Hive, and Spark to directly read from and write to Elasticsearch indexes.

In the MRS Hive-Elasticsearch integration scenario, ES-Hadoop acts as a query translation layer: When you execute SQL queries in Hive, ES-Hadoop converts Hive's execution plan into Elasticsearch query requests sent over REST.

Figure 2-21 Connecting MRS Hive to an Elasticsearch cluster



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

For more information about ES-Hadoop and Hive, see the official document **Apache Hive integration**.

Prerequisites

- Make sure the CSS and MRS clusters are in the same region, AZ, VPC, and subnet, so they are connected to each other.
- Download the ES-Hadoop installation package and unzip it to obtain the **elasticsearch-hadoop-x.x.x.jar** file. Keep the ES-Hadoop version consistent with that of the CSS cluster. For example, use ES-Hadoop 7.6.2 with Elasticsearch 7.6.2.

  Download address: **https://www.elastic.co/downloads/hadoop**
- Download the HttpClient dependency package **commons-httpclient-x.x.jar**. Version: 3.1 or later is recommended.

  Download address: **https://mvnrepository.com/artifact/commons-httpclient/commons-httpclient**
- Obtain the private network address of the CSS cluster.

Connecting MRS Hive to an Elasticsearch Cluster

1. Check the network connectivity between MRS and CSS.
   a. Obtain the private network address of the target Elasticsearch cluster.
   b. Run the cURL command on each MRS cluster node to make sure it can connect to the CSS Elasticsearch cluster.
      - For a cluster with the security mode disabled:
        ```
        curl -X GET http://<host>:<port>
        ```
      - For a security-mode cluster that uses HTTP:
        ```
        curl -X GET http://<host>:<port> -u <user>:<password>
        ```
      - For a security-mode cluster that uses HTTPS:
        ```
        curl -X GET https://<host>:<port> -u <user>:<password> -ik
        ```

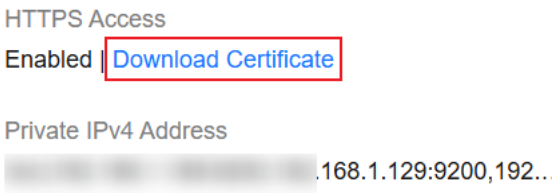
Table 2-88 Variables

Variable	Description
host	IP address of each node in the cluster. If the cluster contains multiple nodes, there will be multiple IP addresses. You can use any of them.
port	Port number for accessing a cluster node. Generally, the port number is 9200.
user	Username for accessing the cluster.
password	Password of the user. If the password contains special characters, enclose the username and password in single quotation marks, for example, **curl -u "user:password!" "http://<host>:<port>"**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. Upload the ES-Hadoop package and dependency package to HDFS. Ensure that all Hive compute nodes can access the ES-Hadoop driver.
   a. Upload the downloaded **elasticsearch-hadoop-x.x.x.jar** and **commons-httpclient-x.x.jar** to the MRS client.
   b. Create an HDFS directory on the MRS client machine and save the JAR files in the directory.
      ```
      hadoop fs -mkdir /tmp/hadoop-es
      hadoop fs -put elasticsearch-hadoop-x.x.x.jar /tmp/hadoop-es
      hadoop fs -put commons-httpclient-x.x.jar /tmp/hadoop-es
      ```
3. (Optional) Convert and distribute the security certificate. When connecting to a security-mode Elasticsearch cluster that uses HTTPS, make sure the Java environment trusts the self-signed Elasticsearch certificate.
   a. Obtain the security certificate **CloudSearchService.cer**.
      i. Log in to the **CSS management console**.
      ii. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
      iii. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
      iv. Click the **Overview** tab. In the **Network Information** area, click **Download Certificate** below **HTTPS Access**.

      Figure 2-22 Downloading a security certificate

   b. Convert the security certificate. Upload the downloaded security certificate to the client and use keytool to convert the **.cer** certificate into a **.jks** certificate that can be read by Java.
      - In Linux, run the following command to convert the certificate:
        ```
        keytool -import -alias newname -keystore ./truststore.jks -file ./CloudSearchService.cer
        ```
      - In Windows, run the following command to convert the certificate:
        ```
        keytool -import -alias newname -keystore .\truststore.jks -file .\CloudSearchService.cer
        ```

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

      After this command is executed, you will be prompted to set the certificate password and confirm the password. Securely store the password. It will be used for accessing the cluster.
   c. Distribute the security certificate. Run the **scp** command to distribute the **.jks** certificate file to the same directory on all MRS cluster nodes, for example, **/tmp**. Ensure that user **omm** has the permission to read this file. You can run the following command to set the permission:
      ```
      chown -R omm truststore.jks
      ```
4. Initialize the Hive session. Load the ES-Hadoop driver to the current Hive process.
   a. Log in to the Hive client from the MRS client.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

b. On the Hive client, enter **beeline** or **hive** to go to the execution page and run the following commands:

```
add jar hdfs:///tmp/hadoop-es/commons-httpclient-3.1.jar;
add jar hdfs:///tmp/hadoop-es/elasticsearch-hadoop-x.x.x.jar;
```

This command is valid only for the current session.

5. Create a Hive foreign table and associate it with an Elasticsearch index by mapping the Hive table structure to Elasticsearch index fields.

– For a cluster with the security mode disabled:

```
CREATE EXTERNAL table IF NOT EXISTS student(
  id BIGINT,
  name STRING,
  addr STRING
)
STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'
TBLPROPERTIES(
   'es.nodes' = 'xxx.xxx.xxx.xxx:9200',
   'es.port' = '9200',
   'es.net.ssl' = 'false',
   'es.nodes.wan.only' = 'false',
   'es.nodes.discovery'='true',
   'es.input.use.sliced.partitions'='false',
   'es.resource' = 'student/_doc'
);
```

– For a security-mode cluster that uses HTTP:

```
CREATE EXTERNAL table IF NOT EXISTS student(
  id BIGINT,
  name STRING,
  addr STRING
)
STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'
TBLPROPERTIES(
   'es.nodes' = 'xxx.xxx.xxx.xxx:9200',
   'es.port' = '9200',
   'es.net.ssl' = 'false',
   'es.nodes.wan.only' = 'false',
   'es.nodes.discovery'='true',
   'es.input.use.sliced.partitions'='false',
   'es.nodes.client.only'='true',
   'es.resource' = 'student/_doc',
   'es.net.http.auth.user' = 'username',
   'es.net.http.auth.pass' = 'password'
);
```

– For a security-mode cluster that uses HTTPS:

```
CREATE EXTERNAL table IF NOT EXISTS student(
  id BIGINT,
  name STRING,
  addr STRING
)
STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'
TBLPROPERTIES(
   'es.nodes' = 'https://xxx.xxx.xxx.xxx:9200',
   'es.port' = '9200',
   'es.net.ssl' = 'true',
   'es.net.ssl.truststore.location' = 'certFilePath',
   'es.net.ssl.truststore.pass' = 'certPassword',
   'es.nodes.wan.only' = 'false',
   'es.nodes.discovery'='true',
   'es.nodes.client.only'='true',
   'es.input.use.sliced.partitions'='false',
   'es.resource' = 'student/_doc',
   'es.net.http.auth.user' = 'username',
   'es.net.http.auth.pass' = 'password'
);
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-89 ES-Hadoop parameters

Parameter	Default Value	Description
es.nodes	localhost	Address for accessing the Elasticsearch cluster.
es.port	9200	Port number for accessing the cluster. It is typically **9200**.
es.nodes.wan.only	false	Whether to perform node sniffing.
es.nodes.discovery	true	Whether to enable node discovery. Retain the default setting.
es.input.use.sliced.partitions	true	Whether to use slices. Its value can be: • true • false When this parameter is set to **true**, the prefetch phase may take much longer than the actual query. You are advised to set it to **false** to improve query efficiency.
es.resource	NA	Target index and type.
es.net.http.auth.user	NA	Username for accessing the cluster. Set this parameter only if the security mode is enabled.
es.net.http.auth.pass	NA	Password of the user. Set this parameter only if the security mode is enabled.
es.net.ssl	false	Whether SSL is enabled. The value can be: • **true**: SSL is enabled, and the security certificate is required. • **false**: SSL is disabled.
es.net.ssl.truststore.location	NA	Path of the .jks certificate file, for example, **file:///tmp/truststore.jks**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

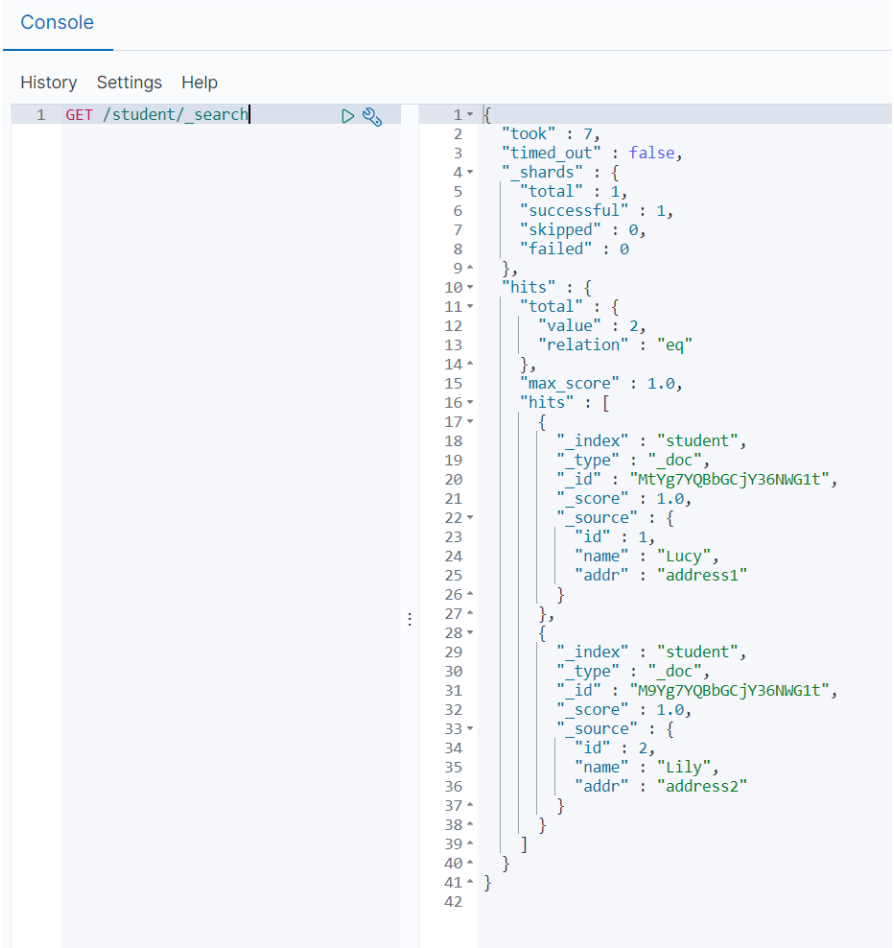
Parameter	Default Value	Description
es.nodes.client.only	false	Whether the Elasticsearch cluster has dedicated client nodes. The value can be: • **true**: The cluster has dedicated client nodes. • **false**: The cluster does not have dedicated client nodes.
es.net.ssl.truststore.pass	NA	Password of the .jks certificate file.

For details about ES-Hadoop configuration items, see the **official configuration description**.

6. Verify the result.
   a. Insert data on the Hive client.
      ```
      INSERT INTO TABLE student VALUES (1, "Lucy", "address1"), (2, "Lily", "address2");
      ```
   b. Query data on the Hive client.
      ```
      select * from student;
      ```
      The query result is as follows:
      ```
      +-------------+---------------+---------------+
      | student.id  | student.name  | student.addr  |
      +-------------+---------------+---------------+
      | 1           | Lucy          | address1      |
      | 2           | Lily          | address2      |
      +-------------+---------------+---------------+
      2 rows selected (0.116 seconds)
      ```
   c. On the Kibana console, search for the data above and verify data persistence.
      i. In the Elasticsearch cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
      ii. In the left navigation pane on Kibana, choose **Dev Tools**.
      The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.
      iii. Run the following query command and check the result.
      ```
      GET /student/_search
      ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 2-23 Kibana query result



2.7.9 Connecting to a Cluster Using cURL Commands

To query or analyze data in Elasticsearch, you often need to connect to your cluster from an external client (such as an ECS). This topic describes how to use cURL commands to access an Elasticsearch cluster.

Prerequisites

- The target Elasticsearch cluster is available.
- An ECS has been created, and it is in the same VPC and security group as the CSS cluster.
  - If they are not in the same security group, modify the ECS security group, or configure the inbound and outbound rules of the group to allow all access from the cluster. For details, see .
  - For details about how to use an ECS, see .

Accessing a Cluster

1. Obtain the private network address of the cluster. It is used to access the cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

This topic uses a private IP address as an example to describe how to access a cluster. The cluster access address varies with the network configurations used. For details, see **Obtaining the Cluster Access Address**.

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

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

c. In the cluster list, obtain the target cluster's internal network address from the **Internal Network Address** column. Typical address format: *<host>*:*<port>* or *<host>*:*<port>*,*<host>*:*<port>*. Example: **10.62.179.32:9200,10.62.179.33:9200**.

   If the cluster has client nodes, only the IP addresses and ports of all the client nodes are displayed. Otherwise, the IP addresses and ports of all data nodes and cold data nodes are displayed.

2. Run one of the following commands on the ECS to access the cluster. The access command varies according to the security mode of the cluster.

   - Cluster in non-security mode
     ```
     curl "http://<host>:<port>"
     ```
   - Cluster in security mode + HTTP
     ```
     curl -u <user>:<password> "http://<host>:<port>"
     ```
   - Cluster in security mode + HTTPS
     ```
     curl -u <user>:<password> -k "https://<host>:<port>"
     ```

Table 2-90 Variables

Variable	Description
host	IP address of each node in the cluster. If the cluster contains multiple nodes, there will be multiple IP addresses. You can use any of them.
port	Port number for accessing a cluster node. Generally, the port number is 9200.
user	Username for accessing the cluster.
password	Password of the user. If the password contains special characters, enclose the username and password in single quotation marks, for example, **curl -u "user:password!" "http://<host>:<port>"**.

An access example is as follows:

```
curl "http://10.62.176.32:9200"
```

Information similar to the following is displayed:

```
HTTP/1.1 200 OK
content-type: application/json; charset=UTF-8
content-length: 513

{
  "name" : "xxx-1",
  "cluster_name" : "xxx",
  "cluster_uuid" : "xxx_uuid",
  "version" : {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
    "number" : "7.10.2",
    "build_flavor" : "oss",
    "build_type" : "tar",
    "build_hash" : "unknown",
    "build_date" : "unknown",
    "build_snapshot" : true,
    "lucene_version" : "8.7.0",
    "minimum_wire_compatibility_version" : "6.7.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

For more commands, see the **Elasticsearch documentation**.

2.8 Ingesting Data

2.8.1 Data Ingestion Methods

When building an enterprise-grade search and analytics platform, you may need to ingest large volumes of heterogeneous data distributed across relational databases (such as MySQL and Oracle), messaging systems (such as Kafka), object storage services (such as OBS), and business applications into an Elasticsearch cluster. To meet varying requirements for data latency, data volume, and development costs, CSS Elasticsearch clusters provide multiple data ingestion methods. This topic briefly introduces and compares these methods, including their applicable scenarios and characteristics, to help you select an optimal option.

NOTE

Before ingesting large amounts of data, you can enhance the destination Elasticsearch cluster's ingestion performance to improve throughput. For details, see **Enhancing Data Ingestion Performance**.

Table 2-91 Comparing different data ingestion methods for Elasticsearch clusters

Method	When to Use	Supported Source/Format	Details
CSS Logstash	You want to perform complex data cleaning (filtering) without building and managing your own Logstash server. For example, you may need to clean Nginx logs from Kafka before ingesting them into Elasticsearch.	MySQL, Kafka, and OBS	**Using Logstash to Synchronize Data to Elasticsearch**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Method	When to Use	Supported Source/Format	Details
Cloud Data Migration (CDM)	You need to perform a full migration of historical data. This method is code-free and guided by a step-by-step wizard. For example, you may ingest years of archived logs from OBS or historical orders from Oracle into Elasticsearch.	OBS (JSON/CSV), Oracle, and MySQL	**Migrating Data Using CDM**
Open-source Logstash	You want to migrate data from an on-premises IDC to Elasticsearch clusters on the cloud, during which you may use special plugins or deeply customized pipeline logic. For example, you may upload system logs from an on-premises data center to Elasticsearch via an SSH tunnel.	Any source supported by Logstash input, such as JSON, CSV, and text	**Ingesting Data Using Self-Managed Logstash**
Open-source APIs	You need to quickly write small amounts of data to Elasticsearch during development or debugging. For example, you can have Java/Python applications directly call Elasticsearch APIs to write data.	JSON	**Ingesting Data Using Open-Source Elasticsearch APIs**

2.8.2 Migrating Data Using CDM

Enterprises often operate heterogeneous platforms, with data distributed across traditional relational databases such as Oracle and storage systems like Object Storage Service (OBS). To enable high-performance full-text search and analytics, this data often needs to be migrated into Elasticsearch. However, migrating large volumes of heterogeneous data using custom scripts is costly and often hindered

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

by complex network configurations and low transfer efficiency. The Cloud Data Migration (CDM) service simplifies this process with a no-code, managed service that supports high concurrency and resumable transfers. Using CDM, you can quickly and securely migrate data from Oracle databases or OBS to CSS Elasticsearch clusters.

Table 2-92 Ingesting data into CSS using CDM

Scenario	Source Data	Target Cluster
Ingesting data from an Oracle database into CSS	A local or third-party Oracle database	Elasticsearch 5.5, 6.2, 6.5, 7.1, 7.6, 7.9, or 7.10
Ingesting data from OBS to CSS	JSON/CSV files in OBS buckets	Elasticsearch 5.5, 6.2, 6.5, 7.1, 7.6, 7.9, or 7.10

Preparations

1. Connect the network.
   - Ensure the CDM cluster, CSS cluster, and OBS bucket reside in the same VPC. This enables data transmission over the internal network for optimal speed.
   - If the source is an on-premises Oracle database, make sure CDM can access it through a VPN, Direct Connect, or public IP address.
2. Obtain connection information
   - Obtain the private network address (for example, **192.168.xxx.xxx:9200**), username, and password of the CSS cluster. (The username and password are required only for a security-mode cluster.)
   - If the source is an Oracle database, obtain the database's IP address, port, database name, username, and password.
   - If the source is OBS, obtain the OBS bucket name, domain name, port, AK, and SK.

Migrating Data

1. Log in to Kibana and go to the command execution page.
   a. Log in to the **CSS management console**.
   b. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
   c. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
   d. In the left navigation pane, choose **Dev Tools**.

      The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.
2. (Optional) Create the destination index in an Elasticsearch cluster. CDM can automatically create destination indexes. However, to ensure optimal query performance, you are advised to define the index mappings in the destination Elasticsearch cluster in advance.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

For example, run the following command to create index **demo**:

- For Elasticsearch 7.x or later:

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

- For Elasticsearch earlier than 7.x:

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

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

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

3. Ingest data from Oracle or OBS to the Elasticsearch cluster using CDM.
   - If the data source is an Oracle database, see **Migrating Data from Oracle to CSS** for an operation guide.
   - If the data source is OBS, see **Migrating Data from OBS to CSS** for an operation guide.

   NOTE

   If the data source is MySQL, see **Creating a MySQL Connector** for how to connect to a MySQL database.

4. After the migration is complete, verify data integrity.
   a. Log in to the Elasticsearch cluster via Kibana, and navigate to the **Dev Tools** page.
   b. Run the following command to check the newly ingested data:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
GET demo/_count      # Check the number of records ingested.
GET demo/_search     # Check the content of the ingested data.
```

If the results are consistent with the source, data ingestion is successful.

2.8.3 Ingesting Data Using Self-Managed Logstash

Enterprises typically generate large volumes of service logs (in JSON or CSV formats), often distributed across local servers or third-party cloud platforms. For centralized search and real-time analytics, you may choose to aggregate such data into CSS Elasticsearch clusters. However, for security reasons, CSS clusters are typically deployed within private networks (for example, VPCs), which restrict direct ingestion from external data sources. This topic describes how to use open-source Logstash to securely and efficiently ingest data into CSS clusters in both private and public network environments.

Solution

Logstash is an open-source, server-side data processing pipeline that ingests data from multiple sources simultaneously, processes and transforms the data, and then sends it to CSS clusters. For more information about Logstash, see the official document **Getting started with Logstash**.

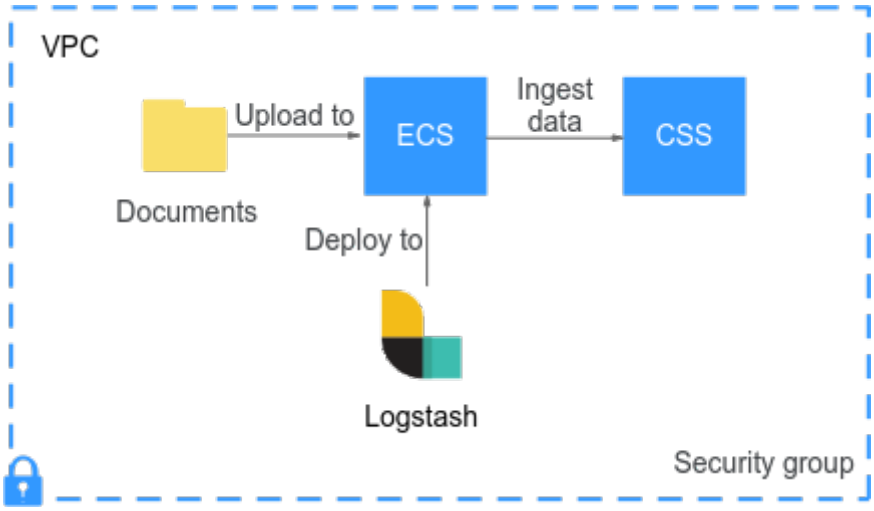
Depending on where the Logstash server is deployed, this solution can use either of the following network architectures:

- Network architecture 1: Direct connection within a VPC (recommended)

  Logstash is deployed on an ECS that is in the same VPC as the destination CSS cluster.

  Characteristics: fast, easy to set up, and secure.

  Figure 2-24 Logstash and CSS within the same VPC

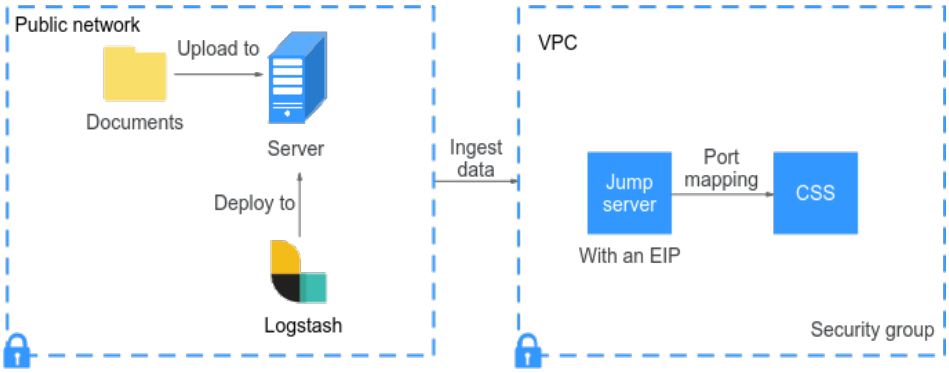
- Network architecture 2: SSH connection over the public network

  Logstash is deployed in an on-premises data center or on a server located on the public network, which prevents it from directly accessing the destination CSS cluster in a private network. To connect the two, we need to use an ECS allocated with a public IP address as a jump server. This jump server leverages SSH port forwarding to forward traffic coming from Logstash to the destination cluster on the cloud.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Characteristics: suitable for migrating on-premises data to the cloud; port forwarding must be configured.

Figure 2-25 Logstash deployed on premises or in the public network



Prerequisites

- The server that deploys Logstash has JDK 1.8 or later installed. A Linux server is recommended.
- You have downloaded the OSS version (pure open-source) of Logstash. The Logstash version should be kept consistent with the CSS Elasticsearch cluster version (for example, both are 7.10.2).

  Download Logstash OSS at **https://www.elastic.co/downloads/past-releases?product=logstash-oss**.
- You have obtained the connection information for both the data source and destination, including IP addresses, port numbers, accounts, and passwords.
- The source data is ready for ingestion.

  For example, the **access_20181029.log** is already in the **/tmp/access_log/** directory on the Logstash server. The following is an example of the 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 |
```

Scenario 1: Ingesting Data When Logstash and CSS Reside in the Same VPC

1. Prepare the network: Make sure the ECS that hosts Logstash and the destination CSS cluster are within the same VPC, and that the VPC's security group allows port 9200.

   Ping the CSS cluster's private network address from the ECS where you plan to deploy Logstash. If successful, the two are connected.
2. Install Logstash on the ECS. The following provides an installation guide for Red Hat Linux. For other operating systems, see **Installing Logstash**.

   a. Use yum to install the JDK first. This is because Logstash depends on Java.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
yum install java  # Replace it with the actual JDK version.
```

b. Upload the downloaded Logstash package to the ECS.

c. Use yum to install Logstash.

```
yum install logstash-oss-7.10.0-x86_64.rpm
```

Replace *logstash-oss-7.10.0-x86_64.rpm* with the actual Logstash installation package name.

a. Modify the JVM configuration of Logstash to accelerate cluster data migration.

The default heap memory of Logstash is 1 GB. You are advised to change it to half the cluster node memory.

```
vim /etc/logstash/jvm.options
-Xms4g
-Xmx4g
```

3. (Optional) Prepare the security certificate. If the security mode and HTTPS are enabled for the destination CSS cluster, you must upload the cluster's security certificate to the ECS.

a. **Obtaining the Security Certificate**.

b. Upload the downloaded security certificate to the ECS.

4. Create a configuration file in the Logstash installation directory.

a. Create a configuration file, for example, **logstash-simple.conf**.

```
cd /logstash-x.x/ # Go to the Logstash installation directory.
vi logstash-simple.conf     # Create a configuration file.
```

b. Define input, filter, and output in the configuration file.

```
input {
    # Read local log files.
    file {
        path => "/tmp/access_log/*.log" # Path to the log files
        start_position => "beginning"   # Start position for reading data
    }
}

filter {
    # Add data filtering logic here, such as grok and mutate.
}

output {
    elasticsearch {
        # Enter the cluster's private network address. For a multi-cluster node, enter multiple addresses to allow for load balancing.
        hosts => ["192.168.xxx.xxx:9200", "192.168.xxx.xxx:9200"]
        # Name of the index that events are written to.
        index => "my_import_index"
        # Configure the following for a security-mode cluster. (Delete them if security mode is disabled.)
        # user => "username"           # CSS cluster account
        # password => "password"       # CSS cluster password
        # ssl => true                  # Whether SSL is enabled
        # ssl_certificate_verification => false              # Whether to enable certificate authentication
        # cacert => "/logstash/config/CloudSearchService.cer" # Absolute path to the security certificate
        # Commonly configured parameters to mitigate compatibility issues
        ilm_enabled => false             # Disable index lifecycle management to prevent permission errors.
        manage_template => false         # Disable template management.
    }
}
```

c. After the configuration is complete, enter **:wq** to save the configuration file.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

5. Execute the configuration file to ingest data into the CSS cluster.
   ```
   ./bin/logstash -f logstash-simple.conf     # Replace the file name with the actual configuration file name.
   ```
   If a message similar to **Successfully started Logstash API endpoint** is returned and no error is reported, the task is running properly.
6. After ingestion, verify data integrity.
   a. Log in to the **CSS management console**.
   b. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
   c. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
   d. In the left navigation pane on Kibana, choose **Dev Tools**.
      The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.
   e. Run the following command to check the newly ingested data:
      ```
      GET my_import_index/_count      # Check the number of records ingested.
      GET my_import_index/_search     # Check the content of the ingested data.
      ```
      If the results are consistent with the source, data ingestion is successful.

Scenario 2: Ingesting Data When Logstash Is Deployed on the Public Network

1. Prepare a jump server. In the VPC where the CSS cluster is located, prepare a Linux ECS allocated with an EIP, and configure security group rules to allow a local port (for example, 9100).
2. Establish an SSH tunnel.
   a. On the jump server, run the following command to configure port mapping, which redirects requests sent to the opened port on the jump server to the destination cluster:
      ```
      ssh -g -L <Local port on the jump server>:<Private IP address of a CSS cluster node>:9200 -N -f root@<The jump server's private network address>
      ```
      For example, if the opened local port (allowed by security group rules) on the jump server is 9100, the CSS cluster node's private IP address is 192.168.0.81, and the jump server's private IP address is 192.168.0.227, the port mapping command is as follows:
      ```
      ssh -g -L 9100:192.168.0.81:9200 -N -f root@192.168.0.227
      ```
      NOTE
      If the specified CSS cluster node is unavailable, the port mapping command will fail. If the cluster contains multiple nodes, replace <Private IP address of a CSS cluster node> with the IP address of another node. If the cluster contains only one node, you have to wait until the node recovers.
      For more information about SSH tunneling, see **SSH official documents**.
   b. Verify connectivity.
      ```
      curl http://<The jump server's public IP address>:<The jump server's local port>
      ```
      If the Elasticsearch version information is returned, the tunnel is established. When you configure the Logstash configuration file later, set **hosts** to <The jump server's public IP address>:<The jump server's local port>.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3. (Optional) Prepare the security certificate. If the security mode and HTTPS are enabled for the destination CSS cluster, you must upload the cluster's security certificate to the ECS.
   a. **Obtaining the Security Certificate**.
   b. Upload the downloaded security certificate to the ECS.
4. (Optional) Install Logstash on a server deployed on an external network. The following provides an installation guide for Red Hat Linux.
   a. Use yum to install the JDK first. This is because Logstash depends on Java.
      ```
      yum install java  # Replace it with the actual JDK version.
      ```
   b. Upload the downloaded Logstash package to the ECS.
   c. Use yum to install Logstash.
      ```
      yum install logstash-oss-7.10.0-x86_64.rpm
      ```
      Replace *logstash-oss-7.10.0-x86_64.rpm* with the actual Logstash installation package name.
   a. Modify the JVM configuration of Logstash to accelerate cluster data migration.
      The default heap memory of Logstash is 1 GB. You are advised to change it to half the cluster node memory.
      ```
      vim /etc/logstash/jvm.options
      -Xms4g
      -Xmx4g
      ```
5. Create a configuration file in the Logstash installation directory.
   a. Create a configuration file, for example, **logstash-simple.conf**.
      ```
      cd /logstash-x.x/ # Go to the Logstash installation directory.
      vi logstash-simple.conf     # Create a configuration file.
      ```
   b. Define input, filter, and output in the configuration file.
      ```
      input {
          # Read local log files.
          file {
              path => "/tmp/access_log/*.log" # Path to the log files
              start_position => "beginning"   # Start position for reading data
          }
      }

      filter {
          # Add data filtering logic here, such as grok and mutate.
      }

      output {
          elasticsearch {
              # Enter the local address for the SSH tunnel.
              hosts => ["<The jump server's public IP address>:<The jump server's local port>"]
              # Name of the index that events are written to.
              index => "my_import_index"
              # Configure the following for a security-mode cluster. (Delete them if security mode is disabled.)
              # user => "username"          # CSS cluster account
              # password => "password"      # CSS cluster password
              # ssl => true                 # Whether SSL is enabled
              # ssl_certificate_verification => false             # Whether to enable certificate authentication
              # cacert => "/logstash/config/CloudSearchService.cer" # Absolute path to the security certificate
              # Commonly configured parameters to mitigate compatibility issues
              ilm_enabled => false            # Disable index lifecycle management to prevent permission errors.
      ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
    manage_template => false        # Disable template management.
  }
}
```

c. After the configuration is complete, enter **:wq** to save the configuration file.

6. Execute the configuration file to ingest data into the CSS cluster.

```
./bin/logstash -f logstash-simple.conf     # Replace the file name with the actual configuration file name.
```

If a message similar to **Successfully started Logstash API endpoint** is returned and no error is reported, the task is running properly.

7. After ingestion, verify data integrity.

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

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

c. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.

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

The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

e. Run the following command to check the newly ingested data:

```
GET my_import_index/_count      # Check the number of records ingested.
GET my_import_index/_search     # Check the content of the ingested data.
```

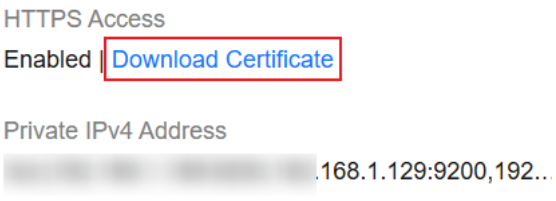
If the results are consistent with the source, data ingestion is successful.

Obtaining the Security Certificate

To access a security-mode Elasticsearch cluster that uses HTTPS, perform the following steps to obtain the security certificate CloudSearchService.cer if it is required.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Overview** tab. In the **Network Information** area, click **Download Certificate** below **HTTPS Access**.

Figure 2-26 Downloading a security certificate



2.8.4 Ingesting Data Using Open-Source Elasticsearch APIs

During application development, debugging, or small-scale data migration, developers frequently need to quickly write local JSON files, such as test datasets and configuration files, to an Elasticsearch cluster. Compared with data migration

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

tools such as Logstash and CDM, which are complex to configure, open-source Elasticsearch APIs (such as the _bulk API) provide a more flexible and lightweight alternative, especially for ad hoc tasks. You can write data through Kibana in a highly interactive manner, or run cURL commands on an ECS for batch ingestion. Both methods enable agile and efficient data ingestion into cloud-based Elasticsearch clusters.

Method 1: Ingesting Data in Kibana

On Kibana, you can run POST commands to import data using an open-source Elasticsearch API.

Applicable scenarios: development and debugging, small ad-hoc data writes, and syntax verification.

1. Log in to Kibana.
   a. Log in to the **CSS management console**.
   b. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
   c. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
   d. In the left navigation pane, choose **Dev Tools**.

      The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.
2. (Optional) Create an index and specify a custom mapping to define field data types.

   For example, run the following command to create index **my_store**:

   – For Elasticsearch 7.x or later:

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

   – For Elasticsearch earlier than 7.x:

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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        }
      }
    }
  }
}
```

3. Write data to the index.

   For example, run the following command to write a record to the **my_store** index:

   - For Elasticsearch 7.x or later:

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

   - For Elasticsearch earlier than 7.x:

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

4. Verify the result.

   Check the result. If the value of **errors** is **false** and the value of **result** in each record in the **items** array is **created**, all records are successfully written.

Method 2: Ingesting Data by Running cURL Commands on an ECS

On an ECS server, you can run cURL commands to import files via an open-source Elasticsearch API.

Applicable scenarios: batch ingestion of local JSON files using an automation script.

1. Prepare an ECS. Buy an ECS located in the same VPC as the destination Elasticsearch cluster.

   For how to buy and use an ECS, see **Purchasing and Logging In to a Linux ECS**.

2. Obtain the private network address of the cluster.

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

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

   c. In the cluster list, obtain the target cluster's internal network address from the **Internal Network Address** column. Typical address format: *<host>*:*<port>* or *<host>*:*<port>*,*<host>*:*<port>*. Example: **10.62.179.32:9200,10.62.179.33:9200**.

      If the cluster has client nodes, only the IP addresses and ports of all the client nodes are displayed. Otherwise, the IP addresses and ports of all data nodes and cold data nodes are displayed.

3. Verify connectivity. Ping the Elasticsearch cluster's private network address from the ECS. If successful, the two are connected.

4. Prepare documents. Upload JSON files to the ECS.

   For example, save the following data as a **test.json** file and upload it to the ECS:

   - For Elasticsearch 7.x or later:

     ```
     {"index": {"_index":"my_store"}}
     {"productName":"Autumn new woman blouses 2019","size":"M"}
     {"index": {"_index":"my_store"}}
     {"productName":"Autumn new woman blouses 2019","size":"L"}
     ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- For Elasticsearch earlier than 7.x:

```
{"index": {"_index":"my_store","_type":"products"}}
{"productName":"Autumn new woman blouses 2019","size":"M"}
{"index": {"_index":"my_store","_type":"products"}}
{"productName":"Autumn new woman blouses 2019","size":"L"}
```

5. (Optional) Create an index and specify a custom mapping to define field data types.

   For example, run the following command to create index **my_store**:

   Sample code for Elasticsearch 7.x or later:

   - For a non-security mode cluster (HTTP): no authentication required.

```
curl -X PUT "http://<host>:<port>/my_store" \
    -H 'Content-Type: application/json' \
    -d '{
      "settings": {
        "number_of_shards": 1
      },
      "mappings": {
        "properties": {
          "productName": {
            "type": "text"
          },
          "size": {
            "type": "keyword"
          }
        }
      }
    }'
```

   - For a security-mode cluster using HTTP: Authentication is required. If the password contains special characters like **!**, enclose the password with a pair of single quotation marks, for example, **-u 'admin':'Pass!word'**.

```
curl -X PUT -u <user>:<password> "http://<host>:<port>/my_store" \
    -H 'Content-Type: application/json' \
    -d '{
      "settings": {
        "number_of_shards": 1
      },
      "mappings": {
        "properties": {
          "productName": {
            "type": "text"
          },
          "size": {
            "type": "keyword"
          }
        }
      }
    }'
```

   - For a security-mode cluster using HTTPS: Authentication is required. If the password contains special characters like **!**, enclose the password with a pair of single quotation marks, for example, **-u 'admin':'Pass!word'**. Furthermore, ignore SSL certificate authentication.

```
curl -X PUT -k -u <user>:<password> "https://<host>:<port>/my_store" \
    -H 'Content-Type: application/json' \
    -d '{
      "settings": {
        "number_of_shards": 1
      },
      "mappings": {
        "properties": {
          "productName": {
            "type": "text"
          },
          "size": {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
          "type": "keyword"
        }
      }
    }
  }'
```

Sample code for Elasticsearch earlier than 7.x:

- For a non-security mode cluster (HTTP): no authentication required.

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

- For a security-mode cluster using HTTP: Authentication is required. If the password contains special characters like **!**, enclose the password with a pair of single quotation marks, for example, **-u 'admin':'Pass!word'**.

```
curl -X PUT -u <user>:<password> "http://<host>:<port>/my_store" \
    -H 'Content-Type: application/json' \
    -d '{
      "settings": {
        "number_of_shards": 1
      },
      "mappings": {
        "products": {
          "properties": {
            "productName": {
              "type": "text"
            },
            "size": {
              "type": "keyword"
            }
          }
        }
      }
    }'
```

- For a security-mode cluster using HTTPS: Authentication is required. If the password contains special characters like **!**, enclose the password with a pair of single quotation marks, for example, **-u 'admin':'Pass!word'**. Furthermore, ignore SSL certificate authentication.

```
curl -X PUT -k -u <user>:<password> "https://<host>:<port>/my_store" \
    -H 'Content-Type: application/json' \
    -d '{
      "settings": {
        "number_of_shards": 1
      },
      "mappings": {
        "products": {
          "properties": {
            "productName": {
              "type": "text"
            },
            "size": {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
          "type": "keyword"
        }
      }
    }
  }
}'
```

6. Ingest the JSON file into the Elasticsearch cluster:

   Run the following command in the ECS directory that stores the JSON file:

   - For a non-security mode cluster (HTTP): no authentication required.
     ```
     curl -X POST "http://<host>:<port>/_bulk" \
         -H 'Content-Type: application/json' \
         --data-binary @test.json
     ```
   - For a security-mode cluster using HTTP: Authentication is required. If the password contains special characters like **!**, enclose the password with a pair of single quotation marks, for example, **-u 'admin':'Pass!word'**.
     ```
     curl -X POST -u <user>:<password> "http://<host>:<port>/_bulk" \
         -H 'Content-Type: application/json' \
         --data-binary @test.json
     ```
   - For a security-mode cluster using HTTPS: Authentication is required. If the password contains special characters like **!**, enclose the password with a pair of single quotation marks, for example, **-u 'admin':'Pass!word'**. Furthermore, ignore SSL certificate authentication.
     ```
     curl -X POST -k -u <user>:<password> "https://<host>:<port>/_bulk" \
         -H 'Content-Type: application/json' \
         --data-binary @test.json
     ```

   NOTE

   If the designated cluster node is unavailable, the command will fail. If the cluster contains multiple nodes, replace <host> with the IP address of another node. If the cluster contains only one node, you have to wait until the node recovers.

7. Verify the result.

   After data ingestion, the following information is returned:
   ```
   {"took":204, "errors":false, "items":[...]}
   ```

   If the value of **errors** is **false**, all data has been successfully ingested.

2.9 Managing the Index Lifecycle

2.9.1 Configuring Index State Management (ISM) Policies

As a business expands, its time series data, such as logs and metrics, can grow exponentially. Storage costs can increase dramatically, and over-sized indexes can lead to poor query performance. To prevent these issues, operations teams traditionally needed to manually create new indexes, migrate old data, and delete obsolete data on a daily basis, which can be a tedious and error-prone process. Index lifecycle automation addresses these challenges by automatically managing index state transitions. This ensures read and write performance for hot data, reduces cold storage costs, and automatically deletes obsolete data. CSS Elasticsearch clusters provide the Index State Management (ISM) plugin for this purpose. By configuring ISM policies, you can automate key lifecycle operations, for example, to automatically roll over an index when it reaches 50 GB, transition it to cold storage when it is 30 days old, and delete it when it is 90 days old. This helps reduce costs and improve efficiency.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Additionally, CSS provides advanced ISM features such as skipping auto rollover for empty indexes and automatic retry of failed ISM tasks. These features further reduce the operational workload and enhance ISM task reliability.

- ISM: the Elasticsearch plugin that automates index lifecycle management. For more information, see the official document **Index State Management**.
- Policy: a set of rules for index lifecycle management. An example of a policy: If the number of documents in an index exceeds 10 million, the index rolls over.
- Rollover: When a write index grows beyond a specified size, age, or number of documents, a new index is automatically created to serve as the new write index. The old index changes to read-only or an archived state based on the ISM policy.

Constraints

- Only Elasticsearch clusters 7.6.2 or later support the ISM plugin.
- The Kibana GUI varies depending on the Kibana version. Kibana 7.10.2 is used as an example here.

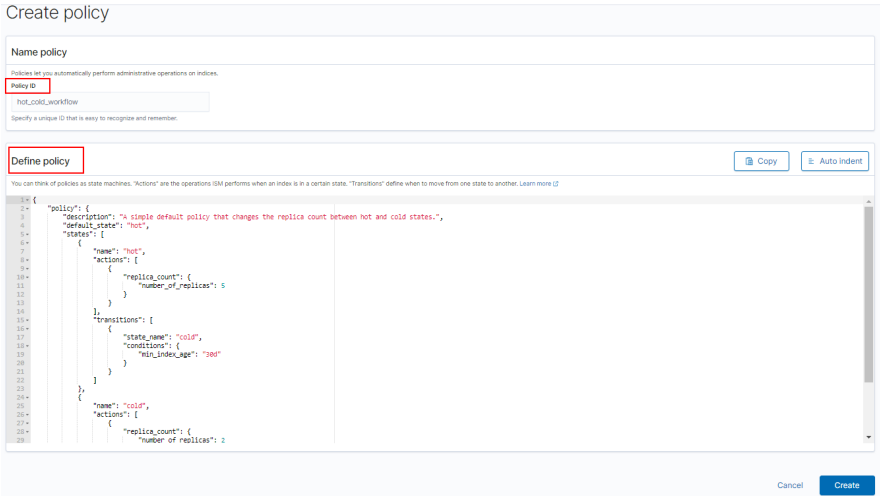
Step 1: Creating an ISM Policy

Define one or more ISM policies on Kibana to tell the cluster how to manage indexes.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
4. On the Kibana console, expand the menu in the upper-left corner, and choose **IM** or **Index Management** to go to the index management page.
5. Select **Index Policies**.
6. Click **Create policy** to configure an index lifecycle policy.
   - **Policy ID**: Enter a custom policy name.
   - **Define policy**: Define the policy in JSON format. You can define a policy by modifying the default system example.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 2-27 Creating a policy



7. Click **Create**.

Step 2: Associating a Policy with Indexes

After an ISM policy is created, associate it with specific indexes to apply it. You are advised to associate a policy with an index template. This ensures that the policy is automatically applied to all indices created based on this template.

- **Method 1: Associate a policy with an index template (recommended)**

  Use this method for indexes that store time series data.

  a. In the left navigation pane on Kibana, choose **Dev Tools**.

     The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

  b. Run the following command to create or update an index template:

```
PUT _template/<template_name>
{
  "index_patterns": ["index_name-*"],  // Match all indexes whose name starts with index_name-.
  "settings": {
    "opendistro.index_state_management.policy_id": "Policy_ID" // Replace it with the target policy ID.
  }
}
```

     For more information, see **Create template**.

  c. On the **Managed Indices** page, check for newly-created indexes whose name starts with **index_name-***, and check whether their **Policy** is the associated policy ID.

- **Method 2: Manually associating an existing index with a policy**

  Use this method to associate a policy with existing indexes that are currently not associated with any policy.

  a. On the Kibana console, expand the menu in the upper-left corner, and choose **IM** or **Index Management** to go to the index management page.

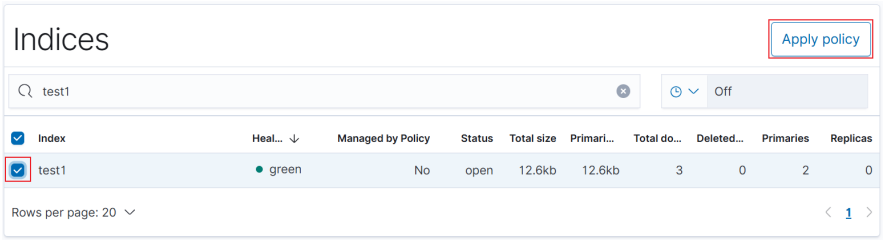
  b. Click **Indices**.

  c. In the **Indices** list, select one or more indexes.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

d. Click **Apply policy** in the upper right corner. In the displayed dialog box, select a policy ID.

Figure 2-28 Apply policy



e. Click **Apply**.

f. On the **Managed Indices** page, locate the indexes and make sure their **Policy** is the associated policy ID.

After you associate a policy with indexes, ISM automatically creates a job that runs every 5 minutes to execute the policy, check criteria, and change index states based on the policy.

Step 3: Managing ISM Policies

To manage deployed ISM policies, choose **Index Management** > **Managed Indices** on Kibana.

- View status: In the **Managed Indices** list, check the policy associated with each index and the policy status.
- Retry policy: If the policy associated with an index is not working properly, select this index and click **Retry policy**.
- Change policy: To update the policy for an index, click Change policy.
- Remove policy: To stop lifecycle management for an index, click Remove policy.

(Advanced Feature) Disabling Auto Rollover for Empty Indexes

A lifecycle policy can be configured to enable automatic index rollover, that is, automatically creating a new index when an existing index meets certain conditions (for example, when it reaches a predefined age in terms of days). The new index then receives subsequent writes. By default, even if an existing index contains no documents, new indexes are still created. Over time, this can result in a large number of empty indexes. To prevent this, CSS allows you to disable automatic rollover for empty indexes. This is achieved by configuring a lifecycle policy to allow automatic rollover only for indexes that contain documents. By doing so, you can prevent excessive empty indexes from being created, thus optimizing storage utilization and enhancing overall system efficiency.

This feature is supported only when an Elasticsearch cluster meets the following conditions:

- Cluster version: 7.6.2 or 7.10.2
- Image version: x.x.x_25.9.0_xxx or later

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

You can disable automatic rollover for empty indexes at either the index level or the cluster level. Index-level settings take precedence over cluster-level settings. **You must configure this manually.** The commands are as follows:

- Disable empty index automatic rollover for a **specified index**:

```
PUT {index_name}/_settings
{
  "index.plugins.index_state_management.rollover.only_if_has_documents": true
}
```

- Disable empty index automatic rollover at the **cluster** level:

```
PUT _cluster/settings
{
  "persistent": {
    "plugins.index_state_management.rollover.only_if_has_documents": true
  }
}
```

(Advanced Feature) Configuring Auto Retry for ISM Tasks

ISM tasks (such as converting hot data to cold data and deleting expired indexes) can fail due to transient cluster issues like resource constraints, node restarts, or network interruptions. To address this, CSS automatically retries (reactivates) failed ISM tasks at configured intervals until they succeed. This ensures the continuity and reliability of ISM tasks.

This feature is supported only when an Elasticsearch cluster meets the following conditions:

- Cluster version: 7.6.2 or 7.10.2
- Image version: x.x.x_25.9.0_xxx or later

Automatic retry of ISM tasks is enabled by default for Elasticsearch clusters that meet the conditions above.

For Elasticsearch clusters that do not meet these conditions, upgrade them to enable this feature. However, ISM tasks that have already failed before the upgrade will not be retried after the cluster is upgraded. You will need to manually retry them before the automatic retry can occur. The command for retrying failed ISM tasks is as follows:

```
POST _opendistro/_ism/retry/{index_name}
```

Run the following command to modify automatic retry settings for index lifecycle management tasks:

```
PUT _cluster/settings
{
  "persistent": {
    "plugins.index_state_management.coordinator.css.reactivate": true,
    "plugins.index_state_management.coordinator.css.reactivate_period": "5m",
    "plugins.index_state_management.coordinator.css.reactivate_duration": "30m",
    "plugins.index_state_management.coordinator.css.reactivate_max_duration": "12h",
    "plugins.index_state_management.coordinator.css.max_inflight_reactivate_tasks": 1000
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-93 Configuration parameters

Parameter	Type	Default Value	Description
plugins.index_state_management.coordinator.css.reactivate	Boolean	true	Whether to enable the Reactivate (auto retry) feature. • **true**: Enables automatic retry of index lifecycle management tasks. • **false**: Disables automatic retry of automatic retry of index lifecycle management tasks.
plugins.index_state_management.coordinator.css.reactivate_period	Time	30m	Reactivate interval. Value range: ≥ 5m
plugins.index_state_management.coordinator.css.reactivate_duration	Time	1h	Initial Reactivate interval, that is, the wait time for another retry after the initial attempt fails. It is not affected by the exponential backoff mechanism. Value range: ≥ 1m
plugins.index_state_management.coordinator.css.reactivate_max_duration	Time	24h	Maximum Reactivate interval, that is, the maximum wait time between retry attempts regardless of the effect of the exponential backoff mechanism. This parameter ensures that the retry interval does not increase indefinitely. Value range: ≥ 1m
plugins.index_state_management.coordinator.css.max_inflight_reactivate_tasks	Long	10000	Maximum number of tasks that can be retried at the same time. Value range: 1 to 100000

Idempotence considerations for ISM task retry

- Idempotent operations: All steps are retried, starting from the beginning, to ensure data consistency.
- Non-idempotent operations: The retry begins from where the failure occurred. If the original failure cause persists, retries may fail repeatedly, in which case, manual intervention is required.

  Examples of non-idempotent operations in Elasticsearch's index lifecycle management tasks: ForceMerge, Notification, and Snapshot.

In the **ISM task details API**, CSS has added fields that record ISM task execution details. You can run the following command to check ISM task records, including

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

the total number of failures, failure occurrence time, activation time, and failure causes:

```
GET /_opendistro/_ism/explain/{index_name}
```

Example response:

```
{
  "rollover-0000001" : {
    "index.opendistro.index_state_management.policy_id" : "logs-policy-rollover",
    "index" : "rollover-0000001",
    ...
    "reactivate_info" : {
      "count": 2, //Total number of failures.
      "latest_failed_time": 1764301006792, //Latest failure time.
      "latest_reactivate_time": 1764301486800, //Latest activation time.
      "failed_infos": [  //Historical execution failure records. Only the latest 10 records are retained.
        {
          "start_time": 0, //Task execution time. The earliest execution time is 0.
          "failed_time": 1764299686814, //Task failure time.
          "info": {  //Task failure cause.
            "message" : "Missing rollover_alias index setting [index=rollover-0000001]"
          }
        },
        {
          "start_time" : 1764300286841,
          "failed_time" : 1764301006792,
          "info" : {
            "message" : "Missing rollover_alias index setting [index=rollover-0000001]"
          }
        }
      ]
    },
    ...
  }
}
```

2.9.2 Automating Index Rollover Using ISM

Time series data processing workloads, such as log analytics and real-time monitoring, often involve rapid data growth over time. If data continues to be written to a single index, the index can become excessively large, leading to degraded query performance and increased storage consumption due to obsolete data. Traditionally, managing this growth required custom scripts to split indexes, which was complex and costly. Now, by configuring index state management (ISM) policies for indexes in Elasticsearch clusters, you can automate index rollover based on index size or age and have obsolete indexes deleted automatically. This helps balance query performance and storage costs.

- ISM: a plugin that automates index lifecycle management.
- Rollover: When the size or age of an index reaches a predefined threshold, a new index is automatically created, and writes are redirected to the new index.
- Alias: the logical index name. During writes, the alias always points to the current active index; and during queries, it points to the current active index plus all historical indexes.

Solution

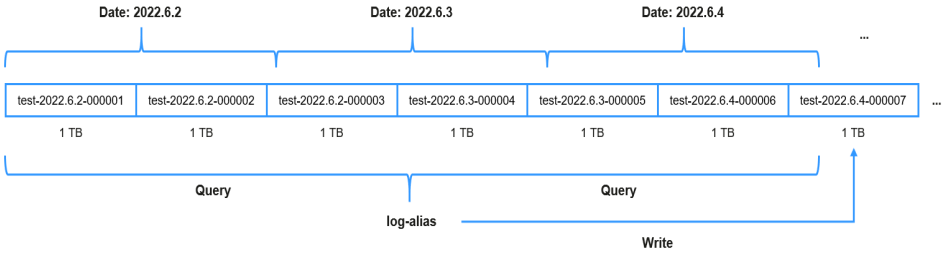
This topic describes how to configure an ISM policy to automate index rollover.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- Hot phase: Indexes handle high-concurrency writes. When the index size reaches 1 TB or its age reaches one day, a new index is automatically created, and writes are redirected to the new index.
- Warm phase: After seven days, index replicas are disabled to save resources.
- Delete phase: After 30 days, the index is automatically deleted to reclaim storage space.

Assume that an application generates around 2.4 TB of logs every day. An ISM policy can be created to optimize log storage and query performance. In this example, we define the index alias as *log-alias*. **Figure 1 log-alias data organization** shows the organization of data in this index. During queries, the alias points to all indexes whose name starts with *test*. During writes, the alias points only to the latest index.

Figure 2-29 log-alias data organization



NOTE

The one day in the rollover time refers to 24 hours following the index creation time, not a calendar day.

Constraints

The Elasticsearch version must be 7.6.2 or later.

Configuring an ISM Policy

1. Log in to Kibana and go to the command execution page.
   a. Log in to the **CSS management console**.
   b. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
   c. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
   d. In the left navigation pane, choose **Dev Tools**.

      The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.
2. Define an ISM policy to automate index rollover, state transitions, and deletion.

   For example, create a policy named "rollover workflow": When the index size reaches 1 TB or the index age reaches one day, it rolls over automatically. When the index age reaches seven days, replicas are disabled. When it reaches 30 days, the index is deleted.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
PUT _opendistro/_ism/policies/rollover_workflow
{
  "policy": {
    "description": "rollover test",
    "default_state": "hot",
    "states": [
      {
        "name": "hot",
        "actions": [
          {
            "rollover": {
              "min_size": "1tb",
              "min_index_age": "1d"
            }
          }
        ],
        "transitions": [
          {
            "state_name": "warm",
            "conditions": {
              "min_index_age": "7d"
            }
          }
        ]
      },
      {
        "name": "warm",
        "actions": [
          {
            "replica_count": {
              "number_of_replicas": 0
            }
          }
        ],
        "transitions": [
          {
            "state_name": "delete",
            "conditions": {
              "min_index_age": "30d"
            }
          }
        ]
      },
      {
        "name": "delete",
        "actions": [
          {
            "delete": {}
          }
        ]
      }
    ]
  }
}
```

After a lifecycle policy is created, run the following command to query the policy details:

```
GET _opendistro/_ism/policies/rollover_workflow
```

3. Associate the policy with an index template so that new indexes created using this template will automatically inherit the ISM policy and alias.

   For example, create an index template named **template_test** so that newly created indexes whose name starts with **test** are automatically associated with the ISM policy **rollover_workflow**, and will use **log_alias** for index rollover.

```
PUT _template/template_test
{
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
 "index_patterns": "test*",
 "settings": {
   "number_of_replicas": 1, // Number of index replicas
   "number_of_shards": 1,    // Number of index shards
   "opendistro.index_state_management.policy_id": "rollover_workflow",    // Index lifecycle policy name
   "index.opendistro.index_state_management.rollover_alias": "log_alias"  // Which alias to roll over
 },
 "mappings": {
   "properties": {
     "name": {
       "type": "text"
     }
   }
 }
}
```

After an index template is created, you can run the following command to query the template details:

```
GET _template/template_test
```

4. Manually create the first index that will start the automatic rollover process.

   For example, create the first index, set the date format, specify **aliases**, and set **is_write_index** to **true**. The index will automatically use the index template *template_test* and inherit the lifecycle policy *rollover_workflow* through the template.

   ```
   PUT %3Ctest-%7Bnow%2Fd%7D-000001%3E
   {
     "aliases": {
       "log_alias": {
         "is_write_index": true // Tells the customer to direct all write requests aiming at log_alias to this index.
       }
     }
   }
   ```

   The index name above is the result of URL-coding **<test-{now/d}-000001>**. By default, the index name contains the creation date. For example, if an index was created on 2022-06-02, the index name is **test-2022.06.02-000001**.

5. Write in data. The alias **log_alias** is used during data writes, and **log_alias** always points to the last index.

   ```
   POST log_alias/_bulk
   {"index":{}}
   {"name":"name1"}
   {"index":{}}
   {"name":"name2"}
   {"index":{}}
   {"name":"name3"}
   {"index":{}}
   {"name":"name4"}
   {"index":{}}
   {"name":"name5"}
   {"index":{}}
   {"name":"name6"}
   ```

6. Query data and check whether the rollover takes effect.

   - One day after the indexes are created, check the indexes starting with **test**.

     ```
     GET _cat/indices/test*?s=i
     ```

     There are supposed to be at least two indexes, for example:

     ```
     green open test-<Date>-000001 r8ab5NX6T3Ox_hoGUanogQ 1 1 6 0 416b 208b
     green open test-<Date>-000002 sfwkVgy8RSSEw7W-xYjM2Q 1 1 0 0 209b 209b
     ```

     In the preceding information, **test-<Date>-000001** is the index created in **4**, and **test-<Date>-000002** is the index generated through rollover.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- To query the index associated with the alias **log_alias**, run the following command:

  ```
  GET _cat/aliases/log_alias?v
  ```

  The alias is supposed to point to multiple indexes, for example:

  ```
  alias     index                filter routing.index routing.search is_write_index
  log_alias test-<Date>-000001     -      -             -            false
  log_alias test-<Date>-000002     -      -             -            true
  ```

FAQ: How Do I Skip Rollover for an Index?

Scenarios

- A manual rollover is performed: A time-based ISM policy was configured to automate index rollover. Before the rollover criteria were met, however, a manual rollover was performed. When the auto rollover time configured by ISM was reached, an automatic rollover was attempted but failed because the index had already been rolled over manually. As a result, the ISM task stopped. To solve this problem, you need to skip the index rollover and then retry the ISM task to resume automatic index rollover.
- For an index that has already been rolled over automatically, a remove or add policy operation is performed: After the rollover policy is modified (through the remove policy or add policy operation), the ISM task restarts from the beginning. When rollover starts, the task fails. To solve this, you also need to skip index rollover and then retry the ISM task to resume automatic index rollover.

WARNING

After rollover is skipped for an index, ISM will no longer attempt rollover or generate rolled-over indexes. This means skipping index rollover may lead to data loss. Please exercise caution.

Constraints

To skip index rollover, the Elasticsearch cluster version must be 7.6.2 or 7.10.2, and the cluster image version must not be earlier than 7.x.2_25.1.0_x.x.x.

Procedure

1. If an ISM task stops due to an index rollover failure, run the following command to skip index rollover:

   ```
   PUT {index_name}/_settings
   {
     "index.plugins.index_state_management.rollover_skip": true
   }
   ```

   If **true** is returned, the configuration is successful.

2. After index rollover is skipped, run the following command to retry the ISM task:

   ```
   POST _opendistro/_ism/retry/{index_name}
   ```

   If the following information is returned, the retry is successful:

   ```
   {
     "updated_indices": 1,
     "failures": false,
     "failed_indices": []
   }
   ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2.9.3 Decoupling Storage and Compute Using ISM

In log analytics and monitoring scenarios, storage costs for historical data accumulate over time. However, most cold data is accessed only occasionally for auditing or backtracking purposes. Storing such data on expensive local SSDs over extended periods does not make economic sense. To balance storage costs with long-term retention requirements, CSS provides a storage-compute decoupling feature for Elasticsearch clusters. By configuring an index state management (ISM) policy, coupled with index freezing, you can have eligible indexes automatically transitioned from local SSDs to lower-cost object storage. The data remains searchable, although query latency may increase.

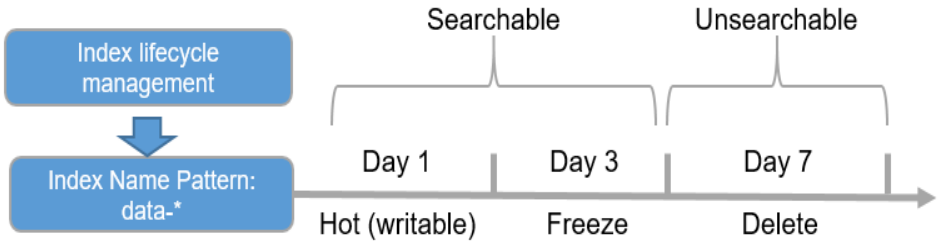
Solution

This topic describes how to configure an ISM policy to decouple storage and compute.

- Hot phase: Data is stored on high-performance local disks to support high-concurrency read and write operations.
- Warm phase: The index is frozen, and data is dumped to OBS. Local disks only retain necessary metadata, and query requests are directly sent to OBS. (Initial queries may experience higher latency, but subsequent queries can accelerate with the use of a cache.)
- Delete phase: When the data retention period expires, the index is automatically deleted to reclaim storage space.

In this example, an ISM policy is created for an Elasticsearch cluster. According to this policy, a newly created index is automatically frozen in three days, with data dumped to OBS; and deleted in seven days.

Figure 2-30 Storage-compute decoupling



Constraints

Only Elasticsearch 7.6.2 and 7.10.2 clusters support decoupled storage and compute.

Configuring an ISM Policy

1. Log in to Kibana and go to the command execution page.
   a. Log in to the **CSS management console**.
   b. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
   c. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

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

The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

2. Define an ISM policy to automate storage-compute decoupling for indexes.

For example, create a policy named *hot_warm_policy*. According to this policy, a newly created index is automatically frozen in three days, with data dumped to OBS; and deleted permanently in seven days.

```
PUT _opendistro/_ism/policies/hot_warm_policy
{
  "policy": {
    "description": "hot warm archive delete workflow",
    "error_notification": null,
    "default_state": "hot",
    "states": [
      {
        "name": "hot",
        "actions": [],
        "transitions": [
          {
            "state_name": "warm",
            "conditions": {
              "min_index_age": "3d"
            }
          }
        ]
      },
      {
        "name": "warm",
        "actions": [
          {
            "freeze_low_cost": {} // Freeze the index.
          }
        ],
        "transitions": [
          {
            "state_name": "delete",
            "conditions": {
              "min_index_age": "7d"
            }
          }
        ]
      },
      {
        "name": "delete",
        "actions": [
          {
            "delete": {}
          }
        ],
        "transitions": []
      }
    ]
  }
}
```

3. Associate the policy with an index template so that new indexes created using this template will automatically inherit this policy.

For example, create the index template **template_hot_warm**, which defines that newly created indexes whose name starts with **data** are automatically associated with the ISM policy **hot_warm_policy**.

```
PUT _template/template_hot_warm
{
  "index_patterns": "data*",
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
 "settings": {
   "number_of_replicas": 5, // Number of index replicas
   "number_of_shards": 1,   // Number of index shards
   "opendistro.index_state_management.policy_id": "hot_warm_policy"    // Index lifecycle policy name
 },
 "mappings": {
   "properties": {
     "name": {
       "type": "text"
     }
   }
 }
}
```

4. Write data to an index.

   For example, create an index named data-2022-06-06 and write data to it in batches.

   ```
   POST data-2022-06-06/_bulk
   {"index":{}}
   {"name":"name1"}
   {"index":{}}
   {"name":"name2"}
   {"index":{}}
   {"name":"name3"}
   {"index":{}}
   {"name":"name4"}
   {"index":{}}
   {"name":"name5"}
   {"index":{}}
   {"name":"name6"}
   ```

   The index will automatically use the index template *template_hot_warm* and inherit the lifecycle policy *hot_warm_policy* through the template.

5. Query data and check whether the ISM policy is automatically applied.

   a. Three days after index creation, query the list of frozen indexes and check whether indexes created three days ago have been frozen.

      ```
      GET _cat/freeze_indices?s=i&v
      ```

      If the returned result contains **data-2022-06-06**, the index has been dumped to OBS.

      ```
      health status index                uuid                   pri rep docs.count docs.deleted store.size
      pri.store.size
      green  open   data-2022-06-06 x8ab5NX6T3Ox_xoGUanogQ   1   1          6            0
      7.6kb          3.8kb
      ```

   b. Seven days after index creation, query the index list and check whether indexes created seven days ago have been deleted.

      ```
      GET /_cat/indices?expand_wildcards=open,closed
      ```

      If the returned result does not contain **data-2022-06-06**, the index has been deleted.

2.10 Search and Analytics

2.10.1 Using DSL for Search

When you need to interact with an Elasticsearch cluster to perform complex full-text search or data analysis, the domain-specific language (DSL) is your best choice. As a JSON-based query language native to Elasticsearch, DSL enables you to express sophisticated query logic with clarity and precision while giving you fine-grained control over how your queries are executed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

What Is DSL?

The Elasticsearch Query DSL is a JSON-based query language that defines the structure and semantics of search and data retrieval requests. It contains two contexts:

- Query context: In the query context, a query clause answers the question "How well does this document match this query clause?". Besides deciding whether or not the document matches, the query clause also calculates a relevance score in the **_score** metadata field. One example of such query clauses: match.
- In a filter context, a query clause answers the question "Does this document match this query clause?" The answer is a simple Yes or No. No scores are calculated, but frequently used filters will be cached automatically by Elasticsearch, to speed up performance. Examples of such queries: term and range.

DSL queries are executed in Dev Tools of Kibana. Both the request body and returned information are in JSON.

This topic lists some of the most commonly used DSL query clauses. For more, see **Query DSL**.

Basic Query: match_all

Use **match_all** to match all documents in the index. It is equivalent to **SELECT * FROM table** in SQL. Use it when you want to search all documents.

For example, run the following command to match all documents in the **test** index:

```
GET /test/_search
{
  "query": {
    "match_all": {}
  }
}
```

Compound Queries: Combining Multiple Query Clauses

Use a bool query with clauses such as must and filter to construct compound query conditions. This is similar to the **where** clause in SQL. Use this query when you need to apply multiple conditions to filter documents.

For example, run the following command to retrieve all documents whose status is **published** and whose publish_date is later than **2015-01-01** (filter condition), and whose title or content contains **Search** (search condition).

```
GET /_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "title": "Search"
          }
        },
        {
          "match": {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
          "content": "search"
        }
      }
    ],
    "filter": [
      {
        "term": {
          "status": "published"
        }
      },
      {
        "range": {
          "publish_date": {
            "gte": "2015-01-01"
          }
        }
      }
    ]
  }
 }
}
```

The differences between **must** and **filter** are as follows:

- **must**: Specifies conditions that documents must meet. These conditions affect the relevance score of documents, and documents with higher scores are ranked higher in the results.
- **filter**: Also specifies conditions that documents must meet, but these conditions do not affect the relevance score. **filter** is typically more efficient than **must**, making it better suited for filtering structured data.

For conditions that do not require relevance scoring (such as status, time range, and category), use **filter** to improve query performance.

Aggregations

The **aggs (such as the terms aggregation)** structure is used to perform aggregation queries. It is similar to the **Group by** clause in SQL. Use it when you want to group documents to calculate metrics.

For example, run the following command to count how many times different titles appear in the **test** index:

```
GET /test/_search
{
  "aggs": {
    "titles": {
      "terms": {
        "field": "title.keyword"
      }
    }
  }
}
```

The reason for using **title.keyword** is as follows:

- Text fields are used for full-text retrieval (they undergo tokenization). For example, "Hello World" will be split into "hello" and "world". If you attempt to aggregate on a text field directly, an error (Fielddata is disabled) will be returned. This is because it is an extremely memory-intensive operation.
- Keyword fields are used for exact match and aggregations. It keeps "Hello World" as a single value.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

By default, the cluster automatically creates multi-fields for strings. That means **title** is used for search, and **title.keyword** for aggregations.

2.10.2 Using SQL for Search

For developers and data analysts who are familiar with relational databases like MySQL, the native, JSON-based Elasticsearch query DSL may present a steep learning curve. In particular, constructing complex aggregation queries can be an error-prone process. To address this challenge, CSS Elasticsearch clusters integrate the Open Distro for Elasticsearch SQL plugin. This plugin enables users to query, aggregate, and analyze data in Elasticsearch using standard SQL syntax. It also supports interconnection with popular BI tools like Tableau through a JDBC driver, enabling a seamless data exploration experience.

How the Feature Works

Open Distro for Elasticsearch SQL is essentially a translator between SQL and Elasticsearch DSL. It works as follows:

1. Parsing: It receives and validates SQL queries submitted by users.
2. Translation: It converts SQL queries into native Elasticsearch DSL queries.
3. Execution: It sends DSL queries to the cluster for execution.
4. Return: It returns the results in JSON or CSV.

For more information, see **SQL Functions**.

Constraints

Only Elasticsearch 6.5.4 or later supports Open Distro for Elasticsearch SQL.

SQL Query Guide

- **Method 1: Running SQL Queries in Kibana (Recommended)**

  Use this method for development, debugging, and ad-hoc queries.

  - Perform a basic query.

    For example, run the following command to search the my-index index for documents where the age field exceeds 20, returning up to 10 results:

    ```
    POST _opendistro/_sql
    {
      "query": "SELECT * FROM my-index WHERE age > 20 LIMIT 10"
    }
    ```

  - Export data in CSV. To export data to an Excel document for analysis, specify **format=csv**.

    For example, run the following command to retrieve 10 documents from the my-index index based on the name and age fields and return the results in CSV format:

    ```
    POST _opendistro/_sql?format=csv
    {
      "query": "SELECT name, age FROM my-index LIMIT 10"
    }
    ```

  - Convert SQL to DSL. To learn how to write DSL queries or analyze the performance of SQL queries, use the **_explain** API to check the translated DSL queries.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

For example, run the following command to convert a SQL query into the equivalent DSL query:

```
POST _opendistro/_sql/_explain
{
  "query": "SELECT * FROM my-index WHERE age > 20"
}
```

- **Method 2: Running SQL Queries via cURL Commands on a Server**

  Use this method when you need to incorporate Elasticsearch queries into automated scripts or scheduled tasks.

  For example, run the following command to retrieve 10 records from the **my-index** index. The following is an example of the code used in a security-mode cluster that uses HTTP:

```
curl -XPOST http://<cluster IP address>:9200/_opendistro/_sql \
  -u <username>:<password> \
  -H 'Content-Type: application/json' \
  -d '{"query": "SELECT * FROM my-index LIMIT 10"}'
```

Supported SQL Syntax

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

- Statements

Table 2-94 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

CAUTION

Large UNION and MINUS queries can cause high resource consumption, potentially leading to degrading cluster performance or even service interruptions. To mitigate this risk, you are advised to optimize the query structure or split large queries into smaller batches.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- Conditions

  Table 2-95 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 ('Bob', 'David')
Not	SELECT * FROM my-index WHERE name NOT IN ('Bob')
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 2-96 Aggregations

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

- Include and exclude fields

  Table 2-97 Include and exclude fields

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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Pattern	Example
exclude()	SELECT exclude('*name') FROM my-index

- Functions

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

Table 2-98 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
/	SELECT number / 100 FROM my-index
%	SELECT number % 100 FROM my-index
date_format	SELECT date_format(date, 'Y') FROM my-index

- Joins

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

CAUTION

Supported join types: INNER JOIN, LEFT JOIN, and CROSS JOIN.

Hard limits:

- Only two tables (indexes) can be joined.
- An alias must be specified for each table, for example, **FROM people p JOIN index b**.
- The ON condition supports only AND combinations, not OR or nested conditions.
- GROUP BY or ORDER BY cannot be used in a JOIN query.
- LIMIT and OFFSET cannot be used together. A negative example: **LIMIT 25 OFFSET 25**
- Fields from multiple indexes cannot be combined in the WHERE clause.

  A positive example:

  ```
  WHERE (a.type1 > 3 OR a.type1 < 0) AND (b.type2 > 4 OR b.type2 < -1)
  ```

  A negative example:

  ```
  WHERE (a.type1 > 3 OR b.type2 < 0) AND (a.type1 > 4 OR b.type2 < -1)
  ```

Table 2-99 Joins

Join	Example
Inner join	SELECT s.firstname, s.lastname, s.gender, sc.name FROM student s JOIN school sc ON sc.name = s.school_name WHERE s.age > 20
Left outer join	SELECT s.firstname, s.lastname, s.gender, sc.name FROM student s LEFT JOIN school sc ON sc.name = s.school_name
Cross join	SELECT s.firstname, s.lastname, s.gender, sc.name FROM student s CROSS JOIN school sc

- Show

  Show commands display indexes and mappings that match an index pattern. You can use ***** or **%** for wildcards.

Table 2-100 Show

Show	Example
Show tables like	SHOW TABLES LIKE logs-*

BI Integration via JDBC Driver

If you are using Tableau, Power BI, or a self-managed Java application, you can connect to a CSS cluster using the JDBC driver.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Download the JDBC driver from the **GitHub repository**.

2.10.3 Configuring Custom Dictionaries

In daily search scenarios, Elasticsearch's built-in dictionaries often fail to recognize domain-specific terms. Such terms may be split into separate words (or separate characters in Chinese), resulting in low search recall and inaccurate results. Additionally, meaningless high-frequency words (such as stop words), if not filtered out, can introduce noise and reduce search efficiency. Furthermore, when users search for "cell phone", they may expect documents containing synonyms such as "mobile phone" to also be returned, but built-in dictionaries do not support synonym expansion. To address these issues, CSS provides the IK analysis plugin (analysis-ik) and dynamic synonym plugin (analysis-dynamic-synonym). By configuring custom dictionaries, you can enable Elasticsearch to accurately recognize domain-specific terms, filter irrelevant words, and expand synonyms, delivering a smarter search experience tailored to business use cases.

How the Feature Works

Dictionaries are core resources that guide analyzers in recognizing special terms. CSS supports the following three types of dictionaries:

- Main word dictionary: Defines special terms that should be recognized as independent terms rather than split into separate words by the analyzer.
- Stop word dictionary: Defines meaningless words that should be filtered out in order to reduce noise.
- Synonym dictionary: Defines groups of semantically equivalent terms to enable synonym-based search expansion.

Dictionaries take effect only when used together with analyzers. CSS provides a built-in standard analyzer, IK analyzers, and a synonym filter. IK analyzers are implemented through the IK analysis plugin, and the synonym filter is implemented through the dynamic synonym plugin. **Table 2-101** describes the mapping between dictionaries and analyzers.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-101 Mapping between dictionaries and analyzers

Analyzer	Description	Dictionary Dependency
Default analyzer: Standard Analyzer	A standard analyzer based on Unicode text segmentation. It automatically converts text to lowercase and splits Chinese text into individual characters. • For example, **Hello World** is split into **hello,world**. • Advantage: Simple configuration; no dictionary maintenance required. • Disadvantage: Poor performance in Chinese segmentation. • Applicable scenario: Indexing Western languages or scenarios with low requirements for Chinese text analysis.	No dictionary required
IK Analyzer ik_smart	A coarse-grained analyzer that generates only the most reasonable segmentation results without enumerating all possible combinations. • Example: Text is segmented into the most likely word combinations rather than all possible combinations. • Advantage: Provides good Chinese text segmentation, reduces noise, and improves search precision. • Disadvantage: May miss some less commonly used terms or word combinations. • Applicable scenario: Query analysis during the search phase.	Requires main word and stop word dictionaries (built-in dictionaries are provided; custom dictionaries are also supported)

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Analyzer	Description	Dictionary Dependency
IK Analyzer ik_max_word	A fine-grained analyzer that enumerates all possible word combinations and generates the finest-grained segmentation results. • Example: Text is segmented into all possible word combinations. • Advantage: Provides good Chinese segmentation, covers more term combinations, and improves search recall. • Disadvantage: Increases index size and slightly reduces write performance. • Applicable scenario: Index building for full-text search during data ingestion.	Requires main word and stop word dictionaries (built-in dictionaries are provided; custom dictionaries are also supported)
Synonym filter dynamic_synonym	A synonym expansion filter that dynamically loads synonym dictionaries and performs semantic expansion or correction on segmentation results. • Example: Searching for **cell phone** also returns documents containing **mobile phone** and **mobile device**. • Advantage: Enables synonym expansion search and improves search recall. • Disadvantage: Must be used together with the standard analyzer or IK analyzer. • Applicable scenarios: Semantic expansion search and term correction.	Requires a synonym dictionary (supports custom dictionaries only)

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

CAUTION

The IK Analyzer is designed primarily for Chinese text. When used to analyze English text, only the following special characters are treated as regular characters and preserved: # & + - . @ _. Other special characters are treated as segmentation points where text can be split into different parts. Example:

- **access address: example@domain.com** is split into **access,address,example@domain.com**.
- **set the parameter value to user|domain** is split into **set,parameter,value,user,domain**. (**the** and **to** are in the built-in Static Stop Word Dictionary and are filtered out during word segmentation.)

How Custom Dictionaries Work

The workflow of custom dictionaries is as follows:

1. **Dictionary loading**: After uploading dictionary files to an OBS bucket, configure the dictionaries for the cluster. The system then loads the dictionaries to all nodes in the cluster.
2. **Main word dictionary matching**: The analyzer first matches entries in the main word dictionary and keeps matched entries as complete terms during segmentation, preventing special terms from being split.
3. **Stop word filtering**: Entries matching the stop word dictionary in the segmentation results are discarded and are not indexed.
4. **Synonym expansion**: The segmentation results are processed by the synonym filter. Entries matching the synonym dictionary are expanded into additional synonym terms to enable semantic expansion search.
5. **Scope of update**: After a dictionary is updated, newly written data is automatically analyzed using the new dictionary. Historical index data is not rebuilt automatically. To re-analyze historical data using the updated dictionary, manually run the **_update_by_query** command.

Dictionary File Specifications

Prepare dictionary files according to the following specifications. Files that do not meet these requirements will fail to be loaded.

Table 2-102 Dictionary file specifications

Item	Requirement
File format	**.txt**
File encoding	UTF-8 without BOM

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Item	Requirement
Content format	• Main word and stop word dictionaries: One term per line. Example: cloud computing search engine • Synonym dictionary: One synonym group per line. Use commas (,) to separate bidirectional synonyms, and => to separate unidirectional synonyms. Example: happy,glad,pleased cell phone=>mobile phone With bidirectional synonyms, searching for **happy** returns documents containing **happy**, **glad**, and **pleased**. With unidirectional synonyms, searching for **cell phone** only returns documents containing **mobile phone**.
English terms	Must be in lowercase.
File size	Maximum 100 MB per file.

Prerequisites

- The target cluster is in the **Available** state and has no ongoing tasks.
- The account you are using has the following permissions (choose **Permissions** > **Policies/Roles** on the IAM console to check your permissions):
  - Permission to configure custom dictionaries:
    ```
    "css:IKThesaurus:*"
    ```
  - To update dictionaries, the account must also have read access to the OBS bucket and objects:
    ```
    obs:bucket:getBucketLocation
    obs:bucket:getBucketStoragePolicy
    obs:bucket:listAllMyBuckets
    obs:object:getObject
    ```
  - If the OBS bucket uses SSE-KMS encryption, the following KMS permissions are also required:
    ```
    "kms:cmk:create",
    "kms:dek:create",
    "kms:cmk:get",
    "kms:dek:decrypt",
    "kms:cmk:list"
    ```
- To update dictionaries, prepare dictionary files that comply with the required format and upload them to an OBS bucket. For details, see **Uploading an Object**.

  The OBS bucket must be in the same region as the CSS cluster, and its storage class must be **Standard**.

Accessing the Custom Dictionary Configuration Page

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Cluster Settings > Custom Word Dictionaries**.

Configuring Custom Dictionaries

If built-in dictionaries do not meet your business requirements for text analysis, you can configure custom dictionaries.

1. On the Custom Word Dictionaries tab, configure custom dictionaries as required.

Table 2-103 Configuring custom dictionaries

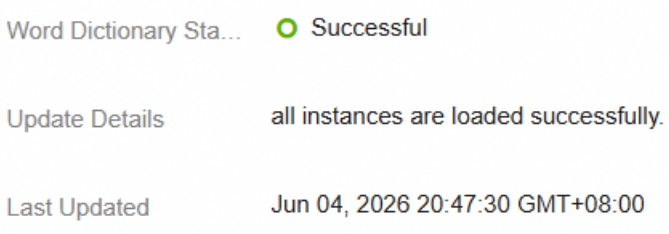
Parameter	Description
OBS Bucket	Select the OBS bucket that stores the dictionary file. This parameter is required only when you update dictionaries. If no OBS buckets meet your requirements, click **Create Bucket** to go to the OBS console and create one. For details, see **Creating a Bucket**.
Main Word Dictionary	A custom collection of main words. It is empty initially. • Default value: **No Update**, meaning not to configure this dictionary. • To create a custom dictionary or update an existing one, choose **Update > Select**, select a dictionary file that complies with the required format from the OBS bucket, and click **OK**. • To delete a dictionary, click **Do Not Use**.
Stop Word Dictionary	A custom collection of stop words. It is empty initially. • Default value: **No Update**, meaning not to configure this dictionary. • To create a custom dictionary or update an existing one, choose **Update > Select**, select a dictionary file that complies with the required format from the OBS bucket, and click **OK**. • To delete a dictionary, click **Do Not Use**.
Synonym Dictionary	A custom collection of synonyms. It is empty initially. • Default value: **No Update**, meaning not to configure this dictionary. • To create a custom dictionary or update an existing one, choose **Update > Select**, select a dictionary file that complies with the required format from the OBS bucket, and click **OK**. • To delete a dictionary, click **Do Not Use**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. Click **Save**. In the displayed dialog box, click **OK** to start updating dictionaries.

   The dictionary configuration information is displayed below. Wait for approximately 1 minute. If the dictionary status changes from **Updating** to **Successful**, the dictionary update is complete.

   Figure 2-31 Dictionary status

3. (Optional) After a dictionary is updated, newly written data uses the updated dictionary for analysis. Historical index data is not rebuilt automatically. To re-analyze and re-index historical data, manually run the following command:

   ```
   POST /{my_index}/_update_by_query
   ```

Updating Built-in Dictionaries

You can update built-in dictionaries if you need to modify or disable them.

1. On the **Custom Word Dictionaries** page, toggle on the switch next to **Modify Preset Word Dictionaries**.

   CAUTION

   If **Modify Preset Word Dictionaries** is not displayed, the current cluster version does not support built-in dictionary updates. In this case, you are advised to upgrade the cluster version or create a new cluster and then migrate data to it.

2. Update built-in dictionaries as required.

   Table 2-104 Updating built-in dictionaries

Parameter	Description
OBS Bucket	Select the OBS bucket that stores the dictionary file. This parameter is required only when you update dictionaries. If no OBS buckets meet your requirements, click **Create Bucket** to go to the OBS console and create one. For details, see **Creating a Bucket**.

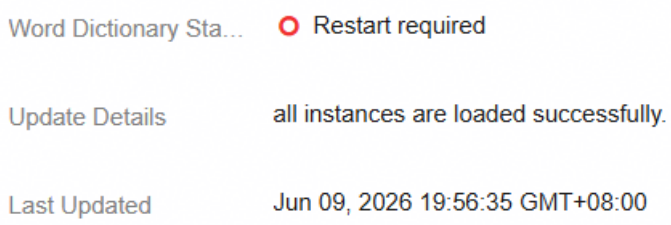
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Static Main Word Dictionary	A built-in collection of commonly used main words. • Default value: **No Update**, meaning not to configure this dictionary. • To update this dictionary, choose **Update** > **Select**, select a dictionary file that complies with the required format from the OBS bucket, and click **OK**. • To delete this dictionary, click **Do Not Use**.
Static Stop Word Dictionary	A built-in collection of commonly used stop words. • Default value: **No Update**, meaning not to configure this dictionary. • To update this dictionary, choose **Update** > **Select**, select a dictionary file that complies with the required format from the OBS bucket, and click **OK**. • To delete this dictionary, click **Do Not Use**.
Extra Main Word Dictionary	A built-in collection of uncommon main words. • Default value: **No Update**, meaning not to configure this dictionary. • To update this dictionary, choose **Update** > **Select**, select a dictionary file that complies with the required format from the OBS bucket, and click **OK**. • To delete this dictionary, click **Do Not Use**.
Extra Stop Word Dictionary	A built-in collection of uncommon stop words. • Default value: **No Update**, meaning not to configure this dictionary. • To update this dictionary, choose **Update** > **Select**, select a dictionary file that complies with the required format from the OBS bucket, and click **OK**. • To delete this dictionary, click **Do Not Use**.

3. Click **Save**. In the displayed dialog box, click **OK** to start updating dictionaries.

   The dictionary configuration information is displayed below. Wait for approximately 1 minute. If the dictionary status changes from **Updating** to **Restart required**, the dictionary update is complete.

   Figure 2-32 Dictionary status



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

4. Click **Restart** in the upper-right corner of the page to restart the cluster and apply changes made to the built-in dictionaries. For details, see **Restarting a Cluster**.
5. (Optional) After a dictionary is updated, newly written data uses the updated dictionary for analysis. Historical index data is not rebuilt automatically. To re-analyze and re-index historical data, manually run the following command:

```
POST /{my_index}/_update_by_query
```

Verifying Main Word and Stop Word Dictionaries

After configuring the main word dictionary and stop word dictionary, you can use the /_analyze API to test the text segmentation results.

Scenario: The custom main word dictionary and stop word dictionary have been configured with the necessary main words and stop words.

1. On the **Custom Word Dictionaries** page, click **Kibana** in the upper-right corner to log in to Kibana.
2. In the left navigation pane on Kibana, choose **Dev Tools**.
3. Run the following commands to check the segmentation results of **ik_smart** and **ik_max_word**:
   - **ik_smart (coarse-grained) segmentation**

     Example request:

     ```
     POST /_analyze
     {
       "analyzer":"ik_smart",
       "text":"Sample Chinese text"
     }
     ```

     Expected result: The main words in the provided text are recognized as independent terms, and stop words are filtered out.

     ```
     {
       "tokens": [
         {
           "token": "main_word_1",
           "start_offset": 0,
           "end_offset": 4,
           "type": "CN_WORD",
           "position": 0
         },
         {
           "token": "main_word_2",
           "start_offset": 5,
           "end_offset": 8,
           "type": "CN_WORD",
           "position": 1
         }
       ]
     }
     ```

   - **ik_max_word (fine-grained) segmentation**

     Example request:

     ```
     POST /_analyze
     {
       "analyzer":"ik_max_word",
       "text":"Sample Chinese text"
     }
     ```

     Expected result: The main words in the provided text are segmented into multiple possible terms, and stop words are filtered out.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
{
  "tokens": [
    { "token": "word_1", "start_offset": 0, "end_offset": 4, "type": "CN_WORD", "position": 0 },
    { "token": "word_2", "start_offset": 0, "end_offset": 2, "type": "CN_WORD", "position": 1 },
    { "token": "word_3", "start_offset": 0, "end_offset": 1, "type": "CN_WORD", "position": 2 },
    { "token": "word_4", "start_offset": 1, "end_offset": 3, "type": "CN_WORD", "position": 3 },
    { "token": "word_5", "start_offset": 2, "end_offset": 4, "type": "CN_WORD", "position": 4 },
    { "token": "word_6", "start_offset": 3, "end_offset": 4, "type": "CN_WORD", "position": 5 },
    { "token": "word_7", "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 }
  ]
}
```

Verifying Synonym Expansion

After configuring the synonym dictionary, you can test the dictionary through synonym search.

Scenario: The synonym dictionary contains synonyms "happy, glad".

1. On the **Custom Word Dictionaries** page, click **Kibana** in the upper-right corner to log in to Kibana.
2. In the left navigation pane on Kibana, choose **Dev Tools**.
3. Run the following command to create the **test** index and configure the synonym filter.

```
PUT test
{
    "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"
            }
        }
    }
}
```

4. Run the following command to verify synonym filter configuration.

```
GET /test/_analyze
{
  "analyzer": "ik_synonym",
  "text": ["I'm very happy"]
}
```

Expected result: Although the text uses "happy", the segmentation results also contain "glad" based on the synonym mapping.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
{
  "tokens": [
    { "token": "I'm", "start_offset": 0, "end_offset": 2, "type": "CN_WORD", "position": 0 },
    { "token": "happy", "start_offset": 2, "end_offset": 4, "type": "CN_WORD", "position": 1 },
    { "token": "glad", "start_offset": 2, "end_offset": 4, "type": "SYNONYM", "position": 1 }
  ]
}
```

Preset Dictionaries

To help you quickly get started with the IK analyzer, CSS provides four built-in dictionaries, as listed in **Table 2-105**. **If these preset main word and stop word dictionaries already meet your requirements, there is no need to configure custom dictionaries.**

Table 2-105 Preset dictionaries

Preset Dictionary	Description	Download URL
Static main word dictionary	A collection of common main words (default main words of the IK analyzer)	**https://github.com/infinilabs/analysis-ik/blob/master/config/main.dic**
Static stop word dictionary	A collection of common stop words (default stop words of the IK analyzer)	**https://github.com/infinilabs/analysis-ik/blob/master/config/stopword.dic**
Extra main word dictionary	A collection of uncommon main words (extended main words of the IK analyzer)	**https://github.com/infinilabs/analysis-ik/blob/master/config/extra_main.dic**
Extra stop word dictionary	A collection of uncommon stop words (extended stop words of the IK analyzer)	**https://github.com/infinilabs/analysis-ik/blob/master/config/extra_stopword.dic**

Related Documents

- **Improving Keyword Search Accuracy with the IK Analyzer and Custom Dictionaries**: Configure custom main word and stop word dictionaries with the IK analyzer to preserve domain-specific terms and filter irrelevant words, enabling more precise keyword search.
- **Improving Search Recall with the Synonym Filter and a Custom Synonym Dictionary**: Configure a custom synonym dictionary and use the

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

dynamic_synonym synonym filter to automatically expand synonyms during query and indexing, enabling synonym-based search across equivalent terms.

2.10.4 Configuring Simplified-Traditional Chinese Conversion Search

In cross-region business scenarios, users may need to search Traditional Chinese content using Simplified Chinese keywords, or search Simplified Chinese content using Traditional Chinese keywords. For example, a user may enter a Simplified Chinese keyword and expect documents containing its Traditional Chinese equivalent to be returned. However, due to differences in character forms between Simplified and Traditional Chinese, direct searches may fail to accurately retrieve the expected documents, resulting in a poor user experience. To address this issue, CSS provides the Simplified-Traditional Chinese conversion plugin (analysis-stconvert) for Elasticsearch clusters. By automatically converting between Simplified and Traditional Chinese during text analysis, the plugin eliminates the need to modify business data or implement custom conversion logic. You only need to configure the corresponding analyzer when creating an index to enable cross-form Chinese search, such as searching Traditional Chinese documents with Simplified Chinese keywords or vice versa.

How the Feature Works

The Simplified-Traditional Chinese conversion plugin (analysis-stconvert) supports bidirectional conversion between Simplified and Traditional Chinese. The plugin is pre-installed in CSS and ready for use out of the box. Its core capability is to convert characters during text analysis, unifying Simplified and Traditional Chinese at the index level to enable cross-form search and matching.

Table 2-106 Supported conversion types

Conversion Type	Description	Example
s2t	Converts Simplified Chinese to Traditional Chinese.	**A Simplified Chinese term** > **Its Traditional Chinese equivalent**
t2s	Converts Traditional Chinese to Simplified Chinese. This is the commonly used type. The IK analyzer dictionaries are primarily based on Simplified Chinese, so IK analyzers generally provide better analysis results for Simplified Chinese text.	**A Traditional Chinese term** > **Its Simplified Chinese equivalent**

The Simplified-Traditional Chinese conversion plugin can be flexibly integrated into different stages of text analysis. The following four usage modes are supported:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-107 Usage modes of the Simplified-Traditional Chinese conversion plugin

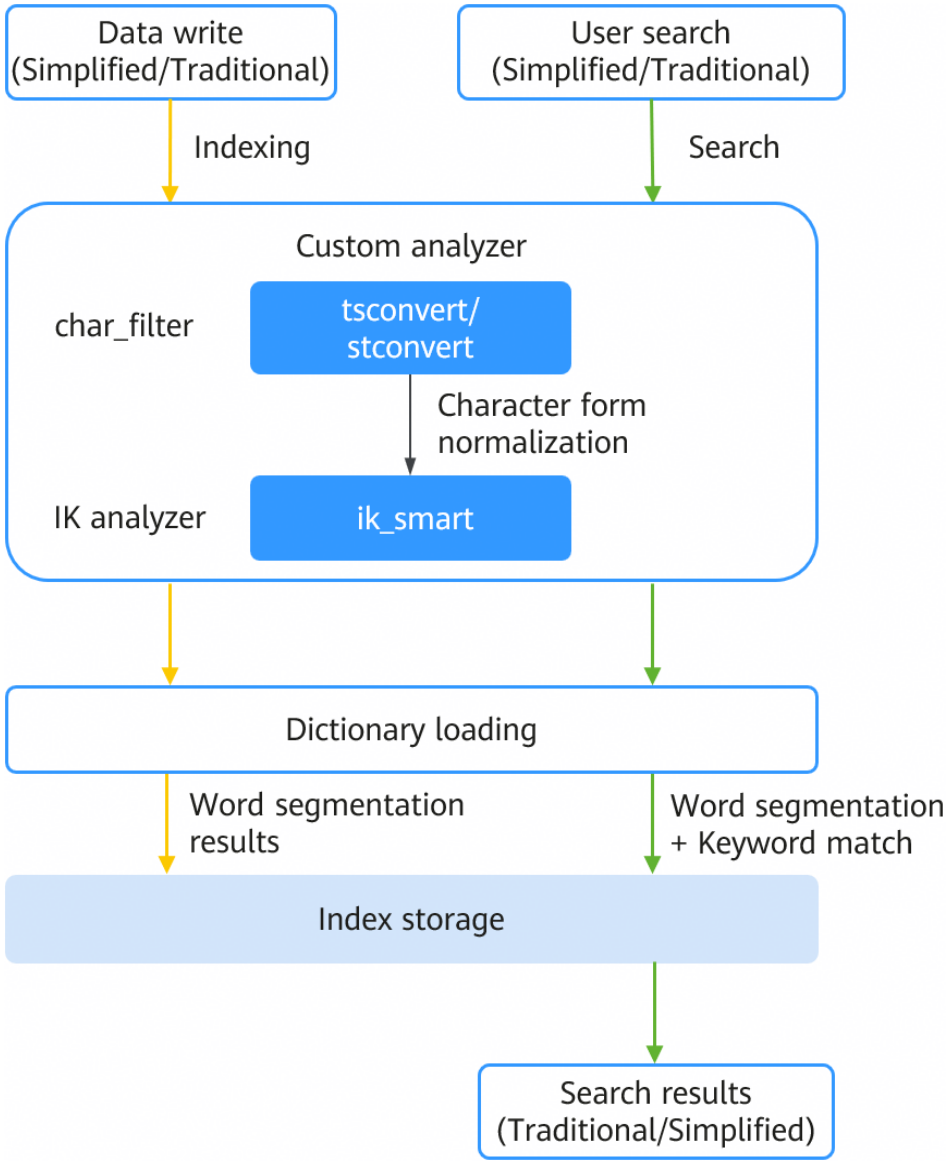
Usage	Description	Applicable Scenario
char_filter	Converts the original string before tokenization. • tsconvert: Traditional Chinese to Simplified Chinese (t2s) • stconvert: Simplified Chinese to Traditional Chinese (s2t)	Recommended mode. Used when character forms need to be unified before tokenization. Requires a tokenizer.
tokenizer	Combines conversion and tokenization into a single step.	Lightweight scenarios where no additional tokenizer is required.
token_filter	Converts tokens after tokenization.	Used when text must be tokenized first before being normalized to the same character form. Requires a tokenizer.
analyzer	Uses the plugin directly as a complete analyzer.	Only Simplified-Traditional Chinese conversion is required. There is no need for Chinese text analysis capabilities such as IK.

How Simplified-Traditional Chinese Conversion Works

This topic uses the char_filter (tsconvert) mode to integrate Simplified-Traditional Chinese conversion before the IK analyzer, enabling character form normalization before word segmentation.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 2-33 Solution architecture



The key workflow is as follows:

- Indexing phase: Original text is first processed by the char_filter (tsconvert) to convert Traditional Chinese characters into Simplified Chinese characters. The converted text is then analyzed by the IK analyzer (ik_smart), and the results are stored in the inverted index.
- Query phase: Search keywords undergo the same character normalization process before segmentation. The resulting terms are matched against the inverted index, enabling cross-form Simplified-Traditional Chinese search.
- Custom analyzer: A custom analyzer is created by combining the ik_smart analyzer with the tsconvert character filter. The same analyzer is used for both indexing and query processing.

Prerequisites

- The target cluster is in the **Available** state and has no ongoing tasks.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- If the custom analyzer depends on custom dictionaries, configure the custom dictionaries in advance. For details, see **Configuring Custom Dictionaries**.

NOTE

The sample code in this topic applies to Elasticsearch 7.x clusters. If you are using an Elasticsearch version earlier than 7.x, see **Example code (Elasticsearch < 7.x)**.

Logging In to Kibana

Log in to Kibana and go to the command execution page. Elasticsearch clusters support multiple access methods. This topic uses Kibana as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Creating an Index

Create an index and define a custom analyzer with Simplified-Traditional Chinese conversion logic. The analyzer normalizes character forms before word segmentation during both indexing and query processing.

Run the following command to create the **test** index and configure the custom analyzer **ts_ik** (which combines **ik_smart** and **tsconvert**).

```
PUT /test
{
  "settings": {
    "number_of_shards": 1,
    "number_of_replicas": 0,
    "analysis": {
      "analyzer": {
        "ts_ik": {
          "tokenizer": "ik_smart",
          "char_filter": [
            "tsconvert"
          ]
        }
      },
      "char_filter": {
        "tsconvert": {
          "type": "stconvert",
          "convert_type": "t2s"
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "desc": {
        "type": "text",
        "analyzer": "ts_ik"
      }
    }
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  }
}
```

Table 2-108 Key parameters

Parameter	Description
ts_ik	A custom analyzer that first uses the **tsconvert** character filter to convert Traditional Chinese characters into Simplified Chinese characters, and then uses **ik_smart** for word segmentation.
char_filter.tsconvert	A custom character filter based on the **stconvert** conversion plugin. The **convert_type: t2s** parameter specifies Traditional-to-Simplified Chinese conversion.
convert_type: t2s	Converts Traditional Chinese into Simplified Chinese characters before word segmentation. Because the IK analyzer provides better segmentation results for Simplified Chinese, you are advised to convert Traditional Chinese to Simplified Chinese.
analyzer: ts_ik	Associates the **desc** field with the custom analyzer. The analyzer is used for both indexing and query processing.

Expected result:

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

Verifying Character-Form Conversion and Text Analysis

After creating the index, verify the custom analyzer to ensure that Simplified-Traditional Chinese conversion and word segmentation are configured correctly before importing data.

Run the following command to check how the ts_ik custom analyzer converts and segments Traditional Chinese text.

```
POST /test/_analyze
{
  "analyzer": "ts_ik",
  "text": "Text in traditional Chinese"
}
```

Expected result:

```
{
  "tokens" : [
    { "token": "Simplified Chinese term 1", "start_offset": 0, "end_offset": 2, "type": "CN_WORD", "position": 0 },
    { "token" : "Simplified Chinese term 1", "start_offset" : 2, "end_offset" : 5, "type" : "CN_WORD", "position" : 1 }
  ]
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Result interpretation: The Traditional Chinese text is converted into Simplified Chinese text first. The ik_smart analyzer then segments the converted text into terms. Both conversion and tokenization are working as expected.

Importing Data

Import both Traditional and Simplified Chinese content into the index to verify cross-form search with mixed Simplified and Traditional Chinese data.

Run the following command to write test data to the **test** index:

```
POST /test/_bulk
{"index":{"_id":"1"}}
{"desc":"Traditional Chinese sample text"}
{"index":{"_id":"2"}}
{"desc":"Simplified Chinese sample text"}
{"index":{"_id":"3"}}
{"desc":"Unrelated text"}
```

Expected result:

```
{
  "took" : 6,
  "errors" : false,
  "items" : [
    { "index" : { "_index" : "test", "_type" : "_doc", "_id" : "1", "_version" : 1, "result" : "created", "_shards" :
{ "total" : 1, "successful" : 1, "failed" : 0 }, "_seq_no" : 0, "_primary_term" : 1, "status" : 201 } },
    { "index" : { "_index" : "test", "_type" : "_doc", "_id" : "2", "_version" : 1, "result" : "created", "_shards" :
{ "total" : 1, "successful" : 1, "failed" : 0 }, "_seq_no" : 1, "_primary_term" : 1, "status" : 201 } },
    { "index" : { "_index" : "test", "_type" : "_doc", "_id" : "3", "_version" : 1, "result" : "created", "_shards" :
{ "total" : 1, "successful" : 1, "failed" : 0 }, "_seq_no" : 2, "_primary_term" : 1, "status" : 201 } }
  ]
}
```

Verifying Simplified-Traditional Chinese Conversion Search

Search using both Simplified Chinese and Traditional Chinese keywords to verify cross-form search.

Run the following command to search using a Simplified Chinese keyword.

```
GET /test/_search
{
  "query": {
    "match": {
      "desc": "Simplified Chinese keyword"
    }
  }
}
```

Run the following command to search using the corresponding Traditional Chinese keyword.

```
GET /test/_search
{
  "query": {
    "match": {
      "desc": "Traditional Chinese keyword"
    }
  }
}
```

The results of the two searches are expected to be the same:

```
{
...
  "hits" : {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  "total" : {
    "value" : 2,
    "relation" : "eq"
  },
  "max_score" : 0.43445712,
  "hits" : [
    {
      "_index" : "test",
      "_type" : "_doc",
      "_id" : "1",
      "_score" : 0.43445712,
      "_source" : {
        "desc": "Traditional Chinese document"
      }
    },
    {
      "_index" : "test",
      "_type" : "_doc",
      "_id" : "2",
      "_score" : 0.43445712,
      "_source" : {
        "desc" : "Simplified Chinese document"
      }
    }
  ]
 }
}
```

Result interpretation: Whether you search using the Simplified Chinese keyword or its Traditional Chinese equivalent, both Traditional Chinese and Simplified Chinese documents containing the keyword are returned. This indicates that bidirectional Simplified-Traditional Chinese search is working. Documents that do not contain the keyword are not returned.

FAQ

- **Question 1: How do I configure Simplified Chinese search to match Traditional Chinese documents?**

  When creating an index, replace the tsconvert (t2s) character filter with the stconvert (s2t) character filter. In this configuration, Simplified Chinese content is converted to Traditional Chinese during indexing, and user queries are also converted to Traditional Chinese during query processing. Configure the analyzer as follows:

  ```
  PUT /test1
  {
    "settings": {
      "number_of_shards": 1,
      "number_of_replicas": 0,
      "analysis": {
        "analyzer": {
          "ts_ik": {
            "tokenizer": "ik_smart",
            "char_filter": [
              "stconvert"
            ]
          }
        },
        "char_filter": {
          "stconvert": {
            "type": "stconvert",
            "convert_type": "s2t"
          }
        }
      }
    },
  ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

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

- **Question 2: Do existing indexes need to be rebuilt to support Simplified-Traditional Chinese conversion?**

  Analyzer configurations can only be specified when an index is created and cannot be modified for existing indexes. Whether an index needs to be rebuilt depends on the existing data:

  – If all existing data is in Simplified Chinese, no index rebuild is required after adding a Traditional-to-Simplified Chinese analyzer. Existing data is already in Simplified Chinese, and newly ingested Traditional Chinese data will be automatically converted.

  – If existing data contains Traditional Chinese content, the index must be rebuilt after adding a Traditional-to-Simplified Chinese analyzer. Otherwise, historical Traditional Chinese data cannot be retrieved by Simplified Chinese keywords.

  The following two methods can be used to rebuild an index:

  – Method 1: Update in place (where there is only a small amount of data)

    Run the following command:

    ```
    POST /{my_index}/_update_by_query
    ```

    This command re-analyzes existing documents but does not modify the analyzer configuration of the index. This method works only if the index was originally created with the Simplified-Traditional Chinese conversion analyzer.

  – Method 2: Switch indexes seamlessly using an alias

    i. Create a new index configured with the Simplified-Traditional Chinese conversion analyzer.

    ii. Migrate data from the old index to the new index using **_reindex**.

    ```
    POST /_reindex
    {
     "source": { "index": "old_index" },
     "dest": { "index": "new_index" }
    }
    ```

    iii. Point the alias to the new index.

    ```
    POST /_aliases
    {
     "actions": [
      {
       "remove": {
        "index": "old_index",
        "alias": "my_alias"
       }
      },
      {
       "add": {
        "index": "new_index",
        "alias": "my_alias"
       }
      }
     ]
    }
    ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

iv. (Optional) Delete the old index to release storage space.

```
DELETE /old_index
```

Example code (Elasticsearch < 7.x)

When the Elasticsearch version is earlier than 7.x, specify a custom type name. See the following command.

1. Run the following command to create the **test** index and specify the custom analyzer:

```
PUT /test
{
  "settings": {
    "number_of_shards": 1,
    "number_of_replicas": 0,
    "analysis": {
      "analyzer": {
        "ts_ik": {
          "tokenizer": "ik_smart",
          "char_filter": [
            "tsconvert"
          ]
        }
      },
      "char_filter": {
        "tsconvert": {
          "type": "stconvert",
          "convert_type": "t2s"
        }
      }
    }
  },
  "mappings": {
    "type1": {
      "properties": {
        "desc": {
          "type": "text",
          "analyzer": "ts_ik"
        }
      }
    }
  }
}
```

2. Run the following command to write test data to the **test** index:

```
POST /_bulk
{ "index" : { "_index" : "test", "_type" : "type1", "_id" : "1" } }
{ "desc" : "Traditional Chinese sample text" }
{ "index" : { "_index" : "test", "_type" : "type1", "_id" : "2" } }
{ "desc" : "Simplified Chinese sample text" }
{ "index" : { "_index" : "test", "_type" : "type1", "_id" : "3" } }
{ "desc" : "Unrelated text" }
```

3. Run the following command to verify the configuration by executing a keyword-based query:

```
GET /test/type1/_search
{
  "query": {
    "match": {
      "desc": "Traditional Chinese keyword"
    }
  }
}
```

2.11 Core Service Enhancements

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2.11.1 CSS Enhancements Over Open-Source Elasticsearch

In enterprise-grade search and analytics scenarios, open-source Elasticsearch, despite its powerful features, often struggles to meet demands such as searching through tens of millions of high-dimensional vectors, managing PB-scale data storage, handling out-of-memory (OOM) exceptions caused by large queries, and maintaining stability during traffic peaks. To overcome these challenges, CSS offers a range of enhancements over the open-source Elasticsearch, such as decoupled storage and compute, large query isolation, and enhanced ingestion performance. These enhancements go beyond performance. They also improve production-environment stability. With simple configurations, your cluster can support high concurrency and cost-effective storage of massive datasets, allowing you to focus on your core applications.

Table 2-109 CSS enhancements over open-source Elasticsearch

Category	Feature	Description	Supported Version	Details
Cost optimization	Storage-compute decoupling	With decoupled storage and compute, hot data that is frequently accessed is stored in high-performance storage media, while cold data that is infrequently accessed is migrated to low-cost storage media—Object Storage Service (OBS). This ensures real-time query performance for hot data while reducing long-term storage costs.	7.6.2, 7.10.2	**Configuring Decoupled Storage and Compute**
	Switching between hot and cold storage tiers	Data is allocated to nodes of different performance standards based on data temperature (that is, how often data is accessed). The goal is to achieve optimal storage costs and query performance.	All versions that support cold data nodes	**Switching Between Hot and Cold Data**
Enhanced stability	Read/write splitting	This feature relies on collaboration between the leader and follower clusters. Writes are directed to the leader and queries to the follower, ensuring optimal write performance while supporting high-concurrency, scalable queries. There is no more resource contention, and peak loads are reduced.	7.6.2, 7.10.2	**Configuring Read/Write Splitting Between Leader and Follower**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Category	Feature	Description	Supported Version	Details
	Flow control	This feature protects clusters from overload through flow control policies, such as client request throttling, shard indexing backpressure, and traffic pattern analysis, ensuring proper resource allocation and cluster stability. **NOTE** Elasticsearch 7.6.2 and 7.10.2 clusters created after February 2023 support Flow Control 2.0 only, while clusters created before that support Flow Control 1.0 only.	7.6.2, 7.10.2	**Configuring Flow Control 2.0** **Configuring Flow Control 1.0**
	Query traffic isolation	You can configure specific nodes as low-priority nodes or isolated nodes and define an index whitelist for exceptions. For shard query requests to indexes that are not on this whitelist, the system preferentially or completely bypasses these nodes, ensuring that faulty nodes do not affect services.	7.10.2	**Configuring Query Traffic Isolation**
	Large query isolation	Large query isolation can be configured to manage queries that have high memory usage or take too long to complete. This helps improve the stability of Elasticsearch clusters and prevent out-of-memory (OOM) exceptions.	7.6.2, 7.10.2	**Configuring Large Query Isolation**
	Kibana query response circuit breaker	The Kibana query circuit breaker estimates heap usage before buffering the response body. It automatically discards oversized responses to prevent the Kibana backend process from crashing due to out-of-memory (OOM) errors.	7.10.2	**Configuring the Kibana Query Response Circuit Breaker**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Category	Feature	Description	Supported Version	Details
	Automatic retry for failed shard allocation	By configuring automatic retry for failed shard allocation with an exponential backoff strategy, you can enable automatic recovery from shard allocation failures. Such failures may occur due to high node load or a triggered circuit breaker.	7.10.2	**Configuring Automatic Retry for Failed Shard Allocation**
Enhanced performance	Enhanced data ingestion performance	CSS provides several ways to enhance ingestion performance: bulk routing, bulk aggregation, text indexing acceleration, and merge optimization. They can help reduce performance overhead and significantly improve write stability and throughput in heavy-load scenarios.	7.10.2	**Enhancing Data Ingestion Performance**
	Enhanced aggregation performance	CSS enhances aggregation performance in the face of large data volumes by leveraging vectorization and optimized clustering techniques, enabling faster analytics and decision-making in complex situations.	7.10.2	**Configuring Enhanced Aggregation**
Enhanced operations	Index recycle bin	To prevent data loss caused by accidental deletion, CSS provides an index recycle bin. When it is enabled, deleted indexes are temporarily stored in the recycle bin, allowing for recovery before they are permanently removed. This feature improves data reliability and operational security.	7.10.2	**Enabling the Index Recycle Bin**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Category	Feature	Description	Supported Version	Details
	Query resource tracker	O&M personnel can call an API to obtain top queries with the highest latency, CPU usage, or memory consumption, filter the queries by time range, and quickly identify problematic queries. This can significantly improve troubleshooting efficiency and accuracy.	7.10.2	**Tracking Query Resource Consumption**

2.11.2 Configuring Decoupled Storage and Compute

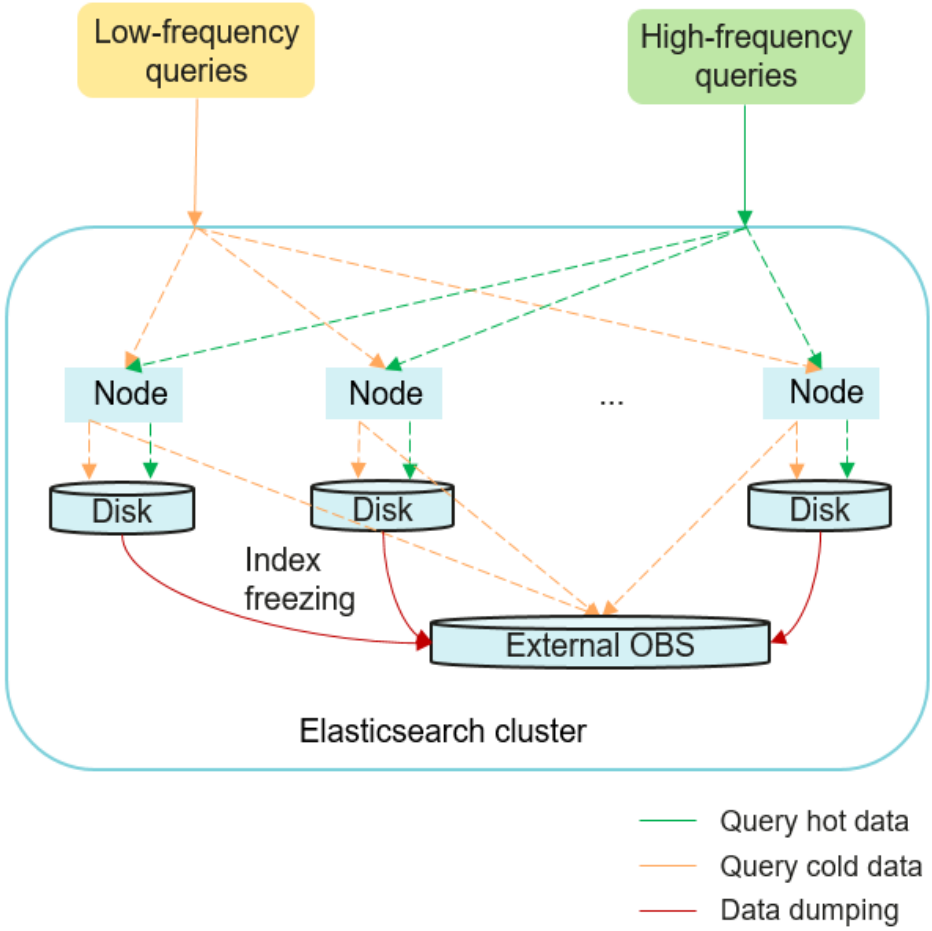
As data grows exponentially, traditional monolithic storage architectures struggle to balance performance and cost efficiency. To support high-speed real-time queries while lowering the cost of storing massive quantities of data, CSS offers the storage-compute decoupling feature for Elasticsearch clusters.

With decoupled storage and compute, hot data that is frequently accessed is stored in high-performance storage media, while cold data that is infrequently accessed is migrated to low-cost storage media—Object Storage Service (OBS). This ensures real-time query performance for hot data while reducing long-term storage costs.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

How the Feature Works

Figure 2-34 Decoupled storage and compute architecture



This is how storage-compute decoupling works:

1. Data writing

   Hot data is written to high-performance local disks (SSDs) to ensure real-time query performance.

2. Hot-to-cold transition

   Infrequently accessed data is migrated from SSDs to cost-effective OBS (OBS buckets are invisible to users), turning it from hot data to cold data. You can call the index freezing API to manually turn hot data to cold data, or define a lifecycle policy (for example, a 90-day data retention period) to have it triggered automatically. Cold data itself is stored in OBS for cost efficiency, but its metadata and date fields are maintained on local disks to enable rapid search.

3. Query processing

   - Hot data is read directly from local disks with millisecond latency.
   - For cold data, the system uses metadata retained on local disks to quickly locate it in OBS, load it, and return it to the user.

In CSS, indexes may go through the following stages during their entire lifecycle.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- Hot index (open): Writes are allowed. Query latency is in milliseconds.
- Frozen index (freeze_low_cost): Rarely accessed data is stored in OBS. The index becomes read-only, and query latency is in seconds or even minutes.
- Deleted index: Frozen indexes are deleted periodically to reclaim storage resources.

Decoupled storage and compute can be automated via index lifecycle management policies. For details, see **Decoupling Storage and Compute Using ISM**.

Impact on Billing

Upon freezing, indexes are dumped to OBS, and the storage class is Standard. OBS buckets that store frozen indexes are invisible to users—users cannot view or manage them on the OBS console. Note that storing indexes in OBS buckets incurs additional fees, billed on a pay-per-use basis according to consumption of Standard storage capacity. For details, see **Cloud Search Service Price Calculator**.

Constraints

- Only Elasticsearch 7.6.2 and 7.10.2 clusters support decoupled storage and compute.
- While an index is being frozen, the system sets the index to the read-only state. Even after the data in the index is dumped to OBS, the index remains read-only and no data can be written into the index.
- During index freezing, the data in it can still be queried. After the freezing is complete, the index is closed and then re-opened. During this transition period, the cluster may be in the red state temporarily, and the index becomes unsearchable. The index becomes searchable again after being re-opened.
- After an index is frozen, its data is dumped to an OBS bucket, and any index data is deleted from local disks. A frozen index has an increased query latency. During an aggregation operation, the latency becomes even longer because the query is complex and a large amount of data needs to be retrieved.
- A frozen index with data already dumped to OBS cannot be unfrozen. That is, a read-only index cannot be rolled back to writable.
- In the case of two clusters configured with read/write splitting, the secondary cluster cannot set storage-compute decoupling for indexes synchronized from the primary cluster.
- Vector indexes do not support storage-compute decoupling.
- When an index is frozen, its data is dumped to an OBS bucket. This process will consume network bandwidth. Make sure data transmission between your cluster and OBS does not exhaust the maximum bandwidth and QPS supported by OBS. If these constraints are violated, the performance of queries that involve OBS will deteriorate. For example, the speed of restoring shards and querying data will be slow.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Logging In to Kibana

Log in to Kibana and go to the command execution page. Elasticsearch clusters support multiple access methods. This topic uses Kibana as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Freezing an Index

Freezing an index means to move cold data from solid state drives (SSDs) to OBS, freeing up SSD storage for hot, frequently accessed data. A frozen index becomes read-only.

1. Freeze an index.

   Run the following command to dump the data of a specified index to OBS and set its storage class to Standard:

   ```
   POST {index_name}/_freeze_low_cost
   ```

   Table 2-110 Parameter description

Parameter	Type	Default Value	Description
index_name	String	N/A	Specifies one or more indexes. • Single index: Enter the index name, for example, **my_index**. • Multiple indexes: Enter multiple index names and use a comma (,) to separate them, for example, **my_index1,my_index2**. • Wildcard: Use the wildcard (*) to match multiple indexes. For example, **myindex*** matches all indexes whose name starts with **myindex**.

   Example response:

   ```
   {
      "freeze_uuid": "pdsRgUtSTymVDWR_HoTGFw" // ID of the index freezing task, which you can use to query the task progress.
   }
   ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

CAUTION

If the Elasticsearch cluster version is 7.10.2 and the image version is 7.10.2_25.9.0_xxx or later, index freezing tasks are not executed immediately upon submission. Instead, they are temporarily cached for a configurable period (default: one minute) and then processed in batches. The caching rules for asynchronous index freezing tasks can be customized. For details, see **Modify batch index freezing settings**.

2. Check the progress of the index freezing task.

   Run the following command to check the index freezing progress by specifying the task ID:

   ```
   GET _freeze_low_cost_progress/{freeze_uuid}
   ```

   If the Elasticsearch cluster version is 7.10.2 and the image version is 7.10.2_25.9.0_xxx or later, the progresses of multiple index freezing tasks of the same batch are returned.

   Example response:

   ```
   {
     "stage" : "STARTED",
     "shards_stats" : {
       "INIT" : 0,
       "FAILURE" : 0,
       "DONE" : 0,
       "STARTED" : 1,
       "ABORTED" : 0
     },
     "indices" : {
       "data1" : [
         {
           "uuid" : "7OS-G1-tRke2jHZPlckexg",
           "index" : {
             "name" : "data1",
             "index_id" : "4b5PHXJITLaS6AurImfQ9A",
             "shard" : 2
           },
           "start_ms" : 1611972010852,
           "end_ms" : -1,
           "total_time" : "10.5s",
           "total_time_in_millis" : 10505,
           "stage" : "STARTED",
           "failure" : null,
           "size" : {
             "total_bytes" : 3211446689,
             "finished_bytes" : 222491269,
             "percent" : "6.0%"
           },
           "file" : {
             "total_files" : 271,
             "finished_files" : 12,
             "percent" : "4.0%"
           },
           "rate_limit" : {
             "paused_times" : 1,
             "paused_nanos" : 946460970
           }
         }
       ]
     }
   }
   ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-111 Response parameters

Parameter	Description
stage	Status of the index freezing task. Value range: • **INIT**: The task has just started or is being initialized. • **FAILURE**: failed • **DONE**: complete • **STARTED**: started • **ABORTED**: canceled. This field is reserved.
shards_stats	Number of shards in each state.
indices	Status details of each index, as described in **Table 2-112**.

Table 2-112 indices parameters

Parameter	Description
uuid	ID of the index freezing task.
index	Index and shard information.
start_ms	Task start time.
end_ms	Task end time. If the task is still ongoing, **-1** is displayed.
total_time	How long the task has been ongoing.
total_time_in_millis	How long the task has been ongoing, in milliseconds.
stage	Shard status.
failure	Task failure cause. If no failure has occurred, **null** is returned.
size.total_bytes	Total bytes to be frozen.
size.finished_bytes	Number of bytes that have been frozen.
size.percent	Freezing progress in terms of bytes that have been frozen vs total bytes to freeze.
file.total_bytes	Total bytes to be frozen.
file.finished_bytes	Number of bytes that have been frozen.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
file.percent	Freezing progress in terms of the number of documents that have been frozen vs. total number of documents to freeze.
rate_limit.paused_times	Number of times freezing has been suspended due to rate limiting.
rate_limit.paused_nanos	Duration of freezing task suspension due to rate limiting, in nanoseconds.

When index freezing is complete, the settings parameters will be returned, as described in **Table 2-113**.

Table 2-113 settings parameters

Parameter	Description
index.frozen_low_cost	Frozen index. The value is fixed to **true**.
index.blocks.write	Writes disallowed to a frozen index. The value is fixed to **true**.
index.store.type	Index storage type. The value is fixed to **obs**.

3. Query the list of frozen indexes.

   NOTE

   Here, only some of the key parameters for querying the frozen index list are described. For more information about the API, see **cat indices API**.

   Run the following command to query the frozen index list based on index freezing status:

```
GET _cat/freeze_indices?stage={STAGE}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-114 Parameter description

Parameter	Type	Default Value	Description
STAGE	String	N/A	Index freezing stage. Value range: • **start**: Freezing has started and is ongoing. • **done**: Freezing has completed. • **unfreeze**: Freezing has not started. • Empty or any other value: Freezing is ongoing or has completed.

Example response:

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

Optimizing Cache Settings for Frozen Indexes

After an index is frozen, its data is dumped to OBS. To reduce direct data retrieval from OBS and thus improve query performance, some data is cached in the cluster. Data that is requested for the first time is directly retrieved from OBS. The retrieved data is then cached in the cluster memory. In response to subsequent queries, the system searches the cache first. CSS allows you to query cache statistics for frozen indexes that are stored in OBS buckets. You can also reset the cache status and modify cache configuration.

- Query cache statistics for frozen indexes stored in OBS buckets.
  - Run the following command to query cache statistics for frozen indexes on all nodes:
    ```
    GET _frozen_stats
    ```
  - Run the following command to query cache statistics for frozen indexes on specified nodes:
    ```
    GET _frozen_stats/{node_id}
    ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-115 Parameter description

Parameter	Type	Default Value	Description
node_id	String	N/A	Specifies one or more cluster nodes. • Single node: Enter the node ID. • Multiple nodes: Enter multiple node IDs and use a comma (,) to separate them. You can run the following command to obtain node IDs: `GET _cat/nodes?s=n&h=n,id&v=true&full_id=true`

The following is an example of the returned information for querying frozen index cache statistics on all nodes:

```
{
  "_nodes" : {
    "total" : 1,     // Total number of nodes
    "successful" : 1,      // Successful nodes
    "failed" : 0      // Failed nodes
  },
  "cluster_name" : "css-zzz1",             //Cluster name
  "nodes" : {
    "7uwKO38RRoaON37YsXhCYw" : {
      "name" : "css-zzz1-ess-esn-2-1",        //Node name
      "transport_address" : "10.0.0.247:9300",    //Node transport address
      "host" : "10.0.0.247",             //Node host
      "ip" : "10.0.0.247",             //Node IP address
      "block_cache" : {
        "default" : {
          "type" : "memory",             //Cache type. memory indicates in-memory cache.
          "block_cache_capacity" : 8192,     //Cache capacity
          "block_cache_blocksize" : 8192,     //Single block size in the cache, in bytes. In the example, the block size is 8 KB.
          "block_cache_size" : 12,          //Cache capacity used
          "block_cache_hit" : 14,          //Number of cache hits
          "block_cache_miss" : 0,          //Number of cache misses
          "block_cache_eviction" : 0,          //Number of cache evictions
          "block_cache_store_fail" : 0          //Number of cache storage failures, which occur when the cache is full.
        }
      },
      "obs_stats" : {
        "list" : {
          "obs_list_count" : 17,          //Number of times the OBS list API was called.
          "obs_list_ms" : 265,             //Total length of time spent calling the OBS list API.
          "obs_list_avg_ms" : 15          //Average time spent calling the OBS list API.
        },
        "get_meta" : {
          "obs_get_meta_count" : 79,     //Number of times the OBS get metadata API was called.
          "obs_get_meta_ms" : 183,     //Total length of time spent calling the OBS get metadata API.
          "obs_get_meta_avg_ms" : 2     //Average time spent calling the OBS get metadata API.
        },
        "get_obj" : {
          "obs_get_obj_count" : 12,     //Number of times the OBS get object API was called.
          "obs_get_obj_ms" : 123,     //Total length of time spent calling the OBS get object API.
          "obs_get_obj_avg_ms" : 10     //Average time spent calling the OBS get object API.
        },
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
      "put_obj" : {
        "obs_put_obj_count" : 12,     //Number of times the OBS put object API was called.
        "obs_put_obj_ms" : 2451,     //Total length of time spent calling the OBS put object API.
        "obs_put_obj_avg_ms" : 204     //Average time spent calling the OBS put object API.
      },
      "obs_op_total" : {
        "obs_op_total_ms" : 3022,     //Total length of time spent calling OBS APIs.
        "obs_op_total_count" : 120,     //Total number of times calling OBS APIs.
        "obs_op_avg_ms" : 25          //Average time spent calling OBS APIs.
      }
    },
    "reader_cache" : {
      "hit_count" : 0,
      "miss_count" : 1,
      "load_success_count" : 1,
      "load_exception_count" : 0,
      "total_load_time" : 291194714,
      "eviction_count" : 0
    }
  }
 }
}
```

- Reset the cache status for frozen indexes stored in OBS buckets.

  Run the following command to reset the cache status for frozen indexes:

  ```
  POST _frozen_stats/reset
  ```

  This command is used to troubleshoot performance issues. For example, if you reset the cache status before executing a query, you can see the cache hit results clearly. It is not advisable to use this command during runtime.

  Example response:

  ```
  {
   "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
   },
   "cluster_name" : "Es-0325-007_01",
   "nodes" : {
    "mqTdk2YRSPyOSXfesREFSg" : {
     "result" : "ok"
    }
   }
  }
  ```

- Modify cache configuration for frozen indexes.

  Elasticsearch accesses different types of files using different methods. The cache system supports multi-level caches and uses blocks of different sizes to cache different types of files. For example, typically, a large number of small blocks are used to cache .fdx and .tip files, while a small number of large blocks are used to cache .fdt files. You can modify the cache configuration based on service needs. **Table 2-116** describes the parameters that can be configured.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-116 Configuring cache settings for frozen indexes

Parameter	Type	Default Value	Description
low_cost.obs.blockcache.names	Array	["default"]	A list of the names of multi-level caches. The cache system supports multi-level caches for data of different access granularities. There is a default cache named **default**. If you configure this parameter, the value must include at least the **default** cache in addition to other custom cache names.
low_cost.obs.blockcache.<NAME>.type	ENUM	memory	Storage type of the cache named <NAME>. Value range: • **memory**: The cache uses in-memory storage, which means it consumes memory resources. • **file**: The cache uses disk storage. Its capacity is limited by available disk space. In this case, you are advised to use ultra-high I/O disks to improve cache performance.
low_cost.obs.blockcache.<NAME>.blockshift	Integer	13 (indicating 8 KB)	Block size in the cache named <NAME> (in bytes). Formula: $2^{blockshift}$. For example, if the value is 16, the block size is 2^{16} bytes, that is, 65536 bytes or 64 KB. Value: number of left-shifts
low_cost.obs.blockcache.<NAME>.bank.count	Integer	1	Number of partitions in the cache named <NAME>. Each partition is an independent unit in the cache.
low_cost.obs.blockcache.<NAME>.number.blocks.perbank	Integer	8192	Number of blocks in each partition of the cache named <NAME>.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
low_cost.obs.blockcache.<NAME>.exclude.file.types	Array	N/A	List of file name extensions for files that cannot be cached in the cache named <NAME>. If a file's extension is not in **exclude.file.types** or **<NAME>.file.types** (if configured), the default cache policy applies.
low_cost.obs.blockcache.<NAME>.file.types	Array	N/A	List of file name extensions for files that can be cached in the cache named <NAME>. If a file's extension is not in **exclude.file.types** or **<NAME>.file.types** (if configured), the default cache policy applies.
index.frozen.obs.max_bytes_per_sec	String	150MB	Maximum rate at which the documents of frozen indexes are uploaded to OBS. Value format: number + unit • Number: a natural number • Unit: B, K, KB, M, MB, G, GB, T, TB, P, or PB (case-insensitive) Value range: 0 to (2^{63} – 1) bytes The value **0** indicates no limit. The change takes effect immediately.
low_cost.obs.index.upload.threshold.use.multipart	String	1GB	Multipart upload threshold. In the process of dumping frozen indexes to OBS, if the size of a single document exceeds this threshold, a multipart upload is performed. Value format: number + unit • Number: a natural number • Unit: B, K, KB, M, MB, G, GB, T, TB, P, or PB (case-insensitive) Value range: 0 to (2^{63} – 1) bytes The value **0** indicates there is no multipart upload.

The following is a typical cache configuration example. A two-level cache system is used: **default** and **large**. The **default** cache has a total of 30 x 4096 64-KB blocks. It is used to cache non-.fdt files. The **large** cache has 5 x 1000 2-MB blocks. It is used to cache .fdx, .dvd, and .tip files.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
low_cost.obs.blockcache.names: ["default", "large"]
low_cost.obs.blockcache.default.type: file
low_cost.obs.blockcache.default.blockshift: 16
low_cost.obs.blockcache.default.number.blocks.perbank: 4096
low_cost.obs.blockcache.default.bank.count: 30
low_cost.obs.blockcache.default.exclude.file.types: ["fdt"]

low_cost.obs.blockcache.large.type: file
low_cost.obs.blockcache.large.blockshift: 21
low_cost.obs.blockcache.large.number.blocks.perbank: 1000
low_cost.obs.blockcache.large.bank.count: 5
low_cost.obs.blockcache.large.file.types: ["fdx", "dvd", "tip"]
```

- Modify batch index freezing settings.

  If the Elasticsearch cluster version is 7.10.2 and the image version is 7.10.2_25.9.0_xxx or later, index freezing tasks are not executed immediately upon submission. Instead, they are temporarily cached for a configurable period and then processed in batches. The caching rules for asynchronous index freezing tasks can be customized. **Table 2-117** describes relevant parameters.

  Table 2-117 Parameters for configuring batch index freezing

Parameter	Type	Default Value	Description
low_cost.freeze.job.interval	TimeValue	1m	Batch index freezing task execution interval. Value range: 10s to 10m If this parameter is set to 1m, the system caches index freezing requests for 1 minute and then executes them all at once.
low_cost.freeze.job.batch_size	Integer	20	The maximum number of indexes that can be included in a single batch index freezing task. Value range: 1 to 100 If this parameter is set to 20, each batch can contain 20 indexes. For example, if an Index Lifecycle Management (ILM) task submits 100 indexes for freezing at once, they will be divided into five batches and frozen one batch at a time.
low_cost.freeze.job.start_time	String	00:00 (executable all day)	Start time for batch index freezing tasks, meaning no such tasks can be started before the specified time. If a submitted index freezing task has not yet started, it will have to wait until the specified start time. Value format: hour:minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
low_cost.freeze.job.end_time	String	23:59 (executable all day)	Start end time for batch index freezing tasks, meaning no such tasks can be started after the specified time. Ongoing tasks are not affected by this setting. Value format: hour:minute

For example, run the following command to modify the default batch index freezing settings:

```
PUT _cluster/settings
{
  "persistent": {
    "low_cost.freeze.job.interval": "5m",
    "low_cost.freeze.job.batch_size": 30,
    "low_cost.freeze.job.start_time": "22:00",
    "low_cost.freeze.job.end_time": "07:00"
  }
}
```

Improving Query Performance for Frozen Indexes

WARNING

Only Elasticsearch 7.6.2 and 7.10.2 clusters created after February 2023 support this feature.

When frozen indexes are queried on the **Discover** page of Kibana for the first time, all data needs to be retrieved from OBS because there is no data in the cache. If a large number of documents need to be returned, it takes a long time to retrieve the corresponding time fields and document metadata from OBS. By caching this part of data in the cluster, query performance can be improved significantly. This is how CSS improves query performance for frozen indexes. Local cache settings are preset. You can review them and modify them as needed.

1. Modify local cache settings for frozen indexes. **Table 2-118** describes these settings. All of them can be modified dynamically, and the changes take effect immediately.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-118 Local cache settings

Parameter	Type	Default Value	Description
low_cost.local_cache.max.capacity	Integer	500	Maximum number of cold data caches that can be made available on each node. (Each shard corresponds to a cache.) Value range: 10–5000 • If the heap memory usage remains high, decrease this value. • If the value of **load_overflow_count** keeps increasing rapidly, increase this value.
index.low_cost.local_cache.threshold	Integer	50	The percentage of **date** fields in an index as a threshold for enabling or disabling the local caching of cold data. • If in an index, the percentage of **date** fields is lower than this value, cold data of the **date** type will be cached locally. Otherwise, such data will not be cached. • If date fields account for the vast majority of the total data size of the current index, you should not cache its data locally. Value range: 0–100 Unit: %
index.low_cost.local_cache.evict_time	String	30d	Retention duration of cold data in the local cache. The value is determined by **index.frozen_date** (time of index freezing). If **index.frozen_date** is unavailable, the value is determined by the index creation time. You are advised to adjust the retention duration based on your disk usage. Value range: 1 to 365 days Unit: d (days)

The following provides some examples:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- Run the following command to change the maximum number of cold data caches that can be made available on each node:

```
PUT _cluster/settings
{
  "persistent": {
    "low_cost.local_cache.max.capacity": 1000
  }
}
```

- Run the following command to change the threshold (in the form of the percentage of **date** fields in an index) for enabling or disabling the local caching of **date** fields as cold data:

```
PUT {index_name}/_settings
{
  "index.low_cost.local_cache.threshold": 20
}
```

- Run the following command to change the retention duration of cold data in the local cache:

```
PUT {index_name}/_settings
{
  "index.low_cost.local_cache.evict_time": "7d"
}
```

2. Query local cache statistics for frozen indexes.
   - Query local cache statistics for frozen indexes on all nodes:

     ```
     GET /_frozen_stats/local_cache
     ```

   - Query local cache statistics for frozen indexes on specified nodes:

     ```
     GET /_frozen_stats/local_cache/{node_id}
     ```

   Example response:

```
{
  "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  },
  "cluster_name" : "test",
  "nodes" : {
    "6by3lPy1R3m55Dcq3liK8Q" : {
      "name" : "node-1",
      "transport_address" : "127.0.0.1:9300",
      "host" : "127.0.0.1",
      "ip" : "127.0.0.1",
      "local_cache" : {
        "get_stats" : {
          "get_total_count" : 562,              //Total number of times data was retrieved from the local cold data cache.
          "get_hit_count" : 562,                //Total number of hits in the local cold data cache.
          "get_miss_count" : 0,                 //Total number of local cold data cache misses.
          "get_total_ns" : 43849200,            //Total duration for retrieving data from the local cold data cache.
          "get_avg_ns" : 78023                  //Average duration for retrieving data from the local cold data cache.
        },
        "load_stats" : {
          "load_count" : 2,                   //Number of times cold data was loaded from the local cache
          "load_total_ms" : 29,                 //Total duration for loading cold data from the local cache
          "load_avg_ms" : 14,                   //Average duration for loading cold data from the local cache
          "load_fail_count" : 0,                //Number of failures for loading cold data from the local cache
          "load_overflow_count" : 0         //Number of times the local cold data cache exceeds the cache pool size.
        },
        "reload_stats" : {
          "reload_count" : 0,                 //Number of times the local cold data cache was regenerated.
          "reload_total_ms" : 0,                //Total duration for regenerating the local cold data cache.
          "reload_avg_ms" : 0,                  //Average duration for regenerating the local cold data cache.
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        "reload_fail_count" : 0            //Number of failures in regenerating the local cold data
cache.
      },
      "init_stats" : {
        "init_count" : 0,                 //Number of times the local cold data cache was initialized.
        "init_total_ms" : 0,              //Total duration for initializing the local cold data cache.
        "init_avg_ms" : 0,                //Average duration for initializing the local cold data cache.
        "init_fail_count" : 0             //Number of failures in initializing the local cold data cache.
      }
    }
  }
 }
}
```

Querying the Real-Time Rates of OBS in Handling Cold Data

WARNING

Only Elasticsearch 7.6.2 and 7.10.2 clusters created after February 2023 support this feature.

To help you understand how the storage-compute decoupling plug-in is working with OBS, an API for collecting statistics on the real-time rates of OBS has been added, and the real-time rates are recorded in the index **.freeze_obs_rate-YYYY.mm.dd**.

Calculation method: The average OBS operation rates in the last 5 seconds are calculated every 5 seconds.

The system index **.freeze_obs_rate-YYYY.mm.dd** records statistics on OBS real-time operation rates, helping you understand relevant trends about the OBS resources that store cold data. The default retention period of this index is 30 days.

1. Querying the real-time rates of OBS in handling cold data.
   - Run the following command to query the real-time OBS rates on all nodes:
     ```
     GET _frozen_stats/obs_rate
     ```
   - Run the following command to query the real-time OBS rates on specified nodes:
     ```
     GET _frozen_stats/obs_rate/{node_id}
     ```

   Example response:

```
{
  "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  },
  "cluster_name" : "test",
  "nodes" : {
    "dflDvcSwTJ-fkillT2zE3A" : {
      "name" : "node-1",
      "transport_address" : "127.0.0.1:9300",
      "host" : "127.0.0.1",
      "ip" : "127.0.0.1",
      "update_time" : 1671777600482,              // Time when the current statistics were updated.
      "obs_rate" : {
        "list_op_rate" : 0.0,                   // Rate of OBS list operations. Unit: times/s.
        "get_meta_op_rate" : 0.0,                 // Rate of OBS get meta operations. Unit: times/s.
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
      "get_obj_op_rate" : 0.0,                // Rate of OBS get operations. Unit: times/s.
      "put_op_rate" : 0.0,                    // Rate of OBS put operations. Unit: times/s.
      "obs_total_op_rate" : 0.0,              // Rate of all OBS operations. Unit: times/s.
      "obs_upload_rate" : "0.0 MB/s",         // OBS data upload rate. Unit: MB/s.
      "obs_download_rate" : "0.0 MB/s"        // OBS data download rate. Unit: MB/s.
    }
   }
  }
}
```

2. Modify the retention period of the **.freeze_obs_rate-YYYY.mm.dd** index that stores the OBS real-time rates. The default retention period of this index is 30 days.

   Run the following command to change the index retention period to seven days:

```
PUT _cluster/settings
{
  "persistent": {
    "low_cost.obs_rate_index.evict_time": "7d"
  }
}
```

Table 2-119 Parameter description

Parameter	Type	Default Value	Description
low_cost.obs_rate_index.evict_time	String	30d	The retention period of the **.freeze_obs_rate-YYYY.mm.dd** index on each node. • Value range: 1 to 365 days • Unit: d (days) Any change takes effect immediately.

Related Documents

To ensure acceptable performance for cold data queries (for example, when historical data is queried periodically), use a tiered storage solution. For details, see **Switching Between Hot and Cold Data**.

2.11.3 Switching Between Hot and Cold Data

In an Elasticsearch cluster, switching between hot and cold data storage means to allocate data to nodes of different performance standards based on data temperature (that is, how often data is accessed). The goal is to achieve optimal storage costs and query performance.

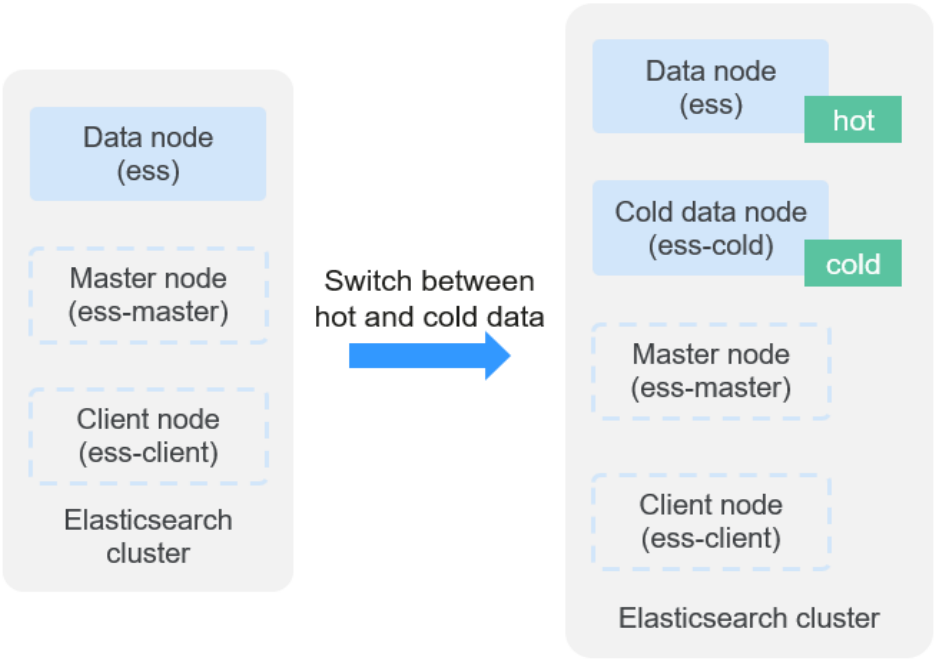
- Hot data: frequently accessed data, low latency required, stored using high-performance hardware (such as SSDs) to ensure fast write and query performance.
- Cold data: rarely accessed data, cost-optimized storage (such as HDDs).

If your cluster stores data used for different purposes, such as real-time analytics, log analytics, and monitoring data archives, you can switch between hot and cold storage for specified indexes to balance performance and costs.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

How the Feature Works

Figure 2-35 How cold/hot storage switchover works



The key is to allocate index data storage by node labels and index allocation policies.

- Node labels:
  - Data node (hot): Stores real-time data (hot data) by default and supports high-concurrency read/write requests.
  - Cold data node (cold): Stores historical data (cold data). Cold data nodes use less expensive hardware and deliver lower query performance than regular data nodes.
- Index allocation: You can configure index templates or directly specify index names to allocate specific indexes to data nodes or cold data nodes.

Procedure: Enable cold data nodes, and configure an index template or specific index settings to allocate index data storage. The cluster automatically allocates data based on your settings.

CAUTION

In comparison with data nodes, cold data nodes deliver lower query performance. Determine which type of node to use based on your service needs.

Switching Over Between Hot and Cold Storage

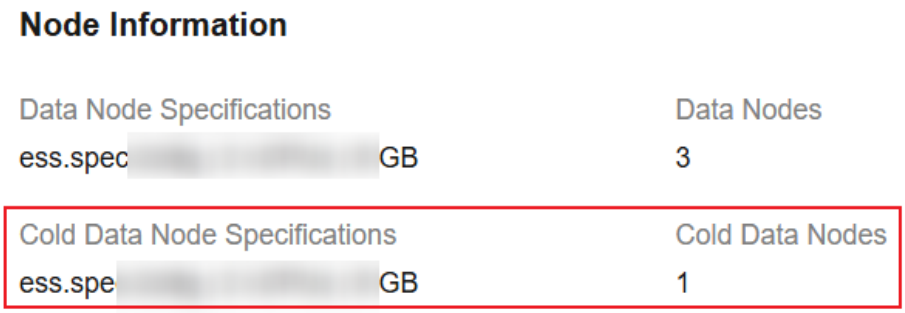
Step 1 Check whether cold data nodes are enabled in the target cluster.

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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, check whether the **Node Information** area contains cold data node information.

Figure 2-36 Cold data node information



   - If there is information about cold data nodes, the cluster has cold data nodes. Go to the next step.
   - If not, the cluster does not have cold data nodes. Add cold data nodes for the cluster and then go to the next step. For details about how to add cold data nodes, see **Adding New Node Types**.

Step 2 Log in to the Kibana console.

1. On the cluster information page, click **Kibana** in the upper-right corner to log in to Kibana.
2. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Step 3 On the Kibana page, configure index allocation policies.

You can configure index templates or directly specify index names to allocate specific indexes to data nodes or cold data nodes.

- **Configuring an index template**

  Configure an index template to allocate matching indexes to cold or hot data nodes. For example, run the following command to store newly created indexes whose name starts with **myindex** to cold data nodes. (The command varies slightly depending on the Elasticsearch version.)

  - For Elasticsearch 6.x or later:

```
PUT _template/{template_name}
{
  "order": 1,
  "index_patterns": "myindex*",
  "settings": {
    "refresh_interval": "30s",
    "number_of_shards": "3",
    "number_of_replicas": "0",
    "routing.allocation.require.box_type": "cold"
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

– For Elasticsearch earlier than 6.x:

```
PUT _template/{template_name}
{
   "order": 1,
   "template": "myindex*",
   "settings": {
      "index": {
         "refresh_interval": "30s",
         "number_of_shards": "3",
         "number_of_replicas": "0",
         "routing.allocation.require.box_type": "cold"
      }
   }
}
```

Table 2-120 Key parameters

Parameter	Type	Description
index_patterns	String	Defines the index name matching pattern. The wildcard (*) can be used to match multiple indexes. For example, **myindex*** indicates all indexes whose name starts with **myindex**.
routing.allocation.require.box_type	String	Specifies the data node type for shard allocation. The value can be: – **cold**: Shards are allocated to cold data nodes. – **hot**: Shards are allocated to regular data nodes. If this parameter is not specified, shards will be evenly and randomly distributed among cold and hot data nodes.

- **Configuring specific indexes**

  To change the node type for an existing index, run the following command:

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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-121 Parameter description

Parameter	Type	Description
index_name	String	Specifies one or more indexes. – Single index: Enter the index name, for example, **my_index**. – Multiple indexes: Enter multiple index names and use a comma (,) to separate them, for example, **my_index1,my_index2**. – Wildcard: Use the wildcard (*) to match multiple indexes. For example, **myindex*** indicates all indexes whose name starts with **myindex**.
routing.allocation.require.box_type	String	Specifies the data node type for shard allocation. The value can be: – **cold**: Shards are allocated to cold data nodes. – **hot**: Shards are allocated to regular data nodes. If this parameter is not specified, shards will be evenly and randomly distributed among cold and hot data nodes.

Step 4 Verify the switchover between hot and cold storage.

Run the following command to check the distribution of index shards:

```
GET _cat/shards/{index_name}?v
```

When the data volume is large, the switchover may take a long time. There may be an intermediate state where the data of an index resides on both cold data nodes and data nodes.

As shown in the following figure, all shards of the **myindex** index are stored on the cold data node **css-e668-ess-cold-esn-1-1**.

```
index    shard prirep state      docs  store ip             node
myindex 1     p      STARTED 14085446 17.8gb 192.168.91.188 css-e668-ess-cold-esn-1-1
myindex 2     p      STARTED 14094005 17.9gb 192.168.91.188 css-e668-ess-cold-esn-1-1
myindex 0     p      STARTED 14094742 17.8gb 192.168.91.188 css-e668-ess-cold-esn-1-1
```

Step 5 Roll back the cold-hot switchover configuration.

To cancel the cold-hot switchover configuration, run the following command:

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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

After the rollback, index data will be evenly and randomly allocated to both cold data nodes and data nodes.

----End

Related Operations

- You can scale cold data nodes by adding or reducing nodes or their storage capacity, or adding cold data nodes dynamically. For details, see **Managing Cluster Changes**.
- If your cluster does not have cold data nodes but you wish to cut storage costs, you can try using decoupled storage and compute. For details, see **Configuring Decoupled Storage and Compute**.

2.11.4 Configuring Read/Write Splitting Between Leader and Follower

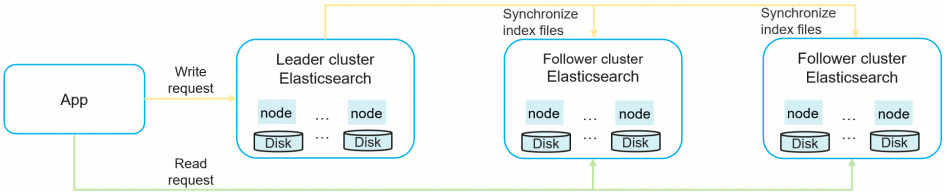
As business expands, and data volumes and access requests grow exponentially, a monolithic architecture where a single cluster handles both write and query requests faces typical challenges like resource contention and overload. To address these challenges, CSS introduces read/write splitting between Elasticsearch clusters.

Elasticsearch read/write splitting works by having leader and follower clusters collaborate together. The feature delivers the following benefits:

- Decoupled read and write loads: The leader cluster ensures data ingestion performance, while the follower clusters deliver scalable, high-concurrency query performance. There is no more resource contention, and peak loads are reduced.
- Flexible scalability: The write and query clusters can be scaled horizontally and independently. Cross-region cluster deployment is supported.
- Data consistency guarantee: Data can be synchronized in real time, with low latency. Incremental synchronization is also supported.

How the Feature Works

Figure 2-37 Read/write splitting



How read/write splitting between Elasticsearch clusters works:

1. Data writes: Users send write requests, and the leader cluster handles these requests.
2. Data synchronization.

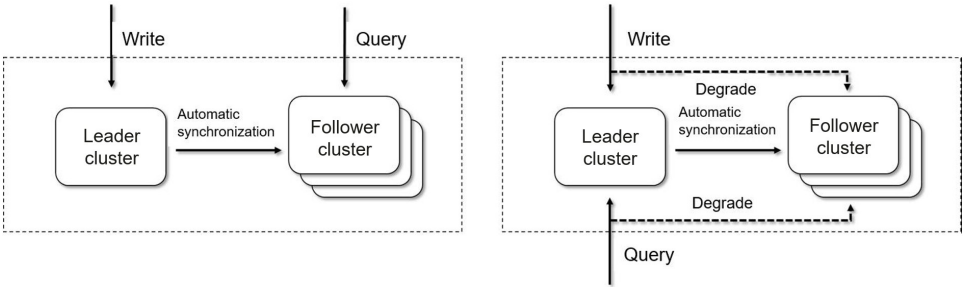
   The leader synchronizes data changes to the follower through a REST API. Two synchronization methods are supported:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

  - Synchronizing a specified index.
  - Synchronizing matched indexes, for example, indexes that match a wildcard expression.
3. Query processing: Users send query requests. The follower cluster handles these requests and returns the results.

Figure 2-38 illustrates how read/write splitting works.

Figure 2-38 How read/write splitting works



- Read and write split when both clusters are available (left): The leader handles writes, and the follower handles queries.
- Leader-to-follower switchover when the leader fails (right): If the leader becomes unavailable, the follower automatically upgrades to leader to ensure service continuity. For details, see **Case: Leader/Follower Switchover**.

Constraints

- Only Elasticsearch 7.6.2 and Elasticsearch 7.10.2 clusters support read/write splitting.
- The versions of the leader and follower clusters must be kept consistent, or errors may occur.

Prerequisites

Two Elasticsearch clusters of the same version have been created. One functions as the leader, and the other the follower. The follower cluster must be able to access the REST API (default port: 9200) of the leader cluster.

Logging In to Kibana

Log in to Kibana and go to the command execution page. Elasticsearch clusters support multiple access methods. This topic uses Kibana as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Connecting the Leader and Follower Clusters

1. In the **follower cluster**, run the following command to configure the leader cluster's information to connect them, so that data can be synchronized from the leader to the follower.

```
PUT /_cluster/settings
{
  "persistent" : {
    "cluster" : {
      "remote.rest" : {
        "{leader_name}" : {
          "seeds" : [
            "http://10.0.0.1:9200",
            "http://10.0.0.2:9200",
            "http://10.0.0.3:9200"
          ] ,
            "username": "test",
            "password": "*****"
        }
      }
    }
  }
}
```

Table 2-122 Parameters for connecting the leader and follower clusters

Parameter	Mandatory	Type	Description
leader_name	Yes	String	Custom name of the leader cluster configuration task, used to identify the leader cluster connection in subsequent index synchronization configuration. Only letters (both uppercase and lowercase), digits, underscores (_), and hyphens (-) are allowed. Spaces and special characters (such as . # @) are not allowed.
seeds	Yes	String	List of addresses for accessing the leader cluster. • For a non-security mode cluster or a security-mode cluster that uses HTTP, the address format is http://xx.xx.xx.xx:9200. • For a security-mode cluster that uses HTTPS, the address format is https://xx.xx.xx.xx:9200.
username	Yes	String	Username of the leader cluster. This parameter is required only when security mode is enabled for the leader cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
password	Yes	String	Password of the leader cluster. This parameter is required only when security mode is enabled for the leader cluster.

Example response:

```
{
  "acknowledged" : true,  //Whether the operation is successful
  "persistent" : {
    "cluster" : {
      "remote" : {
        "rest" : {
          "leader1" : {
            "seeds" : [
            "http://10.0.0.1:9200",
            "http://10.0.0.2:9200",
            "http://10.0.0.3:9200"
            ] ,
            "username": "test",
             "password": "*****"
          }
        }
      }
    }
  },
  "transient" : { }
}
```

2. After the configuration is complete, run the following command in the follower cluster to check the connection between the follower and leader clusters:

```
GET _remote/rest/info
```

Example response:

```
{
  "leader1" : {
    "connected" : true  //The two clusters are connected.
  }
}
```

Configuring Index Synchronization

1. Create an index synchronization task. There are two synchronization modes: synchronizing a specified index; and synchronizing matched indexes.

Table 2-123 Comparing two different synchronization modes

Synchronization Mode	Synchronizing a specified index	Synchronizing matched indexes

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Synchronization Logic	Manually specify an index that you wish to synchronize from the leader cluster to the follower.	Specify an index matching pattern. For each index (including any new indexes created in the future) that matches this pattern in the leader cluster, the system automatically creates a task that will synchronize it to the follower cluster.
Scenario	When you want to precisely control the index synchronization scope (for example, to synchronize mission-critical indexes only)	When you want to synchronize many dynamically created indexes (for example, indexes automatically named by timestamp or predefined rules).
Synchronization Policy Modification	A synchronization task can be re-delivered for the same index. That is when you can modify the synchronization policy.	After a synchronization task is started, the synchronization policy can no longer be modified. You can only create a new pattern-matching index synchronization task to replace an existing one. Procedure: **Delete an existing task** > **Stopping Index Synchronization** > Create a new task.

The following describes how to configure a task of each type:

- **Synchronizing a specified index**

  Run the following command in the follower cluster to create a single-index synchronization task.

```
PUT start_remote_sync
{
  "remote_cluster": "{leader_name}",
  "remote_index": "{index_name_leader}",
  "local_index": "{index_name_follower}",
  "settings": {
    "number_of_replicas": 4
  },
  "settings_sync_enable": true,
  "settings_sync_patterns": ["*"],
  "settings_sync_exclude_patterns": ["index.routing.allocation.*"],
  "alias_sync_enable": true,
  "state_sync_enable": true
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-124 Parameters for creating a single-index synchronization task

Parameter	Mandatory	Type	Default Value	Description
remote_cluster	Yes	String	N/A	The name of the leader cluster configuration task. The value must be consistent with the value of **leader_name** set in **Connecting the Leader and Follower Clusters**, for example, **leader1**.
remote_index	Yes	String	N/A	The name of the index to be synchronized in the leader cluster. Enter the name of an existing index in the leader cluster, for example, **data_leader**.
local_index	Yes	String	N/A	The name of the index in the follower cluster after synchronization from the leader, for example, **data_follower**. You can keep it the same as **remote_index**, but doing so is not recommended. This is because using identical names for indexes in the leader and follower clusters can lead to confusion during index management.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
settings	No	Map	N/A	Index settings to be modified and their target values after synchronization to the follower cluster. This parameter allows you to flexibly configure index settings (such as the number of replicas and refresh interval) to match the hardware resources or service requirements in the follower cluster. If not configured, the index settings of the leader cluster will be used by default. Key and value: • **key**: the name of the setting, for example, number_of_replicas. • **value**: its new value, for example, 4. The following settings cannot be modified: number_of_shards, version.created, uuid, creation_date, and soft_deletes.enabled.
settings_sync_enable	No	Boolean	false	Whether to periodically synchronize index settings from the leader cluster. The value can be: • **true**: Synchronize index settings from the leader cluster. • **false**: Not to synchronize index settings from the leader cluster when they are updated. Instead, keep the initial settings.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
settings_sync_patterns	No	String	* (matches all settings)	Leader-cluster index settings to be synchronized to the follower cluster. Specify the names of the settings to be synchronized. ● A single setting: Enter the setting name (for example, **index.max_regex_length**). ● Multiple settings: Enter multiple setting names and separate them with a comma (,), for example, **index.max_regex_length,index.max_refresh_listeners**. ● Wildcard: Use the wildcard * to match multiple settings. For example, **index*** matches all settings whose name starts with **index**. Constraints: ● This parameter takes effect only when **settings_sync_enable** is set to **true**. ● If the same setting is configured both in **settings** and **settings_sync_patterns**, the setting in **settings_sync_patterns** takes effect. ● The following settings cannot be synchronized to the follower cluster: index.number_of_shards, index.version.created, index.uuid, index.creation_date, index.history.uuid, index.number_of_replica

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
				s, index.auto_expand_replicas, and index.version.upgraded.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
settings_sync_exclude_patterns	No	String	Null (excludes none)	Leader-cluster index settings not to be synchronized to the follower cluster. Specify the names of the settings not to be synchronized. ● A single setting: Enter the setting name (for example, **index.max_regex_length**). ● Multiple settings: Enter multiple setting names and separate them with a comma (,), for example, **index.max_regex_length,index.max_refresh_listeners**. ● Wildcard: Use the wildcard * to match multiple settings. For example, **index*** matches all settings whose name starts with **index**. Constraints: ● This parameter takes effect only when **settings_sync_enable** is set to **true**. ● If both **settings_sync_patterns** and **settings_sync_exclude_patterns** are configured, only the settings that are in **settings_sync_patterns** but not in **settings_sync_exclude_patterns** will be synchronized to the follower cluster. ● If the indexes you plan to synchronize include

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
				cold indexes, make sure the follower cluster contains cold data nodes. Otherwise, data synchronization will fail. The following configuration is required to synchronize cold indexes to the data nodes in the follower cluster (in such case, these indexes cease to be cold indexes in the follower cluster): `"settings_sync_exclude_patterns": ["index.routing.allocation.require.box_type"]`
alias_sync_enable	No	Boolean	false	Whether to periodically synchronize index aliases from the leader cluster. The value can be: • **true**: Periodically synchronize index aliases from the leader cluster. • **false**: Not to periodically synchronize index aliases from the leader cluster.
state_sync_enable	No	Boolean	false	Whether to periodically synchronize index status from the leader cluster. The value can be: • **true**: Periodically synchronize index status from the leader cluster. • **false**: Not to periodically synchronize index status from the leader cluster.

- **Synchronizing matched indexes**

  Run the following command in the follower cluster to create a pattern-matching index synchronization task:

  ```
  PUT auto_sync/pattern/${pattern_name}
  {
  ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  "remote_cluster": "{leader_name}",
  "remote_index_patterns": "{index_name}",
  "local_index_pattern": "{{remote_index}}-sync",
  "apply_exist_index": true,
  "settings": {
    "number_of_replicas": 4
  },
  "settings_sync_enable": true,
  "settings_sync_patterns": [
    "*"
  ],
  "settings_sync_exclude_patterns": [
    "index.routing.allocation.*"
  ],
  "alias_sync_enable": true,
  "state_sync_enable": true
}
```

Table 2-125 Parameters for creating a pattern-matching index synchronization task

Parameter	Mandatory	Type	Default Value	Description
pattern_name	Yes	String	N/A	The name of the index name matching pattern.
remote_cluster	Yes	String	N/A	The name of the leader cluster configuration task. The value must be consistent with the value of **leader_name** set in **Connecting the Leader and Follower Clusters**, for example, **leader1**.
remote_index_patterns	Yes	String	N/A	The leader-cluster index matching pattern. • Single index: Enter the index name, for example, **my_index**. • Wildcard: Use the wildcard (*) to match multiple indexes. For example, **myindex*** matches all indexes whose name starts with **myindex**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
local_index_pattern	Yes	String	N/A	Index naming pattern in the follower cluster after synchronization from the leader cluster. Template replacement is supported. For example, if this parameter is set to **{{remote_index}}-sync**, the index name **log1** changes to **log1-sync** after synchronization. **CAUTION** If leader/follower switchover is likely to happen, set this parameter to **{{remote_index}}** to ensure that the leader and follower clusters use identical index names.
apply_exist_index	Yes	Boolean	true	Whether to synchronize existing indexes from the leader cluster. The value can be: • **true**: Synchronize both existing and new indexes, as long as they match the pattern you define. • **false**: Synchronize new indexes only. Skipping the synchronization of existing indexes may lead to data inconsistency between the leader and follower clusters. Determine how to set this parameter based on your service needs.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
settings	No	Map	N/A	Index settings to be modified and their target values after synchronization to the follower cluster. This parameter allows you to flexibly configure index settings (such as the number of replicas and refresh interval) to match the hardware resources or service requirements in the follower cluster. If not configured, the index settings of the leader cluster will be used by default. Key and value: • **key**: the name of the setting, for example, number_of_replicas. • **value**: its new value, for example, 4. The following settings cannot be modified: number_of_shards, version.created, uuid, creation_date, and soft_deletes.enabled.
settings_sync_enable	No	Boolean	false	Whether to periodically synchronize index settings from the leader cluster. The value can be: • **true**: Synchronize index settings from the leader cluster. • **false**: Not to synchronize index settings from the leader cluster when they are updated. Instead, keep the initial settings.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
settings_sync_patterns	No	String	* (matches all settings)	Leader-cluster index settings to be synchronized to the follower cluster. Specify the names of the settings to be synchronized. ● A single setting: Enter the setting name (for example, **index.max_regex_length**). ● Multiple settings: Enter multiple setting names and separate them with a comma (,), for example, **index.max_regex_length,index.max_refresh_listeners**. ● Wildcard: Use the wildcard * to match multiple settings. For example, **index*** matches all settings whose name starts with **index**. Constraints: ● This parameter takes effect only when **settings_sync_enable** is set to **true**. ● If the same setting is configured both in **settings** and **settings_sync_patterns**, the setting in **settings_sync_patterns** takes effect. ● The following settings cannot be synchronized to the follower cluster: index.number_of_shards, index.version.created, index.uuid, index.creation_date, index.history.uuid, index.number_of_replicas

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
				, index.auto_expand_replicas, and index.version.upgraded.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
settings_sync_exclude_patterns	No	String	Null (excludes none)	Leader-cluster index settings not to be synchronized to the follower cluster. Specify the names of the settings not to be synchronized. • A single setting: Enter the setting name (for example, **index.max_regex_length**). • Multiple settings: Enter multiple setting names and separate them with a comma (,), for example, **index.max_regex_length,index.max_refresh_listeners**. • Wildcard: Use the wildcard * to match multiple settings. For example, **index*** matches all settings whose name starts with **index**. Constraints: • This parameter takes effect only when **settings_sync_enable** is set to **true**. • If both **settings_sync_patterns** and **settings_sync_exclude_patterns** are configured, only the settings that are in **settings_sync_patterns** but not in **settings_sync_exclude_patterns** will be synchronized to the follower cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
				• If the indexes you plan to synchronize include cold indexes, make sure the follower cluster contains cold data nodes. Otherwise, data synchronization will fail. The following configuration is required to synchronize cold indexes to the data nodes in the follower cluster (in such case, these indexes cease to be cold indexes in the follower cluster): `"settings_sync_exclude_patterns" : ["index.routing.allocation.require.box_type"]`
alias_sync_enable	No	Boolean	false	Whether to periodically synchronize index aliases from the leader cluster. The value can be: • **true**: Periodically synchronize index aliases from the leader cluster. • **false**: Not to periodically synchronize index aliases from the leader cluster.
state_sync_enable	No	Boolean	false	Whether to periodically synchronize index status from the leader cluster. The value can be: • **true**: Periodically synchronize index status from the leader cluster. • **false**: Not to periodically synchronize index status from the leader cluster.

2. Check the index synchronization status to confirm successful configuration.

   After synchronization is enabled, the index in the follower cluster becomes read-only. Synchronization is performed periodically to ensure data

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

consistency. The default synchronization interval is 30 seconds. For how to change it, see **Changing the Index Synchronization Interval**.

Run the following command in the follower cluster to obtain the synchronization status of a specified index:

```
GET {index_name}/sync_stats
```

An example output is as follows:

```
{
  "indices" : {
    "data1_follower" : {
      "shards" : {
        "0" : [
          {
            "primary" : false,            // Whether it is a primary shard
            "total_synced_times" : 27,    // Total synchronization times
            "total_empty_times" : 25,     // Total number of times when no data is synchronized between the leader and follower clusters because they have identical shards and data
            "total_synced_files" : 4,     // Number of synchronized files
            "total_synced_bytes" : 3580, // Total size of synchronized files
            "total_paused_nanos" : 0,   //Duration of synchronization pauses due to traffic throttling
            "total_paused_times" : 0,   //Number of synchronization pauses due to traffic throttling
            "current" : {
              "files_count" : 0, //Number of files that are being synchronized
              "finished_files_count" : 0, //Number of files that have been synchronized
              "bytes" : 0, //Size of files that are being synchronized
              "finished_bytes" : 0 //Size of files that have been synchronized
            }
          },
          {
            "primary" : true,             // Whether it is a primary shard
            "total_synced_times" : 28,    // Total synchronization times
            "total_empty_times" : 26,     // Total number of times when no data is synchronized between the leader and follower clusters because they have identical shards and data
            "total_synced_files": 20,    // Number of synchronized files
            "total_synced_bytes": 17547, // Total size of synchronized files
            "total_paused_nanos" : 0,     //Duration of synchronization pauses due to traffic throttling
            "total_paused_times" : 0,     //Number of synchronization pauses due to traffic throttling
            "current" : {
              "files_count" : 0, //Number of files that are being synchronized
              "finished_files_count" : 0, //Number of files that have been synchronized
              "bytes" : 0, //Size of files that are being synchronized
              "finished_bytes" : 0 //Size of files that have been synchronized
            }
          }
        ]
      }
    }
  }
}
```

Managing Pattern-matching Index Synchronization

You can query and delete existing pattern-matching index synchronization tasks.

- **Query** existing pattern-matching index synchronization tasks.
  - Run the following command in the follower cluster to query existing pattern-matching index synchronization tasks:
    ```
    GET auto_sync/pattern
    ```
  - Run the following command in the follower cluster to query details about a specified pattern-matching index synchronization task:
    ```
    GET auto_sync/pattern/{pattern_name}
    ```

  Example response:

  ```
  {
    "patterns" : [
  ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  {
    "name" : "pattern1",
    "pattern" : {
      "remote_cluster" : "leader",
      "remote_index_patterns" : [
        "log*"
      ],
      "local_index_pattern" : "{{remote_index}}-sync",
      "settings" : { }
    }
  }
 ]
}
```

- **Delete** an existing pattern-matching index synchronization task.

  CAUTION

  Deleting a pattern-matching index synchronization task only prevents new synchronization tasks from being created for matched indexes. It does not stop synchronization tasks that have already been created. To stop these tasks, manually perform **Stopping Index Synchronization**.

  a. Run the following command in the follower cluster to delete a specified pattern-matching index synchronization task:
  ```
  DELETE auto_sync/pattern/{pattern_name}
  ```
  b. Run the following command to query the list of pattern-matching index synchronization tasks to confirm successful deletion.
  ```
  GET auto_sync/pattern
  ```

Changing the Index Synchronization Interval

The default index synchronization interval between the leader and follower clusters is 30 seconds. To change it for a specified index, run the following command:

```
PUT {index_name}/_settings
{
  "index.remote_sync.sync_interval": "2s"
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-126 Parameters for modifying the index synchronization interval

Parameter	Type	Default Value	Description
index_name	String	N/A	Specifies one or more indexes. • Single index: Enter the index name, for example, **my_index**. • Multiple indexes: Enter multiple index names and use a comma (,) to separate them, for example, **my_index1,my_index2**. • Wildcard: Use the wildcard (*) to match multiple indexes. For example, **myindex*** indicates all indexes whose name starts with **myindex**.
index.remote_sync.sync_interval	String	30s (recommended)	The index synchronization interval. Value format: number + unit • Number: a natural number • Unit: nanos (nanosecond), micros (microsecond), ms (millisecond), s (second), m (minute), h (hour), or d (day) Minimum value: **1s** You may reduce this interval if the timeliness of data is prioritized, but doing so will increase the cluster's CPU load.

Stopping Index Synchronization

Run the following command in the follower cluster to stop a single-index synchronization task immediately:

```
PUT {index_name}/stop_remote_sync
```

After this command is executed:

- Any subsequent changes made to the index in the leader cluster will no longer be synchronized to the follower cluster.
- The index in the follower cluster will be removed from read-only mode, allowing new data to be written to it.

CAUTION

Index synchronization cannot be stopped when a snapshot is being created for the index. To stop synchronization, you must wait until snapshot creation is complete.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Advanced Settings

In addition to basic synchronization settings, read/write splitting allows you to adjust synchronization behavior through advanced cluster-level settings, including forced synchronization, index deletion synchronization, and synchronization rate control.

Run the following command in the follower cluster to configure advanced settings:

```
PUT _cluster/settings
{
  "persistent": {
    "remote_sync.force_synchronize": true,
    "remote_sync.auto_delete_index": true,
    "remote_sync.chunk_size": "2MB",
    "remote_sync.max_concurrent_file_chunks": 20,
    "remote_sync.max_bytes_per_sec": "100MB"
  }
}
```

Table 2-127 Advanced settings

Parameter	Type	Default Value	Description
remote_sync.force_synchronize	Boolean	false	Whether to force index synchronization. • **true**: Forces synchronization. In each synchronization cycle, the index metadata of the leader cluster is forcibly synchronized to the follower cluster, even if the number of documents has not changed. • **false**: Does not force synchronization. Index metadata is synchronized only when the number of documents in the leader cluster changes.
remote_sync.auto_delete_index	Boolean	false	Whether to synchronize index deletion to the follower cluster. • **true**: When an index is deleted from the leader cluster, the corresponding index is automatically deleted in the follower cluster. • **false**: When an index is deleted from the leader cluster, the corresponding index is retained in the follower cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
remote_sync.chunk_size	String	1MB	The size of a single chunk for index synchronization. Value format: number + unit ● Number: a natural number ● Unit: B, K, KB, M, MB, G, GB, T, TB, P, or PB (case-insensitive) We recommend setting this value to at least 1 MB. Setting a value smaller than 1 MB may degrade system performance.
remote_sync.max_concurrent_file_chunks	Integer	10	Maximum number of document chunks synchronized concurrently on a single node. Value range: 1 to 100. Set this parameter based on the node bandwidth and CPU resources available.
remote_sync.max_bytes_per_sec	String	40MB	Maximum size of data transmitted per second on a single node during synchronization. You can set this parameter to prevent data synchronization tasks from exhausting the network bandwidth. Value format: number + unit ● Number: a natural number ● Unit: B, K, KB, M, MB, G, GB, T, TB, P, or PB (case-insensitive) The value **0** indicates no limit. When your cluster's query traffic is low, you can increase the value, but doing so may increase the network load.

Case: Leader/Follower Switchover

When the leader cluster becomes faulty, you can perform a leader/follower switchover to have the follower cluster take over to ensure service continuity. To allow this to happen seamlessly and ensure data consistency, use identical index names between the leader and follower clusters when configuring index synchronization.

1. Determine the index synchronization mode between the leader and follower clusters.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Run the following command in the follower cluster to check for pattern-matching index synchronization tasks:

```
GET auto_sync/pattern
```

- If there are no such tasks, synchronization is performed for specified indexes between the leader and follower clusters. In this case, go to **3**.
- If there are such tasks, index synchronization between the leader and follower clusters is based on an index matching pattern. In this case, perform the next step.

2. Run the following command in the follower cluster to delete a specified pattern-matching index synchronization task:

```
DELETE auto_sync/pattern/{pattern_name}
```

3. Stop index synchronization in the follower cluster, and switch read and write traffic to it. Make sure the follower cluster can handle requests independently.

   Run the following command in the follower cluster to stop single-index synchronization tasks for specified indexes:

```
PUT */stop_remote_sync
```

4. After the leader cluster recovers, configure information about the follower cluster in the leader cluster, and connect the leader and follower clusters again. For details, see **Connecting the Leader and Follower Clusters**.
5. Under the leader cluster, perform **Configuring Index Synchronization** to synchronize data from the follower cluster to the leader cluster, and then perform a leader/follower switchover to switch back.

2.11.5 Configuring Flow Control 2.0

Configure flow control policies for your Elasticsearch cluster in both the inbound and outbound directions, ensuring cluster stability by safeguarding against abnormal traffic.

An Elasticsearch cluster can become overloaded due to traffic surges, malicious requests, and internal resource competition, which can even lead to node failures. Through policies like client request throttling, backpressure, and traffic pattern analysis, flow control ensures proper resource allocation, thereby protecting clusters from overload. It covers the following scenarios:

- High-concurrency write handling: mitigates the risk of out-of-memory (OOM) exceptions under heavy write loads.
- Security defense: controls access by IP address using both blacklists and whitelists.
- Emergency response: blocks malicious or abnormal traffic in one click.
- Performance optimization: optimizes flow control thresholds and policies based on collected statistics.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

How the Feature Works

Table 2-128 Flow control policies

Policy	How It Works	Details
HTTP/HTTPS flow control	Controls client access traffic using blacklists and whitelists, an upper limit on concurrent connections, and a rate limit on new connection attempts. • Blacklists and whitelists: A whitelist takes precedence over a blacklist. If an IP address is on both lists, it is allowed. Requests sent through blacklisted connections will be ignored. • Concurrent connection limit: Limits the total number of concurrent HTTP connections to prevent overload. • New connection limit: Limits the number of new connections that can be set up per second. The warmup_period parameter protects against connection floods, allowing traffic to grow gradually and steadily. When HTTP/HTTPS flow control is enabled, requests from blacklisted IP addresses are always rejected; for IP addresses on the whitelist, no flow control rules apply; for other IP addresses, when either the concurrent connection limit or the new connection limit is reached, requests from them will be rejected.	**Configuring HTTP/HTTPS Flow Control**
Memory-based flow control	When the heap memory usage exceeds a pre-defined threshold (for example, 80%), the system stops receiving large requests, and garbage collection (GC) is triggered to reclaim memory. Write traffic is throttled by setting the backpressure factor (in_flight_factor) and the maximum delay for request handling (max). When memory-based flow control is enabled, large requests may be delayed for a long time when the cluster's heap memory usage exceeds the configured threshold.	**Configuring Memory-based Flow Control**

Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Policy	How It Works	Details
One-click traffic blocking	When triggered, the system immediately disconnects all client connections that are not whitelisted, with the exception of those used for Kibana access or O&M and monitoring APIs, as an effort to restore the cluster.	**Configuring One-Click Traffic Blocking**
Request statistics sampling and analysis	Records request metrics (such as bulk writes and queries) by client IP address, and exposes them via a statistics API to evaluate the cluster load and proactively identify abnormal traffic patterns.	**Configuring Request Statistics Sampling and Analysis**
Access logging	Records the URLs and bodies of HTTP/HTTPS requests for cluster load and client request analysis. Access logs can also be saved to files (that is, persisted to disk) to facilitate troubleshooting and performance analysis. Enabling access logging incurs extra CPU and memory overhead, which may slow down request handling.	• **Configuring Access Logging** • **Configuring Access Logging in Files**

Constraints

Elasticsearch 7.6.2 and 7.10.2 clusters created after February 2023 support Flow Control 2.0 only, while clusters created before that support Flow Control 1.0 only.

Logging In to Kibana

Log in to Kibana and go to the command execution page. Elasticsearch clusters support multiple access methods. This topic uses Kibana as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Configuring HTTP/HTTPS Flow Control

Control client access traffic using blacklists and whitelists, an upper limit on concurrent connections, and a rate limit on new connection attempts, to prevent overload.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

1. Enable HTTP/HTTPS flow control.

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.http.enabled": true,
    "flowcontrol.http.allow": ["192.168.0.1/24", "192.168.2.1/24"],
    "flowcontrol.http.deny": "192.168.1.1/24",
    "flowcontrol.http.concurrent": 1000,
    "flowcontrol.http.newconnect": 1000,
    "flowcontrol.http.warmup_period": 0
  }
}
```

Table 2-129 Parameters for configuring HTTP/HTTPS flow control

Parameter	Type	Default Value	Description
flowcontrol.http.enabled	Boolean	false	Enable or disable HTTP/HTTPS flow control. When enabled, flow control will be performed based on relevant settings. The value can be: ● true: Enable HTTP/HTTPS flow control. ● false: Disable HTTP/HTTPS flow control.
flowcontrol.http.allow	List<String>	Null (no whitelist)	A whitelist of client IP addresses or CIDR blocks that are allowed to access the cluster, supporting: ● Individual IP addresses, for example, **192.18.0.1**. ● CIDR blocks, for example, **192.168.0.0/24**. ● Multiple IP addresses or CIDR blocks separated by commas (,), for example, **192.168.0.1/24, 192.168.2.1/24**. Setting this parameter to **null** restores the default value.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
flowcontrol.http.deny	List<String>	Null (no blacklist)	A blacklist of client IP addresses or CIDR blocks that are not allowed to access the cluster. A whitelist takes precedence over a blacklist. A blacklist supports the following: • Individual IP addresses, for example, **192.18.0.1**. • CIDR blocks, for example, **192.168.0.0/24**. • Multiple IP addresses or CIDR blocks separated by commas (,), for example, **192.168.0.1/24, 192.168.2.1/24**. Setting this parameter to **null** restores the default value.
flowcontrol.http.concurrent	Integer	Node vCPUs x 600	Maximum number of concurrent HTTP/HTTPS connections that can be handled by a node. Minimum value: **10** Setting this parameter to **null** restores the default value.
flowcontrol.http.newconnect	Integer	Node vCPUs x 200	Maximum number of new HTTP/HTTPS connections that can be created per second per node. Minimum value: **10** Setting this parameter to **null** restores the default value.
flowcontrol.http.warmup_period	Integer	0 (no grace period before reaching the full capacity)	A grace period during which a system gradually ramps up from accepting zero HTTP/HTTPS requests to its full, maximum capacity. Value range: 0–10000 Unit: ms For example, if **flowcontrol.http.newconnect** is set to **100** and **flowcontrol.http.warmup_period** is set to **5000ms**, it takes 5 seconds for the system to reach 100 new connections per second. Setting this parameter to **null** restores the default value.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. Disable HTTP/HTTPS flow control.

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.http.enabled": false
  }
}
```

Configuring Memory-based Flow Control

Enable write throttling to mitigate the risk of OOM exceptions when the heap memory usage of a node exceeds a predefined threshold.

1. Enable memory-based flow control.

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.memory.enabled": true,
    "flowcontrol.memory.heap_limit": "80%"
  }
}
```

Table 2-130 Memory-based flow control parameters

Parameter	Type	Default Value	Description
flowcontrol.memory.enabled	Boolean	true	Whether to enable memory-based flow control. When enabled, node heap memory usage is monitored and a threshold is set, and writes are throttled when this threshold is reached. The value can be: • true: Enable memory-based flow control. • false: Disable memory-based flow control.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
flowcontrol.memory.heap_limit	String	90% (conservative threshold)	Node heap memory usage threshold. When this threshold is exceeded, a backpressure mechanism is triggered. Value range: 10%–100% • When the heap memory usage exceeds this threshold, the system stops processing client requests that are larger than 64 KB. Processing resumes only when the heap memory usage drops below this threshold. • When the heap memory usage is five percentage points below this threshold, the system continues processing requests. However, it restricts the total data read to 5% of the total heap memory capacity per cycle. This limit, which is configured using the **flowcontrol.memory.once_free_max** parameter, creates a memory buffer to prevent immediate exhaustion when processing resumes. • While the heap memory usage stays above this threshold, the system cannot accept new client requests. If the **flowcontrol.memory.nudges_gc** parameter is set to **true**, the system will actively trigger garbage collection (GC) and repeatedly attempt to reclaim memory until usage drops below this threshold. This helps prevent system breakdown caused by potential memory leaks. In practice, you are advised to set this parameter to 80% or lower to reserve heap memory for non-read tasks, for example, segment merge. Setting this parameter to **null** restores the default value.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
flowcontrol.holding.in_flight_factor	Float	1.0 (recommended)	Backpressure factor, which controls the sensitivity of memory-based backpressure. A larger value indicates more powerful write throttling. Value range: ≥ 0.5 This parameter estimates the potential heap memory impact of an incoming large request. The calculation is as follows: **in_flight_factor x Request body size**. The resulting estimate is then used to apply memory-based backpressure and throttling. Setting this parameter to **null** restores the default value.
flowcontrol.holding.max	TimeValue	60s	Maximum request handling delay allowed before requests are handled according to the policy defined by **flowcontrol.holding.max_strategy**. Value range: ≥ 15s Unit: second Generally, you should configure this parameter based on the **flowcontrol.holding.max_strategy** setting. ● When **flowcontrol.holding.max_strategy** is set to **soft**, keep the value of this parameter lower than the client request timeout. Additionally, reserve some request execution time. ● When **flowcontrol.holding.max_strategy** is set to **hard**, keep the value of this parameter higher than the client request timeout. ● When **flowcontrol.holding.max_strategy** is set to **keep**, this parameter is invalid. Setting this parameter to **null** restores the default value.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
flowcontrol.holding.max_strategy	String	keep	Handling policy or action taken for requests delayed longer than **flowcontrol.holding.max**. The value can be: • **keep**: Maintain the backpressure state and wait for the heap memory usage to drop. The server determines whether to release requests based on real-time memory usage. In this mode, requests will be delayed until the memory usage drops to a level that allows the processing to resume. This may cause request timeout. • **soft**: Forcibly execute the requests, but the **inFlight** circuit breaker gets to decide whether to reject them. **inFlight** is a native Elasticsearch circuit breaker designed to prevent system overload. For details, see **Circuit breaker settings**. This mode allows requests that have been delayed longer than **flowcontrol.holding.max** to proceed. However, it may still cause memory usage to spike and eventually lead to memory overflow. • **hard**: Reject the requests immediately and disconnect client connections. This will drop some requests. Setting this parameter to **null** restores the default value.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
flowcontrol.memory.once_free_max	String	5%	Maximum amount of memory (in the form of a percentage of node memory) that flow control is allowed to free in a single reclamation cycle. This parameter prevents overly aggressive memory reclamation that could lead to sudden request surges after memory pressure subsides. Value range: 1%–50% Setting this parameter to **null** restores the default value.
flowcontrol.memory.nudges_gc	Boolean	true (recommended)	Whether to trigger garbage collection (GC) to reclaim memory when the write pressure is too high. (The backpressure connection pool is checked every second. The write pressure is considered high if all existing connections are blocked and new write requests cannot be accepted.) The value can be: • true: Trigger GC. • false: Not to trigger GC. Setting this parameter to **null** restores the default value.

2. Disable memory-based flow control.

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.memory.enabled": false
  }
}
```

Configuring One-Click Traffic Blocking

When triggered, the system immediately disconnects all client connections, but not those used for Kibana access or O&M and monitoring APIs, as an effort to restore the cluster.

1. Enable one-click traffic blocking.

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.break.enabled": true
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-131 Parameters for configuring one-click traffic blocking

Parameter	Type	Default Value	Description
flowcontrol.break.enabled	Boolean	false	Whether to enable one-click traffic blocking (similar to a circuit breaker). When enabled, the system immediately disconnects all client connections, but not those used for Kibana access or O&M and monitoring APIs. The value can be: • true: Enable one-click blocking. • false: Disable one-click blocking.

2. Disable one-click traffic blocking.

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.break.enabled": false
  }
}
```

Configuring Request Statistics Sampling and Analysis

Collect request metrics by client IP address to help identify abnormal traffic patterns.

1. Enable request statistics sampling.

```
PUT _cluster/settings
{
  "transient": {
    "flowcontrol.log.access.enabled": true
  }
}
```

Table 2-132 Parameters for request statistics sampling

Parameter	Type	Default Value	Description
flowcontrol.log.access.enabled	Boolean	false	Whether to enable request statistics sampling, that is, whether to collect request metrics (such as bulk writes and search/msearch requests) by client IP address. The value can be: • true: Enable request statistics sampling. • false (default): Disable request statistics sampling.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
flowcontrol.log.access.count	Integer	10	Maximum number of client IP addresses sampled. Value range: 0–100 Setting this parameter to **null** restores the default value.

2. Check the sampled statistics to analyze the traffic pattern and flow control status by client IP address.
   - Check the flow control status of all nodes.
     ```
     GET /_nodes/stats/filter/v2
     ```
   - Check the flow control details of all nodes.
     ```
     GET /_nodes/stats/filter/v2?detail
     ```
   - Check the flow control status of a specified node.
     ```
     GET /_nodes/{node_id}/stats/filter/v2
     ```

     Table 2-133 Parameter description

Parameter	Type	Default Value	Description
node_id	String	N/A	Specifies one or more cluster nodes. • Single node: Enter the node ID. • Multiple nodes: Enter multiple node IDs and use a comma (,) to separate them. You can run the following command to obtain node IDs: `GET _cat/nodes?s=n&h=n,id&v=true&full_id=true`

Example response:

```
{
  "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  },
  "cluster_name" : "css-xxxx",
  "nodes" : {
    "d3qnVIpPTtSoadkV0LQEkA" : {
      "name" : "css-xxxx-ess-esn-1-1",
      "host" : "192.168.x.x",
      "timestamp" : 1672236425112,
      "flow_control" : {
        "http" : {
          "current_connect" : 52,
          "rejected_concurrent" : 0,
          "rejected_rate" : 0,
          "rejected_black" : 0,
          "rejected_breaker" : 0
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
      },
      "access_items" : [
        {
          "remote_address" : "10.0.0.x",
          "search_count" : 0,
          "bulk_count" : 0,
          "other_count" : 4
        }
      ],
      "holding_requests" : 0
    }
  }
 }
}
```

Table 2-134 Response parameters

Parameter	Description
current_connect	Number of HTTP connections to a node, which is recorded regardless of whether flow control is enabled. This value is equivalent to the **current_open** value of **GET /_nodes/stats/http** API. It shows the current client connections of each node.
rejected_concurrent	Number of concurrent connections rejected during flow control. This metric is available only when **flowcontrol.http.enabled** is set to **true**. The count will not be cleared when flow control is disabled.
rejected_rate	Number of new connections rejected during flow control. This metric is available only when **flowcontrol.http.enabled** is set to **true**. The count will not be cleared when flow control is disabled.
rejected_black	Number of new connections rejected by a preconfigured blacklist during flow control. This metric is available only when **flowcontrol.http.enabled** is set to **true**. The count will not be cleared when flow control is disabled.
rejected_breaker	Number of new connections rejected during one-click traffic blocking. This metric is available only when **flowcontrol.break.enabled** is set to **true**. The count will not be cleared when one-click traffic blocking is disabled.
access_items	IP addresses of clients that recently accessed the cluster. The number of client IP addresses sampled is determined by **flowcontrol.log.access.count**.
remote_address	IP addresses and the number of requests.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
search_count	Number of times a client accessed a database using **_search** and **_msearch**.
bulk_count	Number of times a client accessed a database using **_bulk**.
other_count	Number of times a client accessed a database using other request methods.
holding_requests	Number of connections to the current node where writes are halted due to flow control.

3. Disable request statistics sampling.

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.log.access.enabled": false
  }
}
```

Configuring Access Logging

When access logging is enabled, the system records the URLs and bodies of HTTP/HTTPS requests for cluster load and request analysis. Then, you can use the result to optimize cluster performance.

1. Enable access logging.
   - Enable access logging for all nodes in a cluster.
     ```
     PUT /_access_log?duration_limit=30s&capacity_limit=1mb
     ```
   - Enable access logging for a specified node in a cluster.
     ```
     PUT /_access_log/{node_id}?duration_limit=30s&capacity_limit=1mb
     ```

Table 2-135 Parameters for enabling access logging

Parameter	Type	Default Value	Description
duration_limit	String	30	Maximum duration for access logging. When this limit is reached, access logging stops. Value range: 10 to 120 Unit: s Setting this parameter to **null** restores the default value. Access logging stops when either **duration_limit** or **capacity_limit** is reached.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
capacity_limit	String	1	Maximum access log size. When the size of an access log reaches this limit, access logging stops. Value range: 1 to 5 Unit: MB Setting this parameter to **null** restores the default value. Access logging stops when either **duration_limit** or **capacity_limit** is reached.

2. Check access logs.
   - Check the access logs of all nodes in a cluster.
     ```
     GET /_access_log
     ```
   - Check the access logs of a specified cluster node.
     ```
     GET /_access_log/{node_id}
     ```

   Example response:

```
{
  "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  },
  "cluster_name" : "css-flowcontroller",
  "nodes" : {
    "8x-ZHu-wTemBQwpcGivFKg" : {
      "name" : "css-flowcontroller-ess-esn-1-1",
      "host" : "10.0.0.98",
      "count" : 2,
      "access" : [
        {
          "time" : "2021-02-23 02:09:50",
          "remote_address" : "/10.0.0.98:28191",
          "url" : "/_access/security/log?pretty",
          "method" : "GET",
          "content" : ""
        },
        {
          "time" : "2021-02-23 02:09:52",
          "remote_address" : "/10.0.0.98:28193",
          "url" : "/_access/security/log?pretty",
          "method" : "GET",
          "content" : ""
        }
      ]
    }
  }
}
```

Table 2-136 Response parameters

Parameter	Description
name	Node name

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
host	Node IP address
count	Number of node access requests in a statistical period
access	Details about node access requests in a statistical period
time	Request time
remote_address	Source IP address and port number in the request
url	Original URL of the request
method	Request method
content	Request content. If the value is an empty string (""), there is no request body.

3. Delete access logs. Logs are stored in the memory. After viewing the logs, you should delete them promptly to reclaim memory resources, and by doing so avoid impacting system performance.
   a. Delete access logs for all nodes.
      ```
      DELETE /_access_log
      ```
   b. Check the access logs again to confirm successful deletion.
      ```
      GET /_access_log
      ```

Configuring Access Logging in Files

Access logs can be persisted to disk for troubleshooting and analysis. Use this function sparingly, as it can impact cluster performance. Remember to disable it immediately after resolving the issue.

1. Enable access logging in files.
   ```
   PUT /_cluster/settings
   {
     "persistent": {
       "flowcontrol.log.file.enabled": true
     }
   }
   ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-137 Enabling access logging in files

Parameter	Type	Default Value	Description
flowcontrol.log.file.enabled	Boolean	false	Whether to record access logs in files. When enabled, the log of each access request is recorded in files. The log file name is **Cluster name_access_log.log**. You can check this file only through the log backup function. For details about how to back up logs, see **Enabling Log Backup**. The value can be: • true: Record access logs in files. • false: Not to record access logs in files.

2. Disable access logging in files.

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.log.file.enabled": false
  }
}
```

2.11.6 Configuring Flow Control 1.0

Configure flow control policies for your Elasticsearch cluster in both the inbound and outbound directions, ensuring cluster stability by safeguarding against abnormal traffic.

An Elasticsearch cluster can become overloaded due to traffic surges, malicious requests, and internal resource competition, which can even lead to node failures. Through policies like client request throttling, write backpressure, and traffic pattern analysis, flow control ensures proper resource allocation, thereby protecting clusters from overload. It covers the following scenarios:

- High-concurrency write handling: mitigates the risk of out-of-memory (OOM) exceptions under heavy write loads.
- Security defense: controls access by IP address using both blacklists and whitelists.
- Emergency response: blocks malicious or abnormal traffic in one click.
- Performance optimization: optimizes flow control thresholds and policies based on collected statistics.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Function Description

Table 2-138 Flow control policies

Policy	How It Works	Details
HTTP/HTTPS flow control	• You can control cluster access by client IP address or subnet through the HTTP/HTTPS blacklist or whitelist. If an IP address is in the blacklist, the client is disconnected right away and all its requests are rejected. The whitelist takes precedence over the blacklist. If a client IP address is on both the blacklist and whitelist, requests from it will not be rejected. • Flow control based on concurrent HTTP/HTTPS connections: Limits the total number of concurrent HTTP/HTTPS connections a node can handle. • Flow control based on new HTTP/HTTPS connections: Limits the number of new connections that can be established to a node per second.	**Enabling HTTP/HTTPS Flow Control per Node**
Memory flow control	Memory flow control limits request paths based on heap memory usage. You can configure a whitelist for memory flow control, a global memory usage threshold, and per-path heap memory thresholds. The global memory flow control threshold takes precedence over the memory threshold of a single path. Paths on the whitelist are exempt from memory flow control.	**Enabling Memory Flow Control**
Global path whitelist for flow control	You can configure the global path whitelist for flow control as required when you need to use custom plug-ins.	**Adding a Global Path Whitelist for Flow Control**
Request sampling	Request sampling can record the number of access requests from client IP addresses and the request paths of sampled users. Based on the statistics, you can identify and analyze access traffic by client IP addresses and request paths.	**Enabling Request Sampling**
One-click traffic blocking	One-click traffic blocking blocks all client connections to a node. This, however, does not include connections for Kibana access or Elasticsearch monitor APIs.	**Enable One-Click Traffic Blocking**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Policy	How It Works	Details
Flow control	Flow control provides an independent API for checking traffic statistics and records the number of API calls. You can evaluate the flow control threshold and analyze the cluster load based on these statistics.	**Viewing Flow Control Information**
Access logging	Access logs record the URLs and bodies of HTTP/HTTPS requests received by nodes within a period of time. You can analyze the current traffic load based on the access logs.	**Enabling and Viewing Access Logs**
Access logging in files	Any cluster access is recorded in the **{*Cluster name*_access_log.log}** file. You can use the log backup function to view detailed access logs on OBS.	**Enabling Access Logging in Files**
CPU flow control	You can configure the node CPU usage threshold to limit inbound traffic for each node.	**Enabling CPU Flow Control**

Constraints

- Elasticsearch 7.6.2 and 7.10.2 clusters created after February 2023 support Flow Control 2.0 only, while clusters created before that support Flow Control 1.0 only.
- Flow control may hurt the performance of some nodes.
- If flow control is enabled, user requests that exceed the flow control threshold will be rejected.
- Enabling memory flow control may hurt the performance of some search requests or cause some Kibana search requests to fail.
- Enabling access logging may hurt cluster performance.
- Memory flow control and CPU flow control are based on request paths. Avoid configuring too many paths or paths that are too long, as they may hurt cluster performance.

Enabling HTTP/HTTPS Flow Control per Node

1. Run the following command to enable HTTP/HTTPS flow control for cluster nodes:

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.http.enabled": true,
    "flowcontrol.http.allow": ["192.168.0.1/24", "192.168.2.1/24"],
    "flowcontrol.http.deny": "192.168.1.1/24",
    "flowcontrol.http.concurrent": 1000,
    "flowcontrol.http.newconnect": 1000,
    "flowcontrol.http.warmup_period": 0
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-139 Configuration items for HTTP/HTTPS flow control

Configuration Item	Type	Description
flowcontrol.http.enabled	Boolean	Whether to enable HTTP/HTTPS flow control. HTTP/HTTPS flow control is disabled by default. Enabling it may affect node access performance. Value: **true** or **false** Default value: **false**
flowcontrol.http.allow	List<String>	IP address whitelist. It can contain multiple IP addresses and subnet masks, or lists of IP addresses. Use commas (,) to separate different items. Example: **xx.xx.xx.xx/24,xx.xx.xx.xx/24**, or **xx.xx.xx.xx,xx.xx.xx.xx**. The default value is null.
flowcontrol.http.deny	List<String>	IP address blacklist. It can contain multiple IP addresses and subnet masks, or lists of IP addresses. Use commas (,) to separate different items. The default value is null.
flowcontrol.http.concurrent	Integer	Maximum concurrent HTTP/HTTPS connections. Default value: Number of vCPUs on the node x 400
flowcontrol.http.newconnect	Integer	Maximum new HTTP/HTTPS connections that can be created per second. Default value: Number of vCPUs on the node x 200

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuration Item	Type	Description
flowcontrol.http.warmup_period	Integer	Time required for the HTTP/HTTPS connection setup speed to reach the maximum. If **flowcontrol.http.newconnect** is set to **100** and **flowcontrol.http.warmup_period** is set to **5000ms**, it indicates the system can create up to 100 connections per second 5 seconds later. Value range: 0–10000 Unit: ms Default value: **0**

NOTE

If all parameters are set to **null**, they will be restored to their default values.

2. Run the following command to disable HTTP/HTTPS flow control for cluster nodes:

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.http.enabled": false
  }
}
```

Enabling Memory Flow Control

1. Enable memory flow control.
   - Run the following command to enable memory flow control:

     ```
     PUT /_cluster/settings
     {
       "persistent": {
         "flowcontrol.memory.enabled": true,
         "flowcontrol.memory.allow_path": "/index/_search",
         "flowcontrol.memory.heap_limit": "85%"
       }
     }
     ```

   - Run the following command to enable memory flow control for a single request path.

     Configure the heap memory usage threshold for a request path. You can configure priorities for such threshold rules.

     ```
     PUT /_cluster/settings
     {
       "persistent": {
         "flowcontrol.memory.enabled": true,
         "flowcontrol.memory": {
           "flowcontrol_search": {
             "filter_path": "index1/_search",
             "heap_limit": "50%"
           },
           "flowcontrol_bulk": {
             "filter_path": "index*/_bulk",
     ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
      "heap_limit": "50%"
    }
   }
  }
}
```

Table 2-140 Configuration items for memory flow control

Configuration Item	Type	Description
flowcontrol.memory.enabled	Boolean	Whether to enable memory flow control. This function is disabled by default. Enabling memory flow control may slightly affect node performance. Value: **true** or **false** Default value: **false**
flowcontrol.memory.allow_path	List<String>	Request path whitelist for memory flow control. Whitelisted paths are exempt from memory flow control. Wildcard characters are supported. By default, query APIs controlled by the cluster are exempt from memory flow control. This prevents the failure to query cluster information when the memory usage reaches the threshold. Example: ● "flowcontrol.memory.allow_path": "/index/_search", ● "flowcontrol.memory.allow_path": "/index*/_search", ● "flowcontrol.memory.allow_path": ["/index/_search", "/index1/_bulk"], A maximum of 10 paths can be configured. A path can contain up to 32 characters. The default value is null.
flowcontrol.memory.heap_limit	String	Maximum global heap memory usage of a node before flow control is triggered. The value cannot be less than 10% of the heap memory. Value range: 10%–100% Default value: **90%**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuration Item	Type	Description
flowcontrol.memory.*.filter_path	String	Paths under memory flow control. The default value is ******, indicating all paths. If **flowcontrol.memory.heap_limit** is configured and **flowcontrol.memory.*.filter_path** is not, it indicates that all the paths, except those in the whitelist, are under control. The whitelist takes precedence over the single-path rule. If a path is specified in both **flowcontrol.memory.allow_path** and **flowcontrol.memory.*.filter_path**, the requests from the path will be allowed. For example, if **flowcontrol.memory.allow_path** and **flowcontrol.memory.*.filter_path** are both set to **abc/_search**, then **abc/_search** will not be under flow control. Maximum length: **32** characters
flowcontrol.memory.*.heap_limit	String	Heap memory usage threshold of request paths. If the heap memory usage exceeds the threshold, flow control will be triggered. Value range: 10%–100% Default value: **90%**

NOTE

If all parameters are set to **null**, they will be restored to their default values.

2. Run the following command to disable memory flow control:
   - Run the following command to delete memory flow control for a single request path.

     Before disabling memory flow control, you need to delete per-request path memory flow control configuration.

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.memory.enabled": true,
    "flowcontrol.memory": {
      "flowcontrol_search": {
        "filter_path": null,
        "heap_limit": null
      }
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  }
 }
}
```

- Run the following command to disable memory flow control:

```
PUT /_cluster/settings
{
 "persistent": {
  "flowcontrol.memory.enabled": false
 }
}
```

Adding a Global Path Whitelist for Flow Control

Run the following command to add a global path whitelist for flow control:

```
PUT _cluster/settings
{
 "persistent": {
  "flowcontrol.path.white_list": "xxxx"
 }
}
```

Table 2-141 Configuration items for a global path whitelist for flow control

Configuration Item	Type	Description
flowcontrol.path.white_list	List<String>	Paths that are exempt from flow control. These paths are not affected by memory flow control, CPU flow control, or one-click blocking; but are under IP address-based flow control. A maximum of 10 paths can be configured. A path can contain up to 32 characters. The default value is null. **NOTE** You are advised not to configure this parameter, unless required by plug-ins.

NOTE

If all parameters are set to **null**, they will be restored to their default values.

Enabling Request Sampling

1. Run the following command to enable request sampling:

```
PUT /_cluster/settings
{
 "persistent": {
  "flowcontrol.statics.enabled": true,
  "flowcontrol.statics.threshold": 100,
  "flowcontrol.statics.sample_frequency": 50
 }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-142 Configuration items for request sampling

Configuration Item	Type	Description
flowcontrol.statics.enabled	Boolean	Whether to enable request sampling. Request sampling may affect node performance. Value: **true** or **false** Default value: **false**
flowcontrol.statics.threshold	Integer	Number of recent access requests whose statistics are collected. The value **100** indicates that statistics will be collected on the 100 IP addresses and 100 URLs that are most frequently accessed. Minimum value: **10** Maximum value: **1000** Default value: **100** **NOTE** • The IP address statistics and URL sampling statistics are cached based on their access time. If the number of cached records reaches the threshold configured using **flowcontrol.statics.threshold**, the earliest records will be deleted. • In URL sampling, an access path is uniquely identified by its URL hash.
flowcontrol.statics.sample_frequency	Integer	Path sampling frequency. If this parameter is set to **100**, samples are collected from every 100 requests. Minimum value: **50** Default value: **100**

NOTE

If all parameters are set to **null**, they will be restored to their default values.

2. Run the following command to disable request sampling:

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.statics.enabled": false
  }
}
```

Enable One-Click Traffic Blocking

1. Run the following command to enable one-click traffic blocking:

```
PUT /_cluster/settings
{
  "persistent": {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
    "flowcontrol.break.enabled": true
  }
}
```

2. Run the following command to disable one-click traffic blocking:

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.break.enabled": false
  }
}
```

Viewing Flow Control Information

- Check the flow control status of all nodes.

```
GET /_nodes/stats/filter
```

- View the flow control status of a specific node.

```
GET /_nodes/{nodeId}/stats/filter
```

  {nodeId} indicates the ID of the node you want to check.

Example response:

NOTE

In the response, the information of each node is separated. The **http** field records the numbers of concurrent connections and new connections. The **memory** records memory flow control statistics. The **ip_address** field records the recent client IP addresses that are accessed most recently. The **url_sample** field records the recent URLs that are requested most frequently. The **cpu** field records CPU flow control statistics.

```
{
  "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  },
  "cluster_name" : "css-flowcontroller",
  "nodes" : {
    "ElBRNCMbTj6L1C-Wke-Dnw" : {
      "name" : "css-flowcontroller-ess-esn-1-1",
      "host" : "10.0.0.133",
      "timestamp" : 1613979513747,
      "flow_control" : {
        "transport" : {
          "concurrent_req" : 0,
          "rejected_concurrent" : 0,
          "rejected_new" : 0,
          "rejected_deny" : 0
        },
        "http" : {
          "concurrent_req" : 0,
          "rejected_concurrent" : 0,
          "rejected_new" : 0,
          "rejected_deny" : 0
        },
        "memory" : {
          "memory_allow" : 41,
          "memory_rejected" : 0
        },
        "cpu": {
          "rejected_cpu" : 0
        }
        "ip_address" : [
          {
            "ip" : "/10.0.0.198",
            "count" : 453
          },
          {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
          "ip" : "/198.19.49.1",
          "count" : 42
        }
      ],
      "url_sample" : [
        {
          "url" : "/*/_search?pretty=true",
          "method" : "GET",
          "remote_address" : "/10.0.0.198:16763",
          "count" : 1
        }
      ]
    }
  }
}
```

Table 2-143 Response parameters

Parameter	Description
concurrent_req	Number of TCP connections of a node, which is recorded no matter whether flow control is enabled. This value is similar to the value of **current_open** of the **GET /_nodes/stats/http** API but is smaller, because whitelisted IP addresses and internal node IP addresses are not counted.
rejected_concurrent	Number of concurrent connections rejected during HTTP flow control. Disabling HTTP flow control does not clear this record.
rejected_new	Number of new connections rejected during HTTP flow control. Disabling HTTP flow control does not clear this record.
rejected_deny	Number of requests rejected based on the blacklist during HTTP flow control. Disabling HTTP flow control does not clear this record.
memory_allow	Number of allowed requests during memory flow control. This parameter takes effect when memory flow control is enabled, and its value is not cleared after memory flow control is disabled. The requests from the paths in the **allow_path** whitelist are not recorded. If **allow_path** is set to ******, no requests are recorded.
memory_rejected	Number of rejected requests during memory flow control. This parameter takes effect when memory flow control is enabled, and its value is not cleared after memory flow control is disabled. The requests from the paths in the **allow_path** whitelist are not recorded. If **allow_path** is set to ******, no requests are recorded.
rejected_cpu	Number of requests rejected when the CPU flow control threshold is exceeded. This parameter takes effect when CPU flow control is enabled, and its value is not cleared after CPU flow control is disabled.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
ip_address	IP addresses and the number of requests. For details, see Table 2-144Table 2-144.
url_sample	Request path sampling. The number of URLs of a request are collected based on the configured time and sampling interval. For details, see Table 2-145.

Table 2-144 ip_address

Parameter	Description
ip	Source IP address for accessing the node.
method	Number of access requests from an IP address.

Table 2-145 url_sample

Parameter	Description
url	Request URL
method	Method corresponding to the request path
remote_address	Source IP address and port number in the request
count	How many times a path is sampled

Enabling and Viewing Access Logs

1. Run the following command to enable access logging:
   - Enable access logging for all nodes in a cluster.
     ```
     PUT /_access_log?duration_limit=30s&capacity_limit=1mb
     ```
   - Enable access logging for a specified node in a cluster.
     ```
     PUT /_access_log/{nodeId}?duration_limit=30s&capacity_limit=1mb
     ```
     {nodeId} indicates the node ID.

     Table 2-146 Configuration items for configuring access logging

Configuration Item	Type	Description
duration_limit	String	Maximum duration of access log records. When this duration is reached, the recording stops. Value range: 10 to 120 Unit: s Default value: **30**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuration Item	Type	Description
capacity_limit	String	Maximum memory size for recording access logs. When the size of an access log reaches this value, access logging stops. Value range: 1 to 5 Unit: MB Default value: 1

NOTE

- Access logging stops when either **duration_limit** or **capacity_limit** is reached.
- If all parameters are set to **null**, they will be restored to their default values.

2. Run the following command to check access logs:
   - API for checking the access logs of all nodes in a cluster
     ```
     GET /_access_log
     ```
   - API for checking the access logs of a specified node in a cluster
     ```
     GET /_access_log/{nodeId}
     ```
     {nodeId} indicates the node ID.

   Example response:

```
{
  "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  },
  "cluster_name" : "css-flowcontroller",
  "nodes" : {
    "8x-ZHu-wTemBQwpcGivFKg" : {
      "name" : "css-flowcontroller-ess-esn-1-1",
      "host" : "10.0.0.98",
      "count" : 2,
      "access" : [
        {
          "time" : "2021-02-23 02:09:50",
          "remote_address" : "/10.0.0.98:28191",
          "url" : "/_access/security/log?pretty",
          "method" : "GET",
          "content" : ""
        },
        {
          "time" : "2021-02-23 02:09:52",
          "remote_address" : "/10.0.0.98:28193",
          "url" : "/_access/security/log?pretty",
          "method" : "GET",
          "content" : ""
        }
      ]
    }
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-147 Response parameters

Parameter	Description
name	Node name
host	Node IP address
count	Number of node access requests in a statistical period
access	Details about node access requests in a statistical period. For details, see **Table 2-148**.

Table 2-148 access

Parameter	Description
time	Request time
remote_address	Source IP address and port number in the request
url	Original URL of the request
method	Method corresponding to the request path
content	Request content

3. Run the following commands to delete access logs.

   API for deleting access logs for all nodes:

   ```
   DELETE /_access_log
   ```

Enabling Access Logging in Files

Typically you record access logs in files to locate faults. After faults are rectified, you should disable it.

1. Run the following command to enable access logging in files:

   ```
   PUT /_cluster/settings
   {
     "persistent": {
       "flowcontrol.log.file.enabled": true
     }
   }
   ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-149 Configuration items for enabling access logging in files

Parameter	Type	Description
flowcontrol.log.file.enabled	Boolean	Whether to record the details of each request in the access log file. The log file name is **Cluster name_access_log.log**. You can check this file only through the log backup function. The value can be: ● true ● false (default value)

2. Run the following command to disable access logging in files:

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.log.file.enabled": false
  }
}
```

Enabling CPU Flow Control

1. Run the following command to enable CPU flow control:

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.cpu.enabled": true,
    "flowcontrol.cpu.percent_limit": 80,
    "flowcontrol.cpu.allow_path": ["index/_search"]
  }
}
```

Table 2-150 Configuration items for configuring access logging

Configuration Item	Type	Description
flowcontrol.cpu.enabled	Boolean	Whether to enable CPU flow control. The value can be: ● true ● false (default value)
flowcontrol.cpu.percent_limit	Integer	Maximum CPU usage (%) of a node before flow control is triggered. Default value: **90**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuration Item	Type	Description
flowcontrol.cpu.allow_path	List<String>	Path whitelist for CPU flow control. The paths specified using this parameter are exempt from CPU flow control. By default, query APIs controlled by the cluster are exempt from CPU flow control. This prevents the failure to query cluster information when the CPU usage reaches the threshold. Example: ● "flowcontrol.memory.allow_path": "/index/_search", ● "flowcontrol.memory.allow_path": "/index*/_search", ● "flowcontrol.memory.allow_path": ["/index/_search", "/index1/_bulk"], A maximum of 10 paths can be configured. A path can contain up to 32 characters. The default value is null.

NOTE

If all parameters are set to **null**, they will be restored to their default values.

2. Run the following command to disable CPU flow control:

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.cpu.enabled": false
  }
}
```

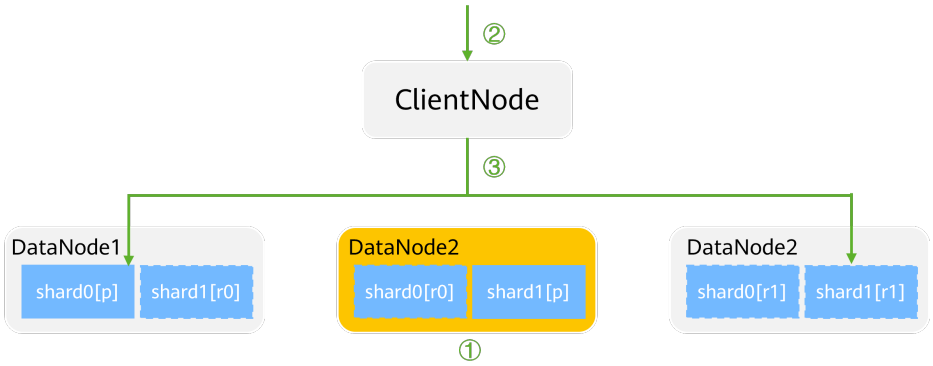
2.11.7 Configuring Query Traffic Isolation

In distributed search and data analytics scenarios, faulty cluster nodes—with memory faults or sustained high CPU usage—may lead to increased query response times or service availability issues. To address this, we need a quick and effective way to route query traffic away from faulty nodes and redirect it to other healthy nodes, thus ensuring service continuity and stability. CSS Elasticsearch clusters provide a query traffic isolation feature for this purpose. With it, you can configure specific nodes as low-priority nodes or isolated nodes and define an index whitelist for exceptions. For shard query requests to indexes that are not on this whitelist, the system preferentially or completely bypasses these nodes, ensuring that faulty nodes won't affect services.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

How the Feature Works

Figure 2-39 How query traffic isolation works



Here is how query traffic isolation works:

1. When a node in your cluster experiences issues, you configure it as either a low-priority node or an isolated node.
2. Query requests are broken down into per-shard queries.
3. Per-shard queries are preferentially allocated to healthy nodes, bypassing isolated nodes completely, and bypassing low-priority nodes when possible.
   - A per-shard query is allocated to a low-priority node only if this shard is not available elsewhere.
   - If the target shard of a query exists only on an isolated node, the query fails.

This way, CSS keeps query traffic away from problematic cluster nodes, ensuring query performance and service availability.

Constraints

- Query traffic isolation is supported only when the Elasticsearch cluster version is 7.10.2 and the image version is 7.10.2_25.9.0_xxx or later.
- By default, indexes prefixed with **.kibana** and **.opendistro** are exempted from query traffic isolation settings.

Logging In to Kibana

Log in to Kibana and go to the command execution page. Elasticsearch clusters support multiple access methods. This topic uses Kibana as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters** > **Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuring Query Traffic Isolation

Run the following command to configure low-priority nodes, isolated nodes, and the index whitelist (or allow-list).

```
PUT _cluster/settings
{
  "persistent": {
    "cluster.routing.css.preference.query.low_priority_nodes": "node1,node2",
    "cluster.routing.css.preference.query.isolated_nodes": "node3,node4",
    "cluster.routing.css.preference.query.allow_list": "index1,index2,test-index.*"
  }
}
```

Table 2-151 Parameter description

Parameter	Type	Default Value	Description
cluster.routing.css.preference.query.low_priority_nodes	String	Empty (no low-priority nodes)	Configure low-priority nodes. For per-shard query requests to indexes that are not on the whitelist, the system preferentially bypasses low-priority nodes. A per-shard query is allocated to a low-priority node only if this shard is not available elsewhere. When configuring multiple nodes, use a comma (**,**) to separate them.
cluster.routing.css.preference.query.isolated_nodes	String	Empty (no isolated nodes)	Configure isolated nodes. With the exception of whitelisted indexes, per-shard query requests bypass isolated nodes completely. If the target shard exists only on an isolated node, the query fails. When configuring multiple nodes, use a comma (**,**) to separate them.
cluster.routing.css.preference.query.allow_list	String	Empty	Configure indexes that are exempted from the above settings. Use a comma (**,**) to separate different index names. Regular expression matching is supported. By default, indexes prefixed with **.kibana** and **.opendistro** are exempted from query traffic isolation settings. This means the whitelist (or allow-list) will include the indexes you configure here, as well as these default ones.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2.11.8 Configuring Large Query Isolation

Large query isolation can be configured to manage queries that have high memory usage or take too long to complete. This helps improve the stability of Elasticsearch clusters and prevent out-of-memory (OOM) exceptions.

As business grows, your Elasticsearch clusters may face mounting query pressure. Some complex queries may occupy excessive node memory, triggering frequent garbage collection or even OOM exceptions, which may compromise cluster performance and stability. Large query isolation enables effective management of memory-intensive, time-consuming query requests, ensuring cluster stability. Large query isolation includes the following:

- Isolating large queries: Manages memory-intensive/time-consuming queries separately to avoid impacting other queries.
- Query cancellation based on a heap memory usage threshold: Cancels a large query in the isolation pool when the node heap memory usage reaches a predefined threshold.
- Global query timeout: Automatically cancels queries when they last longer than a predefined timeout. This timeout applies globally.

How the Feature Works

- Defining large queries:
  - The system checks the memory usage of all ongoing queries and flags queries that exceed a predefined memory usage threshold as large queries.
  - The system periodically checks the execution duration of all ongoing queries and flags queries that exceed a predefined duration threshold as large queries.
- Query cancellation policies:
  - fair: Determines which query to cancel by considering both memory usage and execution duration.
  - mem-first: Cancels the query that has the highest memory usage.
  - time-first: Cancels the query that has lasted the longest.
- Native cancel API: Elasticsearch's native cancel API can be used to cancel tasks, ensuring compatibility.

Constraints

Only Elasticsearch 7.6.2 and 7.10.2 support large query isolation, which is enabled by default. The global timeout is disabled by default for large query isolation. You can enable and configure it via an API when necessary. Any change takes effect immediately.

Logging In to Kibana

Log in to Kibana and go to the command execution page. Elasticsearch clusters support multiple access methods. This topic uses Kibana as an example to describe the operation procedures.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Configuring Large Query Isolation

Large query isolation places large queries in an isolation pool, where they may be cancelled based on preset memory or duration thresholds. Large query isolation is **enabled by default**. You can modify this setting whenever necessary. Any change takes effect immediately.

1. Run the following command to enable or disable large query isolation:

```
PUT _cluster/settings
{
  "persistent": {
    "search.isolator.enabled": true
  }
}
```

Table 2-152 Setting large query isolation

Parameter	Type	Default Value	Description
search.isolator.enabled	Boolean	true	Whether to enable large query isolation. When enabled, large queries are managed separately from other normal queries. The value can be: • **true**: Enable large query isolation. • **false**: Disable large query isolation.

2. Run the following commands to configure thresholds that define large queries:

```
PUT _cluster/settings
{
  "persistent": {
    "search.isolator.memory.task.limit": "50MB",
    "search.isolator.time.management": "10s"
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-153 Parameters for configuring large query isolation thresholds

Parameter	Type	Default Value	Description
search.isolator.memory.task.limit	String	50MB	Large query memory threshold: When a query requests more memory than specified by this threshold, it is placed into an isolation pool. Value format: number + unit • Number: a natural number • Unit: B, K, KB, M, MB, G, GB, T, TB, P, or PB (case-insensitive) Minimum value: 0 (all queries are placed into the isolation pool) Maximum value: maximum node heap memory Lowering this value will cause more queries to be placed into the isolation pool, which will increase its memory usage. If you do lower this value, you should also increase the values of **search.isolator.memory.pool.limit** and **search.isolator.count.limit**, so that the isolation pool can hold more queries. This helps avoid triggering the circuit breaker mechanism due to resource exhaustion (for example, frequent query cancellation).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
search.isolator.time.management	String	10s	Large query execution duration threshold: When a query has lasted longer than specified by this threshold, it is placed into an isolation pool. Value format: number + unit • Number: a natural number • Unit: nanos (nanosecond), micros (microsecond), ms (millisecond), s (second), m (minute), h (hour), or d (day) Minimum value: 0 (all queries are placed into the isolation pool) Lowering this value will cause more queries to be placed into the isolation pool, which will increase its memory usage. If you do lower this value, you should also increase the values of **search.isolator.memory.pool.limit** and **search.isolator.count.limit**, so that the isolation pool can hold more queries. This helps avoid triggering the circuit breaker mechanism due to resource exhaustion (for example, frequent query cancellation).

3. Configure the isolation pool resource usage thresholds for triggering query cancellation.

```
PUT _cluster/settings
{
  "persistent": {
    "search.isolator.memory.pool.limit": "50%",
    "search.isolator.count.limit": 1000,
    "search.isolator.memory.heap.limit": "90%"
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-154 Parameters for configuring query cancellation thresholds

Parameter	Type	Default Value	Description
search.isolator.memory.pool.limit	String	50%	Maximum memory usage of the isolation pool as a percentage of the maximum node heap memory. When the total memory usage of large queries in the isolation pool exceeds this limit, the system cancels one of the large queries in the isolation pool based on a predefined policy to free resources and prevent memory overflow. Value range: 0.0–100.0% If your cluster primarily handles large queries (high memory usage or long execution time), increase this value. Meanwhile, set **search.isolator.memory.task.limit** and **search.isolator.time.management** accordingly to control the number of queries placed into the isolation pool.
search.isolator.count.limit	Integer	1000	Maximum number of large queries allowed in the isolation pool. When this limit is reached, no more queries can be added to the isolation pool, preventing resource exhaustion. Value range: **10–50000** If your cluster primarily handles large queries (high memory usage or long execution time), increase this value. Meanwhile, set **search.isolator.memory.task.limit** and **search.isolator.time.management** accordingly to control the number of queries placed into the isolation pool.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
search.isolator.memory.heap.limit	String	90%	Node heap memory usage that triggers large query cancellation in the isolation pool. When this threshold is reached, the system cancels one of the large queries in the isolation pool based on a predefined policy to free resources and prevent memory overflow. Value range: 0.0–100.0% When **indices.breaker.total.use_real_memory** is enabled, this value must be lower than **indices.breaker.total.limit**. Otherwise, the native Elasticsearch circuit breaker will always be triggered first. For details, see **Circuit breaker settings**. If you anticipate traffic peaks or surges, you can lower this value to have the isolation pool's circuit breaker triggered earlier, thus preventing heap memory overload.

4. Run the following command to set the query cancellation policy:

```
PUT _cluster/settings
{
  "persistent": {
    "search.isolator.strategy": "fair",
    "search.isolator.strategy.ratio": "0.5%"
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-155 Parameters for configuring a query cancellation policy

Parameter	Type	Default Value	Description
search.isolator.strategy	String	fair	Policy for determining which query to cancel when query cancellation is triggered. • fair (default): Determine which query to cancel by considering both memory usage and execution duration. If the difference between the memory usage of two candidate queries ≤ **maximum Elasticsearch heap memory x fair policy threshold**, the query that has a longer execution duration will be cancelled; on the contrary, if the difference is greater than that, the more memory-intensive query will be cancelled instead. Maximum Elasticsearch heap memory = **min(31, total node memory/2)** (GB). • mem-first: Cancels the query that has the highest memory usage. • time-first: Cancels the query that has lasted the longest. The large query isolation pool is checked every second until the heap memory is within a safe range.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
search.isolator.strategy.ratio	String	1%	Fair policy threshold. This is the ratio of the memory usage difference between two candidate queries in the isolation pool to the maximum node heap memory. • When the memory usage difference between large queries in the isolation pool is small, the system preferentially cancels the query with the longest execution duration. • Otherwise, it cancels the query with the highest memory usage. This parameter is valid only when **search.isolator.strategy** is set to **fair**. Value range: 0.0–100.0% You are advised to use the default value. Adjust only if necessary and with caution.

5. Run the following command to set the maximum number of cancelled query records retained in the large query isolation log:

```
PUT _cluster/settings
{
  "persistent": {
    "search.isolator.log.count": "100"
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-156 Parameter description

Parameter	Type	Default Value	Description
search.isolator.log.count	Integer	100	The maximum number of cancelled query records retained in the large query isolation log. The large query isolation log records cancelled large queries for query performance analysis and optimization. Once this limit is exceeded, the system automatically deletes the oldest records to control the log's memory footprint. Value range: 0–5000 Setting this value to **0** disables the large query isolation log.

You can use the following APIs to query log information about cancelled queries:

- Query statistics about cancelled queries on all nodes:
  ```
  GET /_isolator_metrics
  ```
- Query statistics about cancelled queries on a specified node:
  ```
  GET /_isolator_metrics/{nodeId}
  ```
- Query details about cancelled queries on all nodes:
  ```
  GET /_isolator_metrics?detailed
  ```
- Query details about cancelled queries on a specified node:
  ```
  GET /_isolator_metrics/{nodeId}?detailed
  ```

Table 2-157 Parameter description

Parameter	Type	Default Value	Description
node_id	String	N/A	Specifies one or more cluster nodes. • Single node: Enter the node ID. • Multiple nodes: Enter multiple node IDs and use a comma (,) to separate them. You can run the following command to obtain node IDs: `GET _cat/nodes?s=n&h=n,id&v=true&full_id=true`

Example response:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
{
  "_nodes": {
    "total": 1,
    "successful": 1,
    "failed": 0
  },
  "cluster_name": "test",
  "nodes": {
    "CTqrZFXWTzmLonSZyNMKkQ": {
      "name": "test-ess-esn-1-1",
      "host": "172.16.101.116",
      "total_cancel": 0, //Total number of cancelled queries
      "isolator_cancel": 0,    //Number of queries cancelled because isolation pool thresholds were exceeded
      "out_of_time_cancel": 0    //Number of queries cancelled due to timeout
    }
  }
}
```

Configuring Global Query Timeout

When a global query timeout is configured, queries that exceed the specified duration are automatically cancelled, and the message "cancel cause by global time limit" is returned. This prevents long-running queries from consuming excessive resources. Global query timeout is **disabled by default**. You can modify this setting when necessary. Any change takes effect immediately.

Run the following command to enable and configure a global query timeout:

```
PUT _cluster/settings
{
  "persistent": {
    "search.isolator.time.enabled": true,
    "search.isolator.time.limit": "110s"
  }
}
```

Table 2-158 Parameters for setting the global query timeout

Parameter	Type	Default Value	Description
search.isolator.time.enabled	Boolean	false	Whether to enable a global query timeout. When enabled, queries are automatically cancelled when they last longer than a predefined timeout. The value can be: • true: Enable global query timeout. • false: Disable global query timeout.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
search.isolator.time.limit	String	120s	The value of the global query timeout. Value format: number + unit • Number: a natural number • Unit: nanos (nanosecond), micros (microsecond), ms (millisecond), s (second), m (minute), h (hour), or d (day) Minimum value: 0 (to cancel all queries)

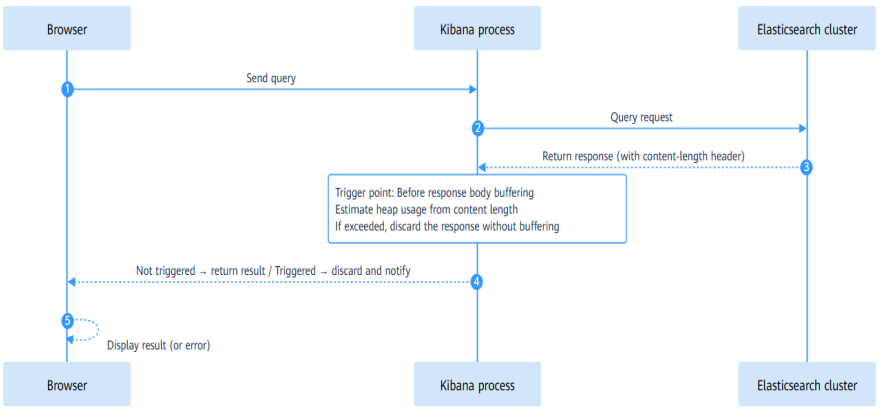
2.11.9 Configuring the Kibana Query Response Circuit Breaker

During query operations, Elasticsearch may return extremely large response bodies, such as full index exports or high-cardinality aggregations. When the Kibana backend process buffers and parses these large responses, heap memory can be quickly exhausted, causing the process to crash due to OOM (Out of Memory), which can affect all users. CSS provides the query response circuit breaker feature. When the Kibana backend process receives a response, before buffering the response body, the circuit breaker estimates the heap usage based on the **content-length** field in the response header. If the estimated heap usage exceeds the configured threshold, the response is discarded and an error is returned, preventing the process from crashing.

How the Feature Works

When the Kibana backend process receives a query response, the circuit breaker determines whether to allow or block the response before buffering the response body.

Figure 2-40 Query process



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

The circuit breaker is triggered when either of the following conditions is met. After it is triggered, the response body is discarded directly:

- **Absolute response size limit**: The circuit breaker is triggered when **maxContentLength > 0** and **the response body's contentLength ≥ maxContentLength**. This indicates that the response body itself is too large and is rejected regardless of the current heap usage.
- **Heap usage limit**: The circuit breaker is triggered when **contentLength × heapMultiplier + heapUsed > heapSizeLimit × maxPercentage**. This indicates that buffering the response is estimated to exceed the configured heap usage limit.

Key parameters are as follows:

- **maxContentLength**: Absolute response body size limit. The circuit breaker is triggered when the response body reaches this size.
- **contentLength**: Size of the current response body, which is the value of the **content-length** field in the HTTP response header.
- **heapSizeLimit**: Maximum heap size of the process.
- **heapUsed**: Current heap usage of the process.
- **heapMultiplier**: Heap usage estimation multiplier. The default value is 4. The value of **contentLength** is multiplied by this value to estimate the heap usage after loading the response into memory. The default value of 4 is used because JSON data parsed into JavaScript objects typically consumes approximately 3 to 5 times more memory than the original text. A multiplier of 4 provides a balance between protection and resource utilization.
- **maxPercentage**: Maximum allowed heap usage in percentage.

NOTE

The query response circuit breaker applies to the Kibana backend process and complements the Elasticsearch cluster feature **Configuring Large Query Isolation**. The former protects the Kibana process from oversized responses, while the latter manages large queries on the Elasticsearch engine side.

Constraints

- The Kibana query response circuit breaker is supported only for Elasticsearch 7.10.2 clusters (image version 7.10.2_26.6.0_xxx or later). It is **enabled by default** for clusters that meet these version requirements.
- For clusters with the security mode enabled, Kibana query response circuit breaker parameters are visible and configurable only in the Global tenant.
- Changes made in **Advanced Settings** do not take effect immediately. They are synchronized periodically in the background and take effect within 1 minute.

Logging In to the Kibana Advanced Settings Page

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

4. (Optional) For a security-mode cluster, switch to the Global tenant before configuring Kibana query response circuit breaker parameters.

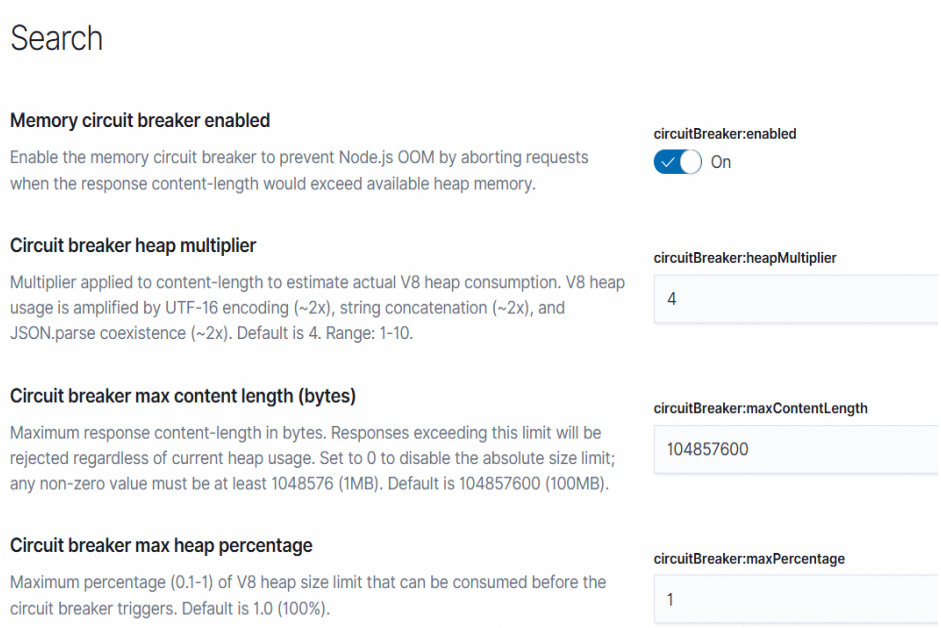
   On the Kibana page, click the username in the upper right corner and choose **Switch tenants**. On the **Select your tenant** page, select **Global**, and click **Confirm**.

5. In the left navigation pane, choose **Management > Stack Management**, and select **Advanced settings**.

Configuring Kibana Query Response Circuit Breaker Parameters

1. On the **Settings** page, locate the **Search** module and configure the following Kibana query response circuit breaker parameters.

   Figure 2-41 Configuring the Kibana query response circuit breaker



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

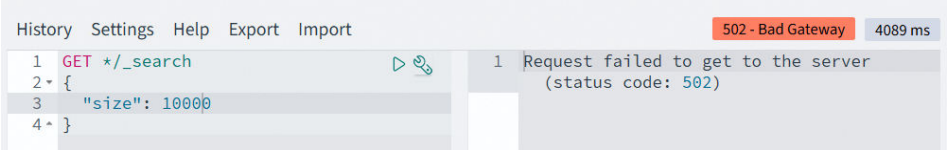
Table 2-159 Configuring Kibana query response circuit breaker parameters

Parameter	Type	Default Value	Description
circuitBreaker:enabled	Boolean	On	Whether to enable the Kibana query response circuit breaker. • **On**: Enables the circuit breaker. • **Off**: Disables the circuit breaker. After the circuit breaker is disabled, the Kibana backend process no longer blocks oversized responses. Large query responses may cause the process to crash due to OOM, affecting all users. Disable this feature only when there is no risk of large queries in your workloads.
circuitBreaker:heapMultiplier	Integer	4	Heap usage estimation multiplier used to estimate heap consumption after a response is loaded into the heap. Value range: 1–10
circuitBreaker:maxContentLength	Integer	104857600	Absolute response body size limit. Value range: • 0 (no limit) • 1048576 (1 MB) to 1073741824 (1 GB) Unit: bytes
circuitBreaker:maxPercentage	Float	1.0	Maximum allowed heap usage in percentage. Value range: 0.1–1.0 1.0 indicates that the entire available heap can be used.

2. Click **Save changes** to save the changes.
3. Wait for about 1 minute. Then run a large query to verify that the circuit breaker takes effect.
   a. On Kibana, expand the menu in the upper-left corner, and choose **Dev Tools**.
   b. Run a large query. The request fails with HTTP status code 502.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 2-42 Large query



c. In the browser, press **F12** to launch Developer Tools, switch to the **Network** tab, and check **Response**. If **Memory circuit breaker triggered** is displayed, the query response has been blocked by the circuit breaker.

Figure 2-43 Query response



2.11.10 Configuring Automatic Retry for Failed Shard Allocation

As an Elasticsearch cluster runs, shard allocation may fail due to factors such as high node load or a triggered circuit breaker. After a shard allocation failure, the system automatically retries the allocation. However, once the number of retries reaches the configured limit, the shard remains in the unassigned state and is no longer considered for reallocation. This may prevent read and write requests for the affected index from being processed properly. Normally, after this occurs, you can only retry the allocation manually by calling the Reroute API (**/_cluster/reroute?retry_failed=true**). This not only increases the operational burden, but also means that hours or even days may pass before manual intervention, preventing temporary failures from recovering automatically. CSS provides automatic retry for failed shard allocation. Once a shard reaches its maximum allocation failure limit, the system periodically retries allocation using an exponential backoff strategy, enabling self-healing without manual intervention.

How the Feature Works

After automatic retry for failed shard allocation is enabled, the system automatically retries shards whose number of allocation failures has reached the retry limit. The limit is controlled by **index.allocation.max_retries** and defaults to 5. The retry process is as follows:

1. Automatic retries start after the base delay.
2. If a retry succeeds, the shard is no longer tracked. If the retry fails, the backoff delay is doubled before the next retry.
3. Once the backoff delay reaches the configured maximum delay, it no longer increases. Subsequent retries are performed at the maximum delay.
4. The system periodically checks for failed shards that need to be retried at the configured polling interval. If any are found, they are retried.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-160 Exponential backoff example (base delay: 1m, maximum delay: 120m)

Retry Count	Backoff Delay	Description
0	1m	Shard is detected for the first time. The retry is allowed to proceed after the base delay.
1	2m	The first retry fails. The backoff delay is doubled.
2	4m	The second retry fails.
3	8m	The third retry fails.
4	16m	The fourth retry fails.
...	...	The delay continues to double until it reaches the maximum delay.
N	120m	The maximum delay is reached. Subsequent retries are performed at the maximum delay.

Constraints

Automatic retry for failed shard allocation is supported only for Elasticsearch 7.10.2 clusters (image version 7.10.2_26.6.0_xxx or later). The feature is **enabled by default** for clusters that meet these version requirements.

Logging In to Kibana

Log in to Kibana and go to the command execution page. Elasticsearch clusters support multiple access methods. This topic uses Kibana as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Configuring Automatic Retry Parameters for Failed Shard Allocation

You can dynamically configure the parameters for automatic retry of failed shard allocation. The changes take effect immediately after configuration.

1. Enable or disable automatic retry for failed shard allocation.

```
PUT _cluster/settings
{
  "persistent": {
    "cluster.routing.allocation.retry_failed.auto_enabled": true
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  }
}
```

Table 2-161 Parameter for enabling or disabling automatic retry for failed shard allocation

Parameter	Type	Default Value	Description
cluster.routing.allocation.retry_failed.auto_enabled	Boolean	true	Enables or disables automatic retry for failed shard allocation. The value can be: • **true**: Enables automatic retry. • **false**: Disables automatic retry. After automatic retry is disabled, shards whose number of allocation failures reaches the retry limit remain in the unassigned state and must be retried manually using the Reroute API, which may result in prolonged service interruption. It is recommended that you disable this feature only when automatic recovery is not required for your cluster.

2. Configure the exponential backoff parameters that control the retry interval and frequency.

```
PUT _cluster/settings
{
  "persistent": {
    "cluster.routing.allocation.retry_failed.auto_base_delay": "1m",
    "cluster.routing.allocation.retry_failed.auto_max_delay": "120m",
    "cluster.routing.allocation.retry_failed.auto_poll_interval": "5m"
  }
}
```

Table 2-162 Exponential backoff parameters

Parameter	Type	Default Value	Description
cluster.routing.allocation.retry_failed.auto_base_delay	TimeValue	1m	Base delay for the backoff strategy. This specifies the initial wait time before the first automatic retry. After each failed retry, the delay doubles. Value range: 1m–1440m Format: number + unit. Supported units: ms (milliseconds), s (seconds), m (minutes), h (hours), and d (days).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
cluster.routing.allocation.retry_failed.auto_max_delay	TimeValue	120m	Maximum delay for the backoff strategy. Once the retry interval reaches this value, it no longer increases, preventing the backoff delay from growing indefinitely. Value range: 1m–1440m Format: number + unit. Supported units: ms (milliseconds), s (seconds), m (minutes), h (hours), and d (days).
cluster.routing.allocation.retry_failed.auto_poll_interval	TimeValue	5m	Polling interval. The system periodically checks for shards that have exceeded the retry limit, ensuring that shards are not missed when the cluster state remains stable. Value range: 1m–1440m Format: number + unit. Supported units: ms (milliseconds), s (seconds), m (minutes), h (hours), and d (days). It is recommended that you keep the default value. A smaller value increases the polling frequency and may have a slight impact on cluster performance. A larger value may delay the detection of shards that need to be retried.

3. (Optional) Verify the configuration. Query the current automatic retry configuration for failed shard allocation:

```
GET _cluster/settings?filter_path=**.retry_failed.auto_*
```

If the returned result contains the configured parameters and their values match the configured values, the configuration has taken effect.

FAQ

- **Q1: How do I manually retry failed shard allocation after disabling automatic retry?**

  After automatic retry is disabled, you can manually retry failed shard allocation by calling the Reroute API:

```
POST /_cluster/reroute?retry_failed=true
```

- **Q2: Does automatic retry for failed shard allocation affect ongoing shard allocation?**

  No. Automatic retry only allocates shards that have reached the **index.allocation.max_retries** limit and are now in the unassigned state. It

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

does not trigger **Move** or **Balance** operations and therefore does not affect ongoing shard allocation.

2.11.11 Enhancing Data Ingestion Performance

In large-scale data ingestion scenarios, such as log analytics and real-time monitoring, the traditional Elasticsearch routing mechanism can generate excessive internal forwarding requests as the number of index shards and the write throughput increase. This can lead to write rejections and increased long-tail latency. To address this issue, CSS provides several ways to enhance ingestion performance: bulk routing, bulk aggregation, text indexing acceleration, and merge optimization. They can help reduce performance overhead and significantly improve write stability and throughput in heavy-load scenarios.

How the Feature Works

To balance performance and functionality, CSS provides four ingestion performance enhancement solutions. The core idea is to eliminate or mitigate performance bottlenecks by reducing cross-node communication and accelerating index building.

Table 2-163 Ways to enhance data ingestion performance

Solution	Description	When to Use	Details
Bulk routing	Bulk routing reduces cross-node communication and request forwarding by the coordinator node, helping to prevent write rejections. How it works: Rather than routing each document individually based on its ID (the default behavior), this optimization forces multiple documents within the same bulk request to be routed together to the same shard or local node.	Enable this feature for large-scale clusters with a high number of shards. It is particularly effective when bulk request throughput is so intense that it exhausts the coordinator node's network I/O or CPU resources.	**Configuring Bulk Routing**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Solution	Description	When to Use	Details
Bulk aggregation	Bulk aggregation reduces system overhead and improves single-node capacity by minimizing resource lock contention during write operations. How it works: Instead of processing documents individually within the same bulk request, the system processes them as a unified batch, reducing memory allocations and lock contention.	Enable this feature for ultra-high concurrency write scenarios where throughput is critical. We recommend enabling it when monitoring indicates high system CPU usage or significant memory allocation pressure.	**Configuring Bulk Aggregation**
Text indexing acceleration	Text indexing acceleration significantly speeds up the indexing of text and keyword fields. How it works: Through an optimized indexing process, improved memory utilization, and accelerated tokenization, this feature improves the overall efficiency of text processing.	Enable this feature when you need high-speed index building with relatively simple tokenization logic. Examples of such scenarios include log auditing and full-text search, where large-volume text and numerous keyword fields are involved.	**Configuring Text Indexing Acceleration**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Solution	Description	When to Use	Details
Merge optimization	Merge optimization ensures write stability by preventing write throttling caused by slow segment merging. How it works: By dynamically increasing merging threads for index shards, this feature accelerates the merging of small segments and reclaims write cache space in a timely manner.	Enable this feature in high-frequency write scenarios, particularly when the write throughput increases drastically after enabling the three optimizations above and writes become unstable due to merge throttling.	**Configuring Merge Optimization**

Constraints

Only Elasticsearch 7.10.2 clusters support data ingestion performance enhancement.

Logging In to Kibana

Log in to Kibana and go to the command execution page. Elasticsearch clusters support multiple access methods. This topic uses Kibana as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Configuring Bulk Routing

CAUTION

After bulk routing is enabled (that is, **index.bulk_routing** is set to **pack** or **local_pack**), documents are no longer routed by ID. This may affect routing-related functionality. For example, ID-based GET requests may fail because the shards cannot be located.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Enable bulk routing to alleviate request forwarding pressure for the node that receives requests.

```
PUT /{index_name}/_settings
{
  "index.bulk_routing": "local_pack"
}
```

Table 2-164 Parameters for bulk routing

Parameter	Type	Default Value	Description
index_name	String	N/A	Specifies one or more indexes. • Single index: Enter the index name, for example, **my_index**. • Multiple indexes: Enter multiple index names and use a comma (,) to separate them, for example, **my_index1,my_index2**. • Wildcard: Use the wildcard (*) to match multiple indexes. For example, **myindex*** indicates all indexes whose name starts with **myindex**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
index.bulk_routing	String	default	Specifies the internal routing policy for processing documents within bulk write requests. • **default**: Uses the default routing behavior. Each document within the same bulk request is hashed and routed separately based on its **_id**. This ensures even data distribution across all shards but can generate significant cross-node forwarding overhead during large-scale bulk writes. • **pack**: Routes all documents within the same bulk request to a single, randomly selected shard. This eliminates per-document hash calculations and cross-node forwarding, reducing request forwarding pressure on the coordinator node. • **local_pack** (recommended): First attempts to route all documents within the same bulk request to a local shard on the data node that received the bulk request. If that node does not host a shard of the target index, it falls back to the **pack** mode. This option is recommended because it achieves zero cross-node forwarding.

Configuring Bulk Aggregation

Enable bulk aggregation to enable the system to process documents within each bulk request as a unified batch, instead of processing each document individually. This significantly reduces CPU overhead and lock contention during ultra-high concurrency write operations.

```
PUT /{index_name}/_settings
{
  "index.aggr_perf_batch_size": "128"
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-165 Parameters for bulk aggregation

Parameter	Type	Default Value	Description
index.aggr_perf_batch_size	Integer	1	Specifies the maximum number of documents to process as a single batch. Minimum value: **1** (disables bulk aggregation so that documents will be processed individually) Recommended value: **128** Any value greater than 1 enables bulk aggregation. The system will internally reaggregate documents from bulk requests into batches up to this size. Larger batch sizes increase memory consumption per batch. Adjust this parameter based on your workload requirements and available system resources. The actual number of documents per batch is calculated as: MIN(Number of documents within the bulk request, aggr_perf_batch_size)

Configuring Text Indexing Acceleration

CAUTION

Text indexing acceleration cannot be enabled for indexes whose mapping contains nested fields. The system will not be able to start acceleration due to compatibility issues.

Enable text indexing acceleration to speed up the indexing of text and keyword fields.

```
PUT /{index_name}/_settings
{
  "index.native_speed_up": true,
  "index.native_analyzer": true
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-166 Parameters for text indexing acceleration

Parameter	Type	Default Value	Description
index.native_speed_up	Boolean	false	Whether to enable text indexing acceleration. The value can be: • **true**: Enables text indexing acceleration. This will significantly accelerate the indexing of text and keyword fields, optimize memory usage during indexing, and reduces the frequency of garbage collection (GC) caused by high-frequency writes. • **false**: Disables text indexing acceleration.
index.native_analyzer	Boolean	false	Whether to enable tokenization acceleration. This parameter is available only when **index.native_speed_up** is set to **true**, and it applies to text fields only. The value can be: • **true**: Enables tokenization acceleration. This will significantly reduce CPU time for tokenization. • **false**: Disables tokenization acceleration.

Configuring Merge Optimization

Increase the number of threads used for segment merging for index shards. This helps prevent write throttling caused by slow segment merging.

```
PUT /{index_name}/_settings
{
  "index.merge.scheduler.max_thread_count": 8
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-167 Parameters for segment merging

Parameter	Type	Default Value	Description
index.merge.scheduler.max_thread_count	Integer	4	Specifies the maximum number of threads used for segment merging in a single shard. Value range: 1 to (Number of CPU cores on the node/2) The recommended value is 8. It allows you to accelerate segment merging by leveraging the multi-core architecture of CPUs, eliminating write throttling caused by slow segment merging. However, you must ensure that your cluster nodes have sufficient CPU cores to run these threads.

2.11.12 Configuring Enhanced Aggregation

Low aggregation query performance often slows down analysis when large datasets are involved. CSS Elasticsearch clusters leverage vectorization and optimized clustering to accelerate aggregation queries, enabling faster analytics and decision-making.

How the Feature Works

Enhanced aggregation works by pre-sorting and physically clustering data using carefully selected sorting and clustering keys upon ingestion, thereby minimizing data scanning and computational overheads during aggregation.

- Sorting key: A field (such as timestamps) used to physically order documents on disk, ensuring that documents with identical or similar sorting key values are stored contiguously.
- Clustering key: A subset of fields from the sorting key that groups related documents into contiguous physical blocks. This way, aggregations can process contiguous data blocks, instead of scattered documents.

For example, for an index that contains the **region**, **host**, and **value** fields, documents can be sorted by these three fields upon ingestion, and then clustered and stored based on the **region** and **host** fields. In this case, documents with the same **region** or **host** are stored together, and are then sorted by the **value** field within each cluster.

When an aggregation query is performed, data is queried by clustering key (**region** and **host**), or by the **value** field within each cluster. This improves query efficiency.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 2-44 An example of enhanced aggregation

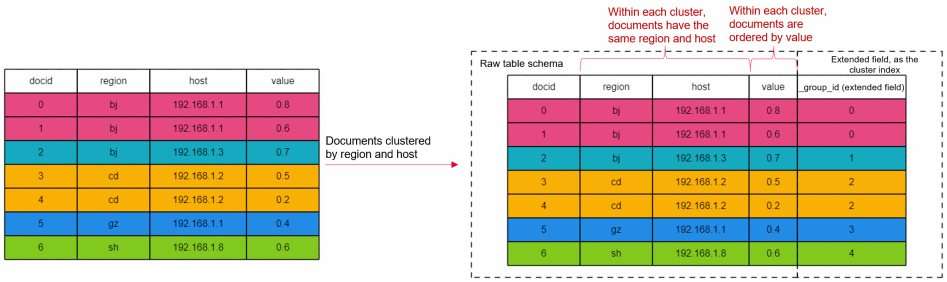


Table 2-168 Common scenarios for enhanced aggregation

Scenario	Description
Aggregation of low-cardinality fields	Aggregates fields that have a small number of unique values. In this case, grouping is used. One example is counting the number of orders per city. A clustering operation can easily achieve this purpose.
Aggregation of high-cardinality fields	Aggregates fields that have a large number of unique values. Histogram aggregation is typically used. One example is counting the hourly visits. A sorting key can be used to accelerate data aggregation by a specified scope or range.
Hybrid aggregation of low- and high-cardinality fields	First groups and aggregates low-cardinality fields (for example, orders per city) using a clustering key, and then creates a histogram for high-cardinality fields (for example, timestamps) using a sorting key. Multi-level clustering improves the efficiency of hybrid queries.

Constraints

Only Elasticsearch 7.10.2 supports enhanced aggregation.

Logging In to Kibana

Log in to Kibana and go to the command execution page. Elasticsearch clusters support multiple access methods. This topic uses Kibana as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Aggregation of Low-Cardinality Fields

Generally, low-cardinality fields use grouping as a way of aggregation. With appropriate sorting keys, grouping prepares the data for batch processing.

For example, to aggregate two low-cardinality fields **city** and **product**, perform the following steps:

1. Run the following command to set the index **testindex**:

```
PUT testindex
{
  "mappings": {
    "properties": {
      "date": {
        "type": "date",
        "format": "yyyy-MM-dd"
      },
      "city": {
        "type": "keyword"
      },
      "product": {
        "type": "keyword"
      },
      "trade": {
        "type": "double"
      }
    }
  },
  "settings": {
    "index": {
      "search": {
        "turbo": {
          "enabled": "true"
        }
      },
      "sort": {
        "field": [
          "city",
          "product"
        ]
      },
      "cluster": {
        "field": [
          "city",
          "product"
        ]
      }
    }
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-169 Parameters for low-cardinality field aggregation

Parameter	Type	Default Value	Description
index.search.turbo.enabled	Boolean	false	Whether to enable enhanced aggregation. Normally, enhanced aggregation must be enabled where aggregations are used. The value can be: • **false**: Disable enhanced aggregation. • **true**: Enable enhanced aggregation.
index.sort.field	Array of strings	N/A	Specify sorting keys. Sorting keys are fields used to sequence or rank documents. You can specify one or multiple fields as sorting keys. When multiple fields are specified, they will apply in the sequence in which they are specified. Documents are first ranked by the first field, then the initial result set is ranked by the second field, and so on. Value range: The value must be fields contained in the index.
index.cluster.field	Array of strings	N/A	Specify clustering keys. Clustering keys determine which documents are collected into the same clusters. During an aggregation operation, documents in the same cluster can be processed in batches, significantly enhancing aggregation performance. Constraint: The clustering keys must be a subset of the sorting keys. Value range: The value must be fields contained in the index.

2. Run the following command to import sample data.

```
PUT /_bulk
{ "index": { "_index": "testindex", "_id": "1" } }
{ "date": "2025-01-01", "city": "cityA", "product": "books", "trade": 3000.0}
{ "index": { "_index": "testindex", "_id": "2" } }
{ "date": "2025-01-02", "city": "cityA", "product": "books", "trade": 1000.0}
{ "index": { "_index": "testindex", "_id": "3" } }
{ "date": "2025-01-01", "city": "cityA", "product": "bottles", "trade": 100.0}
{ "index": { "_index": "testindex", "_id": "4" } }
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
{ "date": "2025-01-02", "city": "cityA", "product": "bottles", "trade": 300.0}
{ "index": { "_index": "testindex", "_id": "5" } }
{ "date": "2025-01-01", "city": "cityB", "product": "books", "trade": 7000.0}
{ "index": { "_index": "testindex", "_id": "6" } }
{ "date": "2025-01-02", "city": "cityB", "product": "books", "trade": 1000.0}
```

3. Run the following query command to aggregate low-cardinality fields.

   For example, query the average **trade** value for each product in different cities.

```
POST testindex/_search
{
  "size": 0,
  "aggs": {
    "groupby_city": {
      "terms": {
        "field": "city"
      },
      "aggs": {
        "groupby_product": {
          "terms": {
            "field": "product"
          },
          "aggs": {
            "avg_trade": {
              "avg": {
                "field": "trade"
              }
            }
          }
        }
      }
    }
  }
}
```

Aggregation of High-Cardinality Fields

High-cardinality fields commonly use sorting keys for histogram aggregation, which facilitates data processing per range or scope.

For example, to aggregate the typical high-cardinality field **date**, perform the following steps:

1. Run the following command to set the index **testindex**:

```
PUT testindex
{
  "mappings": {
    "properties": {
      "date": {
        "type": "date",
        "format": "yyyy-MM-dd"
      },
      "score": {
        "type": "double"
      }
    }
  },
  "settings": {
    "index": {
      "search": {
        "turbo": {
          "enabled": "true"
        }
      },
      "sort": {
        "field": [
          "date"
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
      ]
    }
   }
  }
}
```

Table 2-170 Parameters for high-cardinality field aggregation

Parameter	Type	Default Value	Description
index.search.turbo.enabled	Boolean	false	Whether to enable enhanced aggregation. Normally, enhanced aggregation must be enabled where aggregations are used. The value can be: • **false**: Disable enhanced aggregation. • **true**: Enable enhanced aggregation.
index.sort.field	Array of strings	N/A	Specify sorting keys. Sorting keys are fields used to sequence or rank documents. You can specify one or multiple fields as sorting keys. When multiple fields are specified, they will apply in the sequence in which they are specified. Documents are first ranked by the first field, then the initial result set is ranked by the second field, and so on. Value range: The value must be fields contained in the index.

2. Run the following command to import sample data.

```
PUT /_bulk
{ "index": { "_index": "testindex", "_id": "1" } }
{ "date": "2025-01-01", "score": "12.0"}
{ "index": { "_index": "testindex", "_id": "2" } }
{ "date": "2025-01-02","score": "24.0"}
{ "index": { "_index": "testindex", "_id": "3" } }
{ "date": "2025-01-01","score": "53.0"}
{ "index": { "_index": "testindex", "_id": "4" } }
{ "date": "2025-01-02", "score": "22.0"}
{ "index": { "_index": "testindex", "_id": "5" } }
{ "date": "2025-01-01", "score": "99.0"}
{ "index": { "_index": "testindex", "_id": "6" } }
{ "date": "2025-01-02","score": "26.0"}
```

3. Run the following query command to aggregate high-cardinality fields.

   This query groups the **date** field using a histogram and then calculates the average score.

```
POST testindex/_search?pretty
{
  "size": 0,
  "aggs": {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  "groupbytime": {
    "date_histogram": {
      "field": "date",
      "calendar_interval": "day"
    },
    "aggs": {
      "avg_score": {
        "avg": {
          "field": "score"
        }
      }
    }
  }
}
```

Hybrid Aggregation of Low- and High-Cardinality Fields

Where low-cardinality and high-cardinality fields are mixed, first group and aggregate low-cardinality fields using a clustering key, then create a histogram for high-cardinality fields using a sorting key.

For example, to first group the low-cardinality field **city**, then group the low-cardinality field **product**, and then group the high-cardinality field **date** into a histogram, perform the following steps:

1. Run the following command to set the index **testindex**:

```
PUT testindex
{
  "mappings": {
    "properties": {
      "date": {
        "type": "date",
        "format": "yyyy-MM-dd"
      },
      "city": {
        "type": "keyword"
      },
      "product": {
        "type": "keyword"
      },
      "trade": {
        "type": "double"
      }
    }
  },
  "settings": {
    "index": {
      "search": {
        "turbo": {
          "enabled": "true"
        }
      },
      "sort": {
        "field": [
          "city",
          "product",
          "date"
        ]
      },
      "cluster": {
        "field": [
          "city",
          "product"
        ]
      }
    }
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  }
}
```

Table 2-171 Parameters for hybrid aggregation of low- and high-cardinality fields

Parameter	Type	Default Value	Description
index.search.turbo.enabled	Boolean	false	Whether to enable enhanced aggregation. Normally, enhanced aggregation must be enabled where aggregations are used. The value can be: • **false**: Disable enhanced aggregation. • **true**: Enable enhanced aggregation.
index.sort.field	Array of strings	N/A	Specify sorting keys. Sorting keys are fields used to sequence or rank documents. You can specify one or multiple fields as sorting keys. When multiple fields are specified, they will apply in the sequence in which they are specified. Documents are first ranked by the first field, then the initial result set is ranked by the second field, and so on. Constraint: High-cardinality fields must be among the sorting keys, and must follow the last low-cardinality field. Value range: The value must be fields contained in the index.
index.cluster.field	Array of strings	N/A	Specify clustering keys. Clustering keys determine which documents are collected into the same clusters. During an aggregation operation, documents in the same cluster can be processed in batches, significantly enhancing aggregation performance. Constraint: The clustering keys must be a subset of the sorting keys. Value range: The value must be fields contained in the index.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. Run the following command to import sample data.

```
PUT /_bulk
{ "index": { "_index": "testindex", "_id": "1" } }
{ "date": "2025-01-01", "city": "cityA", "product": "books", "trade": 3000.0}
{ "index": { "_index": "testindex", "_id": "2" } }
{ "date": "2025-01-02", "city": "cityA", "product": "books", "trade": 1000.0}
{ "index": { "_index": "testindex", "_id": "3" } }
{ "date": "2025-01-01", "city": "cityA", "product": "bottles", "trade": 100.0}
{ "index": { "_index": "testindex", "_id": "4" } }
{ "date": "2025-01-02", "city": "cityA", "product": "bottles", "trade": 300.0}
{ "index": { "_index": "testindex", "_id": "5" } }
{ "date": "2025-01-01", "city": "cityB", "product": "books", "trade": 7000.0}
{ "index": { "_index": "testindex", "_id": "6" } }
{ "date": "2025-01-02", "city": "cityB", "product": "books", "trade": 1000.0}
```

3. Run the following query command to perform a hybrid aggregation of low- and high-cardinality fields.

   For example, query the average **trade** value of each product in different cities on each day specified by the **date** field.

```
POST testindex/_search
{
  "size": 0,
  "aggs": {
    "groupby_region": {
      "terms": {
        "field": "city"
      },
      "aggs": {
        "groupby_host": {
          "terms": {
            "field": "product"
          },
          "aggs": {
            "groupby_timestamp": {
              "date_histogram": {
                "field": "date",
                "interval": "day"
              },
              "aggs": {
                "avg_score": {
                  "avg": {
                    "field": "trade"
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

Performance Testing

Test environment

- Dataset: esrally nyc_taxis
- Cluster specifications: 4U16G, 100 GB, high I/O x 3 nodes

Test Procedure

1. Create an index template in the cluster, specify sorting keys, and disable enhanced aggregation.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
PUT /_template/nyc_taxis
{
  "template": "nyc_taxis*",
  "settings": {
    "index.search.turbo.enabled": false,
    "index.sort.field": "dropoff_datetime",
    "number_of_shards": 3,
    "number_of_replicas": 0
  }
}
```

2. Use esrally to test the nyc_taxis dataset and obtain the result when enhanced aggregation is disabled.
3. Create another index template in the same cluster, specify sorting keys, and enable enhanced aggregation.

```
PUT /_template/nyc_taxis
{
  "template": "nyc_taxis*",
  "settings": {
    "index.search.turbo.enabled": true,
    "index.sort.field": "dropoff_datetime",
    "number_of_shards": 3,
    "number_of_replicas": 0
  }
}
```

4. Use esrally to test the nyc_taxis dataset and obtain the result when enhanced aggregation is enabled.

Test Result

This test focuses on the query result of **dropoff_datetime** aggregation, that is, the results of tasks **autohisto_agg** and **date_histogram_agg**. The following table compares the test results between when enhanced aggregation is disabled and when it is enabled.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric	Task	Unit	Enhanced Aggregation Disabled			Enhanced Aggregation Enabled			Enhanced Aggregation Disabled	Enhanced Aggregation Enabled	open/close	Conclusion
			Test Round 1	Test Round 2	Test Round 3	Test Round 1	Test Round 2	Test Round 3	Mean Value	Mean Value		
Min Throughput	autohisto_agg	ops/s	4.42	4.44	4.43	11.66	11.94	11.96	4.43	11.85	2.68	Throughput improves more than 2.5 times.
Mean Throughput	autohisto_agg	ops/s	4.50	4.46	4.44	11.81	11.99	12.00	4.47	11.93	2.67	
Median Throughput	autohisto_agg	ops/s	4.51	4.46	4.44	11.83	11.98	12.00	4.47	11.94	2.67	
Max Throughput	autohisto_agg	ops/s	4.54	4.48	4.45	11.90	12.07	12.02	4.49	12.00	2.67	
100th percentile latency	autohisto_agg	ms	216.30	-	-	-	84.56	80.38	216.30	82.47	0.38	Latency decreases by more than 60%.
100th percentile service time	autohisto_agg	ms	216.30	-	-	-	84.56	80.38	216.30	82.47	0.38	
error rate	autohisto_agg	%	0	0	0	0	0	0	0	0	0	-

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric	Task	Unit	Enhanced Aggregation Disabled			Enhanced Aggregation Enabled			Enhanced Aggregation Disabled	Enhanced Aggregation Enabled	open/close	Conclusion
			Test Round 1	Test Round 2	Test Round 3	Test Round 1	Test Round 2	Test Round 3	Mean Value	Mean Value		
Min Throughput	date_histogram_agg	ops/s	4.72	4.67	4.65	12.57	12.40	12.59	4.68	12.52	2.68	Throughput improves more than 2.5 times.
Mean Throughput	date_histogram_agg	ops/s	4.73	4.67	4.67	12.61	12.46	12.61	4.69	12.56	2.68	
Median Throughput	date_histogram_agg	ops/s	4.73	4.67	4.67	12.62	12.46	12.60	4.69	12.56	2.68	
Max Throughput	date_histogram_agg	ops/s	4.74	4.67	4.67	12.64	12.49	12.63	4.69	12.59	2.68	
50th percentile latency	date_histogram_agg	ms	202.61	218.09	213.43	77.64	76.02	82.63	211.38	78.77	0.37	Latency decreases by more than 60%.
100th percentile latency	date_histogram_agg	ms	207.35	223.88	246.63	77.99	-	-	225.95	77.99	0.35	

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric	Task	Unit	Enhanced Aggregation Disabled			Enhanced Aggregation Enabled			Enhanced Aggregation Disabled	Enhanced Aggregation Enabled	open/close	Conclusion
			Test Round 1	Test Round 2	Test Round 3	Test Round 1	Test Round 2	Test Round 3	Mean Value	Mean Value		
50th percentile service time	date_histogram_agg	ms	202.61	218.09	213.43	77.64	76.02	82.63	211.38	78.77	0.37	
100th percentile service time	date_histogram_agg	ms	207.35	223.88	246.63	77.99	-	-	225.95	77.99	0.35	
error rate	date_histogram_agg	%	0	0	0	0	0	0	0	0	0	-

Test Conclusion

Given the same cluster configuration, aggregation performance improves significantly when enhanced aggregation is enabled. Query throughput improves by more than 2.5 times, and latency decreases by more than 60%.

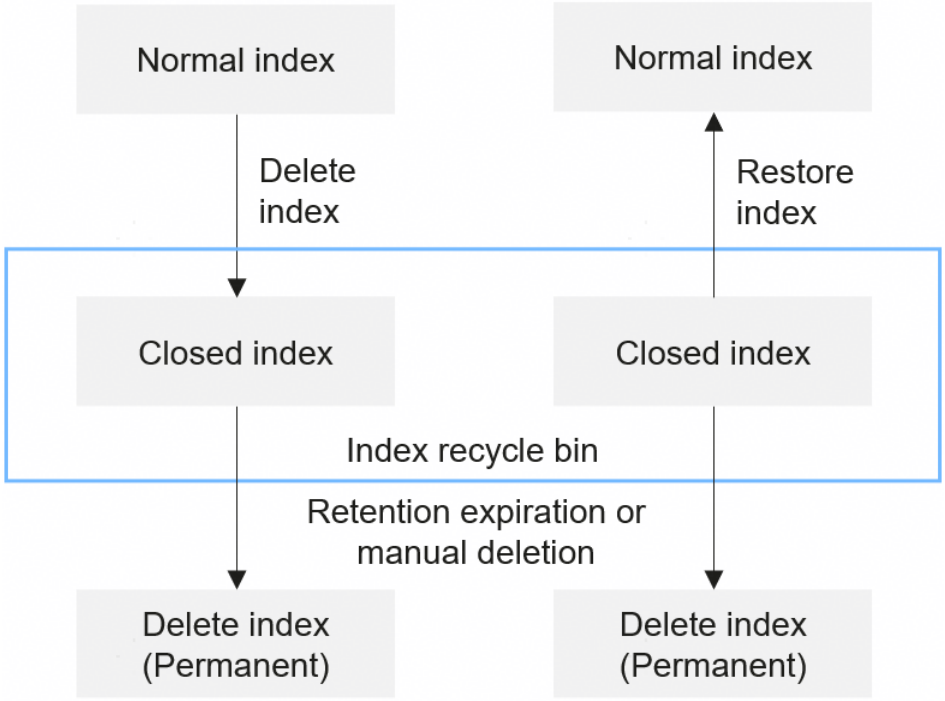
2.11.13 Enabling the Index Recycle Bin

By default, when indexes are deleted in an Elasticsearch cluster, the index data is deleted permanently. To prevent data loss caused by accidental deletion, CSS provides an index recycle bin. When enabled, deleted indexes are temporarily stored in the recycle bin, allowing for recovery before they are permanently removed. This feature improves data reliability and operational security.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

How the Feature Works

Figure 2-45 How the index recycle bin works



The index recycle bin prevents permanent data loss caused by accidental index deletion. As shown in **Figure 2-45**, when the index recycle bin is enabled:

- When you delete an index, the index status changes to Closed, and the closed index is moved to the recycle bin. This may cause the cluster status to briefly turn Red. This status indicates there are unallocated shards in the cluster, but it does not affect service availability.
- Deleted indexes are retained in the recycle bin for a predefined period. When this retention period expires, the deleted index and its data are removed from the recycle bin and are thus permanently deleted. You can also manually delete them from the index recycle bin at any time, which is also permanent.
- To restore a deleted index, select it in the recycle bin, and click Restore. The index becomes Open again, and its shards are reinitialized. This process can also cause the cluster status to become Red temporarily.

Constraints

- Only Elasticsearch 7.10.2 supports the index recycle bin.
- Indexes in the recycle bin remain part of the cluster metadata. This is why you cannot create indexes with the same name as those in the recycle bin.

Logging In to Kibana

Log in to Kibana and go to the command execution page. Elasticsearch clusters support multiple access methods. This topic uses Kibana as an example to describe the operation procedures.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Enabling the Index Recycle Bin

1. Run the following command to enable the index recycle bin:

```
PUT _cluster/settings
{
  "persistent": {
    "index.trash.enabled": true
  }
}
```

Table 2-172 Parameters for enabling the index recycle bin

Parameter	Type	Default Value	Description
index.trash.enabled	Boolean	false	Whether to enable the index recycle bin. When the index recycle bin is enabled, deleting indexes will first move them to the recycle bin. The indexes will be permanently deleted only when they are cleared from the recycle bin. The value can be: • **true**: Enable the index recycle bin. • **false**: Disable the index recycle bin.
indices.trash.keep.time	String	1d	Index retention period in the recycle bin. Expired indexes are permanently removed via a scheduled deletion task. Value format: number + unit • Number: a natural number • Unit: nanos (nanosecond), micros (microsecond), ms (millisecond), s (second), m (minute), h (hour), or d (day) Minimum value: **1d**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
indices.trash.clean.interval	String	5m	Recycle bin cleanup interval. The system runs a periodic cleanup job to permanently remove expired indexes in the recycle bin. Value format: number + unit • Number: a natural number • Unit: nanos (nanosecond), micros (microsecond), ms (millisecond), s (second), m (minute), h (hour), or d (day) Minimum value: 5m

2. To manually delete specified indexes from the index recycle bin right away, instead of waiting for them to be removed periodically, run the following command:

```
DELETE {index_name}
```

Table 2-173 Parameter description

Parameter	Type	Default Value	Description
index_name	String	N/A	Specifies one or more indexes. • Single index: Enter the index name, for example, **my_index**. • Multiple indexes: Enter multiple index names and use a comma (,) to separate them, for example, **my_index1,my_index2**. • Wildcard: Use the wildcard (*) to match multiple indexes. For example, **myindex*** indicates all indexes whose name starts with **myindex**.

Viewing Indexes in the Recycle Bin

Run the following command to view indexes in the index recycle bin to confirm index status:

```
GET _cat/trash?v=true&s=index
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-174 Request parameters

Parameter	Type	Default Value	Description
v	Boolean	false	Whether the returned table will have a header. The value can be: • **true**: display the table header. • **false**: not to display the table header.
s	String	N/A	Index sorting field. The value can be: • **index**: Sorts indexes by index name. • **uuid**: Sorts indexes by index UUID. • **pri**: Sorts indexes by their number of primary shards. • **rep**: Sorts indexes by their number of replica shards. • **trash.ts**: Sorts indexes by the timestamp of when they were moved to the recycle bin. • **delete.time**: Sorts indexes by their remaining retention period in the recycle bin.
format	String	None specified (table format)	The format of the returned result. Value range: json (text), yaml (text), cbor (binary), or smile (binary) If none of these four formats is specified, the result will be returned in a table.
h	String	None specified (all)	Column names to be displayed in the returned table. If you wish to display only some of the columns, specify their names and separate them with a comma (,), for example, **h=index,uuid,delete.time**.

The following is an example of the output.

```
index    uuid                   pri rep      trash.ts delete.time
index1   CMD3FCLzTOyTg4RUekWNNA   1   1 1714465116615       23.6h
index1   6ATijuu6SfqamVI-WMyOKg   1   1 1714466233898       23.9h
```

NOTE

When **delete.time** becomes 0, an index will be permanently deleted.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Restoring an Index from the Recycle Bin

1. Run the following command to restore an index from the recycle bin:
   ```
   POST /trash/recover/{index_name}
   ```
2. Run the following command to query the index list and check whether the index has been successfully restored:
   ```
   GET _cat/indices?v=true
   ```

Emptying the Recycle Bin

WARNING

Emptying the recycle bin permanently deletes its data. Before doing that, make sure important data has been backed up.

1. Run the following command to empty the recycle bin:
   ```
   POST trash/empty
   ```
2. Run the following command to confirm that the recycle bin has been emptied:
   ```
   GET _cat/trash?v=true&s=index
   ```

2.11.14 Tracking Query Resource Consumption

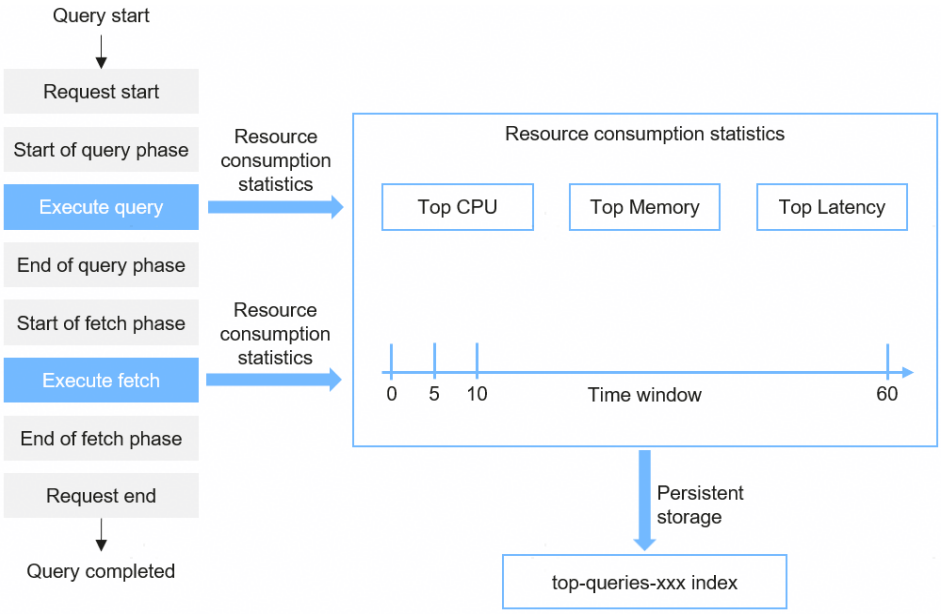
During routine maintenance, O&M engineers may need to identify and analyze top (resource-consuming) queries that are causing high resource consumption or performance issues in Elasticsearch clusters. Typically, this requires calling Elasticsearch APIs to retrieve a list of ongoing tasks and examining hot threads to determine which queries are causing excessive resource usage, such as high CPU consumption. The process can be complex and time-consuming. To improve O&M efficiency, CSS provides a query resource tracker. With this feature, O&M personnel can call an API to obtain top queries with the highest latency, CPU usage, or memory consumption, filter the queries by time range, and quickly identify problematic queries. This can significantly improve troubleshooting efficiency and accuracy.

How the Feature Works

The query resource tracker helps identify and optimize top resource-consuming queries, improving system performance and resource utilization. It is a useful tool in big data analytics and log processing scenarios.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 2-46 How the query resource tracker works



During query execution, the system records the resource consumption of each sub-phase (such as Query, Fetch, and Scroll), tracking metrics like CPU time and memory usage. The data is aggregated by a fixed time window (for example, 5 minutes). Queries with the highest resource consumption are recorded in a dedicated top queries index for analysis. A new top queries index is created daily to store these query resource statistics. The naming format is top-queries-xxx (where xxx indicates the date).

The top queries are ranked by the metric you specify, which can be CPU usage, memory usage, or latency. By default, the system ranks queries by latency.

Constraints

- The query resource tracker adds memory cache fields, which may impact cluster performance.
- The query resource tracker is supported only when the Elasticsearch cluster version is 7.10.2 and the image version is 7.10.2_25.9.0_xxx or later. It is **enabled by default** for clusters that meet these version requirements.

Modifying Top Queries Monitoring Settings

Run the following command to modify the top queries monitoring settings as needed:

```
PUT _cluster/settings
{
  "persistent": {
    "search.insights.top_queries.cpu.enabled": true,
    "search.insights.top_queries.cpu.window_size": "10m",
    "search.insights.top_queries.cpu.top_n_size": 20,
    "search.insights.top_queries.exporter.delete_after_days": 8,
    "search.insights.top_queries.group_by": "none"
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-175 Parameter description

Parameter	Type	Default Value	Description
search.insights.top_queries.<metric>.enabled	Boolean	true	Whether to enable top query monitoring by the specified metric. Supported metrics: **latency**, **cpu** (CPU usage), or **memory** (memory usage). The value can be: • **true**: Enable top query monitoring. • **false**: Disable top query monitoring.
search.insights.top_queries.<metric>.window_size	String	5m	The size of the observation window. Monitoring data is aggregated and computed by a fixed window (for example, 5 minutes). Supported metrics: **latency**, **cpu** (CPU usage), or **memory** (memory usage). Value range: 1m (1 minute), 5m (5 minutes), 10m (10 minutes), 30m (30 minutes), or *x*h (*x* hours, where *x* ranges from 1 to 24)
search.insights.top_queries.<metric>.top_n_size	Integer	10	Number of top N queries monitored in each time window. For example, if this parameter is set to 20, only the top 20 queries are monitored in each time window. Supported metrics: **latency**, **cpu** (CPU usage), or **memory** (memory usage). Value range: 1 to 100
search.insights.top_queries.exporter.delete_after_days	Integer	7	Retention period of the top-queries-*xxx* index. For example, if this parameter is set to 8, the index is retained for eight days. Value range: 1 to 180 Unit: days

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
search.insights.top_queries.group_by	String	none	Whether to enable top query grouping. The value can be: • **none**: No grouping. • **similarity**: Group queries by feature similarity. Within each time window, only the first query is displayed for each group. For more information, see **Introduction to Query Grouping**.
search.insights.top_queries.grouping.attributes.field_name	Boolean	true	Whether to use query field names for query grouping. This item takes effect only when **search.insights.top_queries.group_by** is set to **similarity**. The value can be: • **true**: Group queries by field name. • **false**: Ignore query field names.
search.insights.top_queries.grouping.attributes.field_type	Boolean	true	Whether to use query field types for query grouping. This item takes effect only when **search.insights.top_queries.group_by** is set to **similarity**. The value can be: • **true**: Group queries by field type. • **false**: Ignore query field types.

Obtaining Top Queries

The following is an example of the command that you can run to obtain top queries by a specified metric and time range:

```
GET _insights/top_queries?type=cpu&from=2025-12-02T00:00:00.000Z&to=2025-12-02T17:00:00.000Z
```

Table 2-176 Request parameters

Parameter	Type	Default Value	Description
type	String	latency	The metric by which top queries are identified. Value range: **latency**, **cpu** (CPU usage), or **memory** (memory usage).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
from	String	Null (obtain the top N queries in the last two windows)	Start time of the query time range. **from** and **to** must both be configured. Value format: YYYY-MM-DDTHH:mm:ss.SSSZ (timestamp in ISO 8601 format)
to	String	Null (obtain the top N queries in the last two windows)	End time of the query time range. **from** and **to** must both be configured. Value format: YYYY-MM-DDTHH:mm:ss.SSSZ (timestamp in ISO 8601 format)

Example response:

```
{
  "top_queries": [
    {
      "timestamp": 1764662136273,                          //Timestamp of the query.
      "date": "2025-12-02 07:55:36Z",                      //Time when the query was executed.
      "id": "4a5b4b1e-b502-4621-a5c1-09b8d9a1b81c",        //Unique ID of the query.
      "task_resource_usages": [
        {
          "action": "indices:data/read/search[phase/query]", // CPU and memory consumption of the query phase on node FB2ixw4IQCuXzCR83GT5Yg
          "taskId": 1877927,
          "parentTaskId": 111295,
          "nodeId": "FB2ixw4IQCuXzCR83GT5Yg",
          "taskResourceUsage": {
            "cpu_time_in_nanos": 3017078,
            "memory_in_bytes": 102360
          }
        },
        {
          "action": "indices:data/read/search[phase/query]", // CPU and memory consumption of the query phase on node FB2ixw4IQCuXzCR83GT5Yg
          "taskId": 1877926,
          "parentTaskId": 111295,
          "nodeId": "FB2ixw4IQCuXzCR83GT5Yg",
          "taskResourceUsage": {
            "cpu_time_in_nanos": 5618940,
            "memory_in_bytes": 271680
          }
        },
        {
          "action": "indices:data/read/search[phase/query]", // CPU and memory consumption of the query phase on node 2ICOHICoSS26YeQu5PIrlg
          "taskId": 107710,
          "parentTaskId": 111295,
          "nodeId": "2ICOHICoSS26YeQu5PIrlg",
          "taskResourceUsage": {
            "cpu_time_in_nanos": 8703914,
            "memory_in_bytes": 501560
          }
        },
        {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        "action": "indices:data/read/search[phase/fetch/id]", // CPU and memory consumption of the fetch
phase on node FB2ixw4IQCuXzCR83GT5Yg
        "taskId": 1877928,
        "parentTaskId": 111295,
        "nodeId": "FB2ixw4IQCuXzCR83GT5Yg",
        "taskResourceUsage": {
          "cpu_time_in_nanos": 424055,
          "memory_in_bytes": 59000
        }
      },
      {
        "action": "indices:data/read/search[phase/fetch/id]", // CPU and memory consumption of the fetch
phase on node 2ICOHICoSS26YeQu5PIrlg
        "taskId": 107711,
        "parentTaskId": 111295,
        "nodeId": "2ICOHICoSS26YeQu5PIrlg",
        "taskResourceUsage": {
          "cpu_time_in_nanos": 1279677,
          "memory_in_bytes": 337504
        }
      },
      {
        "action": "indices:data/read/search",               // CPU and memory consumption on the access node
gzsjh_47SjCe6QFs9pKjEg during the query start phase
        "taskId": 111295,
        "parentTaskId": -1,
        "nodeId": "gzsjh_47SjCe6QFs9pKjEg",
        "taskResourceUsage": {
          "cpu_time_in_nanos": 297268,
          "memory_in_bytes": 8632
        }
      }
    ],
    "source": {                                     //Specific query statement
      "query": {
        "match": {
          "message": {
            "query": "http",
            "operator": "OR",
            "prefix_length": 0,
            "max_expansions": 50,
            "fuzzy_transpositions": true,
            "lenient": false,
            "zero_terms_query": "NONE",
            "auto_generate_synonyms_phrase_query": true,
            "boost": 1
          }
        }
      }
    },
    "indices": [                                    //Queried index
      "log1"
    ],
    "total_shards": 3,                              //Total number of shards queried
    "phase_latency_map": {                          //Time consumed during each phase
      "expand": 0,                                  //Time consumed in the expand phase
      "query": 16,                                  //Time consumed in the query phase
      "fetch": 3                                    //Time consumed in the fetch phase
    },
    "labels": {},
    "group_by": "NONE",                             //Query grouping type
    "node_id": "gzsjh_47SjCe6QFs9pKjEg",            //ID of the node that received the request
    "search_type": "query_then_fetch",              //Query type
    "measurements": {                               //Metrics used
      "memory": {                                   //Memory consumption
        "number": 1280736,
        "count": 1,
        "aggregationType": "NONE"
      },
```

```
      "latency": {                                  //Latency
        "number": 20,
        "count": 1,
        "aggregationType": "NONE"
      },
      "cpu": {                                      //CPU consumption
        "number": 19340932,
        "count": 1,
        "aggregationType": "NONE"
      }
    }
  }
 ]
}
```

Obtaining the Resource Consumption of the Query Resource Tracker

Run the following command to obtain the resource consumption of the query resource tracker:

```
GET _insights/health_stats
```

Example response:

```
{
  "QDyhJ8Q6Td2acc3KGQ43bQ" : {
    "ThreadPoolInfo" : {
      "query_insights_executor" : {  //Dedicated queue for the plugin, used to execute top query analysis tasks
        "type" : "scaling",
        "core" : 1,
        "max" : 1,
        "keep_alive" : "5m",
        "queue_size" : -1
      }
    },
    "QueryRecordsQueueSize" : 0,  //Number of unprocessed tasks in the query_insights_executor queue
    "TopQueriesHealthStats" : {
      "latency" : {
        "TopQueriesHeapSize" : 0,  //Memory occupied by top queries statistics
        "QueryGroupCount_Total" : 0,  //Number of groups kept in the memory when grouping is enabled
        "QueryGroupCount_MaxHeap" : 0  //Memory used to store groups when grouping is enabled
      },
      "cpu" : {
        "TopQueriesHeapSize" : 0,
        "QueryGroupCount_Total" : 0,
        "QueryGroupCount_MaxHeap" : 0
      },
      "memory" : {
        "TopQueriesHeapSize" : 0,
        "QueryGroupCount_Total" : 0,
        "QueryGroupCount_MaxHeap" : 0
      }
    },
    "FieldTypeCacheStats" : { //Cache statistics. When query grouping is enabled, the field mapping is cached
to avoid repeated mapping lookups.
      "size_in_bytes" : 0,
      "entry_count" : 0,
      "evictions" : 0,
      "hit_count" : 0,
      "miss_count" : 0
    }
  }
}
```

Introduction to Query Grouping

When a single query continuously consumes excessive resources, it can monopolize the TopN statistics, obscuring other resource-intensive queries. Query

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

grouping addresses this by aggregating similar queries through pattern matching, ensuring that only the first query from each group appears in the TopN results.

Table 2-177 Grouping modes

Grouping Mode	Description
Complete query structure (structure + field + type)	Precisely matches the field type.
Query structure only	Considers the query structure only, while ignoring field names and types.
Query structure + field only	Considers the query structure and field names only, while ignoring field types.
Query structure + type only	Considers the query structure and field types only, while ignoring field names.

NOTE

Choose field type matching if the field types in your database remain relatively constant. The system will cache field types to improve grouping efficiency.

Example

The mapping of an index is as follows:

```
"mappings": {
  "properties": {
    "field1": {
      "type": "keyword"
    },
    "field2": {
      "type": "text"
    },
    "field3": {
      "type": "text"
    },
    "field4": {
      "type": "long"
    }
  }
}
```

Perform the following query on the index:

```
{
  "query": {
    "bool": {
      "must": [
        {
          "term": {
            "field1": "example_value"
          }
        }
      ],
      "filter": [
        {
          "match": {
            "field2": "search_text"
          }
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
      },
      {
        "range": {
          "field4": {
            "gte": 1,
            "lte": 100
          }
        }
      }
    ],
    "should": [
      {
        "regexp": {
          "field3": ".*"
        }
      }
    ]
  }
 }
}
```

Table 2-178 provides examples that help you understand how each grouping mode behaves.

Table 2-178 Behavior of different grouping modes

Grouping Mode	Pattern
Complete query structure (structure + field + type)	bool [] must: term [field1, keyword] filter: match [field2, text] range [field4, long] should: regexp [field3, text]
Query structure only	bool must: term filter: match range should: regexp
Query structure + field only	bool [] must: term [field1] filter: match [field2] range [field4] should: regexp [field3]
Query structure + type only	bool [] must: term [keyword] filter: match [text] range [long] should: regexp [text]

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

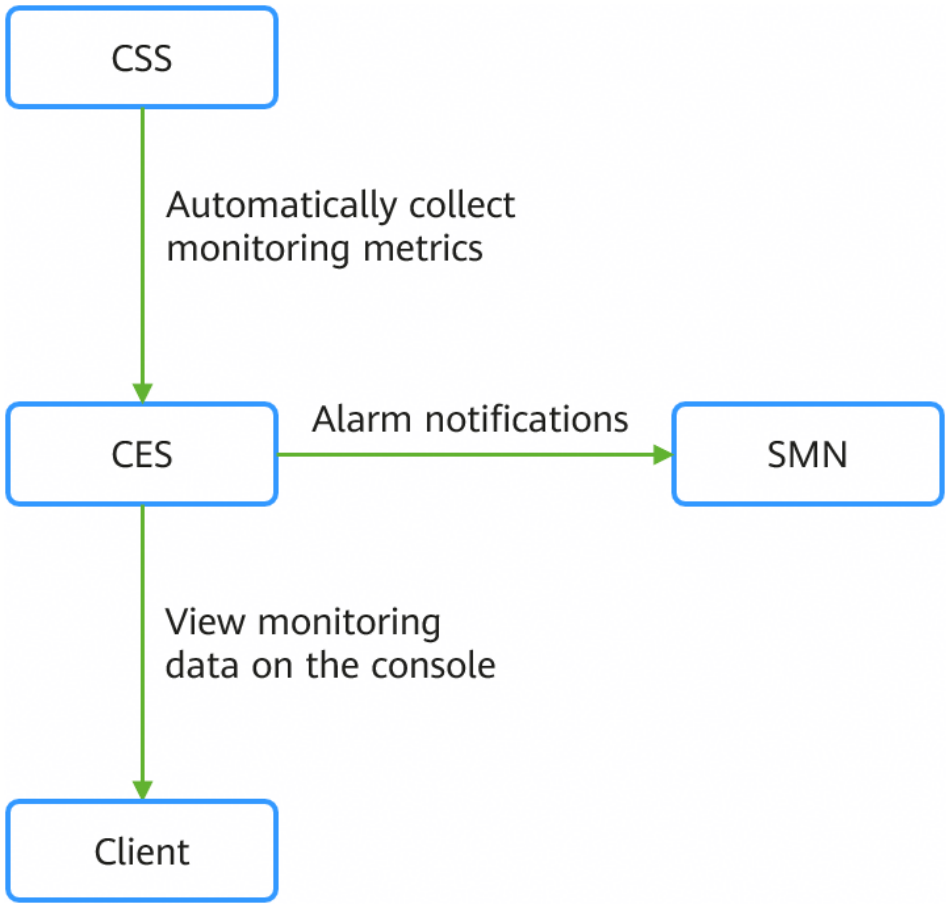
2.12 Monitoring

2.12.1 Using Cloud Eye to Monitor Clusters

In production environments, Elasticsearch clusters run continuously, and issues such as full disks, JVM out-of-memory errors, and query queue backlogs often emerge before causing failures. However, relying only on cluster logs makes it difficult to detect these issues early and analyze metric trends visually. To address this, CSS integrates with Cloud Eye to collect core Elasticsearch cluster metrics, such as disk usage, CPU usage, JVM heap usage, and read/write latency, in real time. Cloud Eye provides visual monitoring dashboards to display metric trends and sends alarm notifications through Simple Message Notification (SMN) when metrics exceed configured thresholds. This helps operations personnel shift from post-failure troubleshooting to proactive issue prevention.

How the Feature Works

Figure 2-47 Monitoring data flow



The monitoring data flow is as follows:

1. Clusters automatically report monitoring data to Cloud Eye.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. You can set alarm rules for CSS on Cloud Eye. You can configure alarm notifications via SMN.
3. You can then view monitoring metrics on the console.

Cloud Eye provides the following monitoring dimensions for Elasticsearch clusters.

Table 2-179 Monitoring dimensions

Dimension	Description	Applicable Scenario
Cluster	Aggregates metrics at the Elasticsearch cluster level, such as cluster health status and cluster-level disk usage.	Overall health checks and cluster-level alerting
Node	Collects resource metrics at the node level, such as the CPU usage and JVM heap usage of individual nodes.	Identifying hot nodes and troubleshooting node-level performance issues

Recommended Monitoring Metrics

Table 2-180 Recommended metrics

Dimension	Recommended Metrics	Description
Disk and cluster health (routine monitoring)	Disk usage, cluster health	Full disks are a primary cause of cluster unavailability. You are advised to set an alarm threshold of 85% for disk usage.
CPU and JVM	Average JVM heap usage, maximum JVM heap usage, average CPU usage, and maximum CPU usage	Continuously high JVM heap usage (> 75%) may cause frequent GC and affect query performance.
Read/Write latency and throughput	Average indexing latency, average indexing rate, average query latency, and average query rate	Continuously increasing indexing/query latency usually indicates excessive cluster load or the presence of slow queries.
Write/Search queues and rejected requests	Total queued tasks in the Write queue, total queued tasks in the Search queue, total rejected tasks in the Write queue, and total rejected tasks in the Search queue	Rejected requests indicate that the cluster is overloaded and requests are being rejected. You are advised to configure an alarm rule that triggers when the number of rejected requests exceeds 0.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Constraints

- Runtime duration: The cluster must run for more than 10 minutes before Cloud Eye can collect valid monitoring data.
- Operation permissions: The account used for operations must have the required Cloud Eye permissions. Otherwise, you cannot view monitoring information or configure alarm rules.

Prerequisites

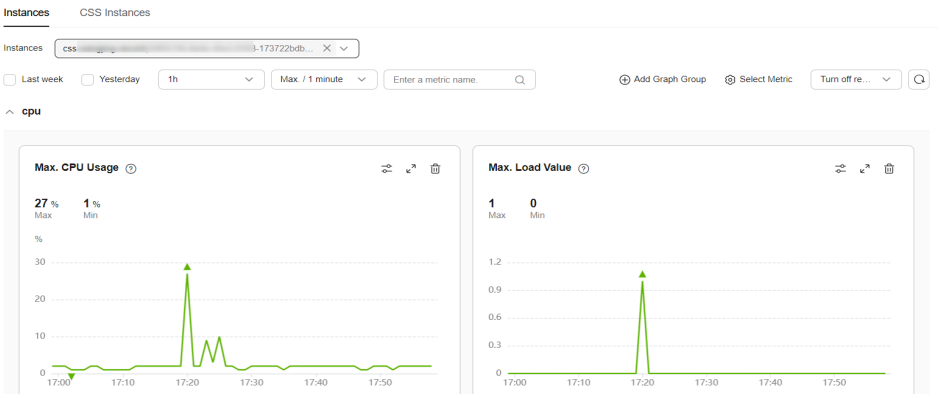
The target Elasticsearch cluster is available.

Viewing Monitoring Metrics

You can view the real-time monitoring metrics of clusters and nodes on the built-in monitoring page of the CSS console. You can compare data across multiple clusters and perform period-over-period analysis.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Monitoring Metrics** in the **Operation** column.
4. View cluster-level or node-level monitoring data as needed.

Figure 2-48 Monitoring Metrics



- **Viewing cluster-level monitoring metrics**

  On the monitoring metrics page, select the **Instances** tab.

  - **Instances**: Select the cluster whose monitoring data you want to view. Data for the current cluster is displayed by default. You can select multiple clusters for horizontal comparison.
  - You can set the monitoring time range and view period-over-period comparisons for the same cluster to identify periodic patterns or abnormal deviations.
  - **Create Graph Group**: Add custom groups on top of the default groups to group related metrics for focused analysis. When you move the pointer over a group name on the left, the icons for changing the

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

group name and deleting the group are displayed on the right. You can modify the group.

- **Select Metric**: In the **Select Metric** dialog box, you can select both **All Metrics** (single line charts) and **Top N Metrics** (comparison of the top N nodes by metric value).

– **Viewing node-level monitoring metrics**

On the monitoring metrics page, select the **CSS Instances** tab.

- **CSS Instances**: Select the cluster nodes whose monitoring data you want to view. Data for the first node in the current cluster is displayed by default. You can select multiple nodes for horizontal comparison.
- You can also set monitoring periods so that you can compare metrics between different periods for the same cluster.
- **Create Graph Group**: Add custom groups on top of the default groups to group related metrics for focused analysis. When you move the pointer over a group name on the left, the icons for changing the group name and deleting the group are displayed on the right. You can modify the group.
- Select Metric: In the displayed **Select Metric** dialog box, set **All Metrics** (per-metric charts).

5. (Optional) To view monitoring dashboards, see **Method 1: Monitoring Trends on the Monitoring Page (Monitoring Dashboards)**.

   Monitoring dashboards display time-series trend charts for multiple metrics in one place, allowing you to quickly understand the overall cluster status during routine checks.

Configuring Alarm Rules

You can configure alarm thresholds and notification methods for monitoring metrics. When metrics exceed configured thresholds, notifications are automatically sent to specified destinations, enabling operations personnel to shift from reactive troubleshooting to proactive monitoring.

1. Log in to the **Cloud Eye console**.
2. In the navigation pane on the left, choose **Alarm Management** > **Rule Settings**.
3. In the **Resource Type** column, search for alarm rules whose **Resource Type** is **Cloud Search Service**, and check whether these rules meet your requirements.

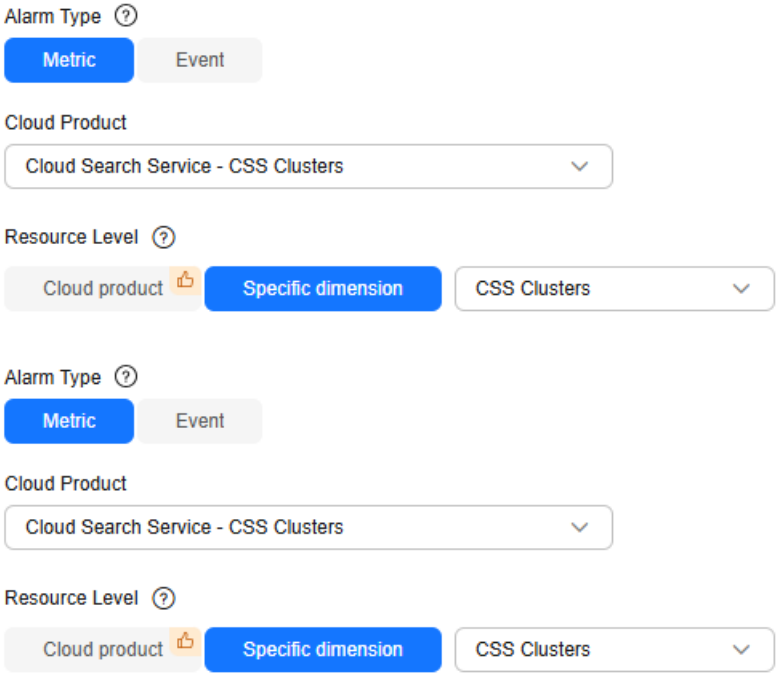
   To create new alarm rules, see **Creating an Alarm Rule and Notifications**. **Table 2-181** describes key parameters. Configure other parameters based on your requirements.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-181 Key parameters for configuring an alarm rule

Parameter	Description
Alarm Type	Select **Metric**.
Cloud Product	Select **Cloud Search Service - CSS Clusters**.
Resource Level	Select **Specific dimension**. In the drop-down list box on the right, select a specific dimension. • **CSS Clusters**: Creates an alarm rule at the cluster level. • **CSS Clusters - CSS Instances**: Creates an alarm rule at the individual node level.

Figure 2-49 Configuring a cluster/node-level alarm rule



4. After an alarm rule is created, it is displayed in the alarm rule list. The rule **Status** is **Enabled**.

2.12.2 Monitoring P99 Latency and HTTP Status Codes

During routine O&M or development, you may need to monitor the query performance and API health of your Elasticsearch clusters in real time. Open-source Elasticsearch only provides average latency metrics for this purpose, which cannot reflect the real impact of slow requests on user experience. Additionally, Elasticsearch lacks native statistics on HTTP request status codes, making it difficult to monitor anomalies such as sudden spikes in error rates or frequent rate limiting. To address these issues, CSS introduces P99 latency and HTTP status code monitoring to help you quantify search performance fluctuations, quickly detect

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

API exceptions, and provide data support for capacity planning and troubleshooting.

How the Feature Works

P99 latency (99th percentile latency) indicates that 99% of search requests have a response time less than or equal to this value within a specified statistical period. Unlike average latency, P99 latency accurately reflects the actual impact of slow queries on overall service quality. Even if the average latency appears normal, a small number of high-latency requests may still cause noticeable delays for many users. When the difference between P99 latency and the median latency (P50) is large, it usually indicates the presence of severe slow queries or resource contention in the cluster, requiring immediate investigation.

HTTP status code monitoring collects statistics on the HTTP status codes returned by Elasticsearch APIs (such as 200, 201, 400, 404, and 429) to help you understand the service status of the cluster. For example, a sudden increase in client errors (4xx) may indicate that upstream applications are sending invalid requests, while server errors (5xx) may indicate that cluster nodes are in an abnormal state. By continuously tracking status code distributions, you can quickly identify application-side issues and take prompt actions.

Constraints

- Cluster version requirements: Only Elasticsearch 7.6.2 and 7.10.2 support P99 latency and HTTP status code monitoring. The commands for querying HTTP status codes are different between the two versions. Use the appropriate command based on your cluster version.
- Data reset: After a cluster restart, all P99 latency statistics (overall, last_one_day, and latest) are cleared.
- Approximate calculation: P99 latency is an approximate value rather than an exact measurement. The accuracy is higher for percentiles closer to the boundary (such as P99 and P1).

Logging In to Kibana

Log in to Kibana and go to the command execution page. Elasticsearch clusters support multiple access methods. This topic uses Kibana as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Monitoring P99 Latency

- **Query the full percentile latency distribution.**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Obtain the complete percentile latency distribution of the current cluster to quickly determine whether slow queries exist.

Run the following command:

```
GET /search/stats/percentile
```

Example response:

```
{
  "overall" : {
    "1.0" : 2.0,
    "5.0" : 2.0,
    "25.0" : 6.5,
    "50.0" : 19.5,
    "75.0" : 111.0,
    "95.0" : 169.0,
    "99.0" : 169.0,
    "max" : 169.0,
    "min" : 2.0
  },
  "last_one_day" : {
    "1.0" : 2.0,
    "5.0" : 2.0,
    "25.0" : 6.5,
    "50.0" : 19.5,
    "75.0" : 111.0,
    "95.0" : 169.0,
    "99.0" : 169.0,
    "max" : 169.0,
    "min" : 2.0
  },
  "latest" : {
    "1.0" : 26.0,
    "5.0" : 26.0,
    "25.0" : 26.0,
    "50.0" : 26.0,
    "75.0" : 26.0,
    "95.0" : 26.0,
    "99.0" : 26.0,
    "max" : 26.0,
    "min" : 26.0
  }
}
```

In this example, P99 equals P50, indicating that there are no excessively slow queries.

Table 2-182 Response parameters

Parameter	Description
overall	Percentile latency distribution of all search requests since the cluster was started. It is used to create a historical overall performance baseline for the cluster. The unit is milliseconds (ms), indicating the request latency at each percentile.
last_one_day	Percentile latency distribution in the last 24 hours. It is used for daily inspection and recent performance monitoring. The unit is milliseconds (ms), indicating the request latency at each percentile.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
latest	Percentile latency distribution since the most recent **reset** operation. It is used to compare and verify performance before and after an O&M operation. The unit is milliseconds (ms), indicating the request latency at each percentile.

- **Query the latency in custom percentiles.**

  Specify the percentiles to query to reduce irrelevant data and focus on specific performance metrics.

  Run the following command to query latency data for the 1st, 50th, and 90th percentiles:

  ```
  GET /search/stats/percentile
  {
    "percents": [1, 50, 90]
  }
  ```

  The percents array supports any value between 0 and 100. You can specify the percentiles to query based on your service-level agreement (SLA) requirements. For example, if the SLA requires P95 latency to be less than 500 ms, you can focus on monitoring P95.

  Example response:

  ```
  {
    "overall" : {
      "1.0" : 0.0,
      "50.0" : 0.0,
      "90.0" : 0.0,
      "max" : 146.0,
      "min" : 0.0
    },
    "last_one_day" : {
      "1.0" : 0.0,
      "50.0" : 0.0,
      "90.0" : 0.0,
      "max" : 146.0,
      "min" : 0.0
    },
    "latest" : {
      "1.0" : 0.0,
      "50.0" : 0.0,
      "90.0" : 0.0,
      "max" : 146.0,
      "min" : 0.0
    }
  }
  ```

- **Reset the latest statistics.**

  Before and after a cluster change (such as scaling out or parameter adjustment), reset the latest statistics baseline to accurately compare performance differences.

  Run the following command:

  ```
  POST /search/stats/reset
  ```

  NOTE

  The reset operation only clears the **latest** statistics. The **overall** and **last_one_day** statistics are not affected.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Example response:

```
{
  "nodes" : {
    "css-c9c8-ess-esn-1-1" : "ok"
  }
}
```

If the status of each node is **ok**, the reset is successful.

Monitoring HTTP Status Codes

Obtain cumulative HTTP status code statistics for each cluster node to understand request success rates and abnormal distributions.

- **Elasticsearch 7.10.2**

  Run the following command:

  ```
  GET /_nodes/stats/http
  ```

  Example response:

  ```
  {
  ...
    "cluster_name" : "css-2985",
    "nodes" : {
  ...
      "omvR9_W-TsGApraMApREjA" : {
  ...
        "http" : {
          "current_open" : 4,
          "total_opened" : 37,
          "http_code" : {
            "200" : 25,
            "201" : 7,
            "429" : 0,
            "400" : 3,
            "404" : 0,
            "405" : 0
          },
          "http_worker_pending_tasks" : {
            "elasticsearch[css-xxx-ess-esn-1-1][transport_worker][T#1]" : 0,
            "elasticsearch[css-xxx-ess-esn-1-1][transport_worker][T#2]" : 0
          }
        }
      }
    }
  }
  ```

Table 2-183 Response parameters

Parameter	Description
current_open	Number of HTTP connections currently maintained by the node. A continuously high value (for example, greater than 1000) may indicate connection leaks on the client side.
total_opened	Total cumulative number of HTTP connections established by the node since startup. Comparing this value with **current_open** helps evaluate connection reuse efficiency.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
http_code	Cumulative number of requests for each HTTP status code since the node was started.
http_worker_pending_tasks	Number of pending tasks for HTTP worker threads. A continuously high number of pending tasks for each transport_worker thread may indicate request processing bottlenecks on the node.

- **Elasticsearch 7.6.2**

  Run the following command:

  ```
  GET /_nodes/http_stats
  ```

  Example response:

  ```
  {
    "_nodes" : {
     "total" : 1,
     "successful" : 1,
     "failed" : 0   },
    "cluster_name" : "css-8362",
    "nodes" : {
     "F9IFdQPARaOJI7oL7HOXtQ" : {
       "http_code" : {
         "200" : 114,
         "201" : 5,
         "429" : 0,
         "400" : 7,
         "404" : 0,
         "405" : 0
        }
       }
     }
  }
  ```

Table 2-184 Response parameters

Parameter	Description
http_code	Cumulative number of requests for each HTTP status code since the node was started.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

NOTE

Reference for evaluating cluster health in terms of HTTP status code distribution:

- 200/201 account for the vast majority (>95%): Normal.
- A small number of 400 errors (<1%): Acceptable. These usually indicate a small number of client-side syntax errors.
- A small number of 404 errors: Acceptable.
- 429 errors continuously increase: The cluster is overloaded and requests are being throttled. Recommended actions:
  - Check for large queries or full scans.
  - Consider scaling out data nodes.
  - Consider configuring index-level throttling policies.
- 400 errors increase significantly: Client code likely contains DSL syntax issues. Check the client code.
- 5xx errors occur: Indicates internal cluster errors. Check cluster logs and node status immediately.

Joint Monitoring of P99 Latency and HTTP Status Codes

P99 latency monitoring and HTTP status code monitoring can be used together to quickly locate performance issues. **Table 2-185** lists typical combinations of symptoms and possible causes.

Table 2-185 Joint monitoring of P99 latency and HTTP status codes

Symptom Combination	Possible Cause	Recommended Action
P99 latency increases + 429 errors increase	Insufficient cluster resources. Slow queries accumulate and trigger throttling.	Optimize slow query DSL statements and consider scaling out.
P99 latency increases + 200 OK	Complex queries or data skew exists.	Enable slow query logs to identify slow queries.
P99 latency is normal + 400 errors increase	The client code is problematic, but it does not yet affect successful requests.	Check client logs and fix DSL syntax errors.
P99 latency is normal + 429 errors occur	A traffic spike has occurred, but most requests are not affected.	Check whether traffic shaping or a scale-out is required.

2.12.3 Configuring Index-Level Monitoring

When maintaining large-scale Elasticsearch clusters, operations teams frequently encounter a critical visibility gap: while cluster-wide metrics (such as CPU and memory) indicate a healthy state, specific workloads or services experience long read/write latency. Traditional cluster-level monitoring often relies on aggregated

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

averages and fails to identify the specific indexes causing the problem. As a result, root-cause analysis is often time-consuming and labor-intensive. CSS supports index-level monitoring. It automatically collects real-time read/write traffic, latency, and storage changes for each index, and visualizes the data using a built-in Kibana dashboard. This helps you quickly identify abnormal indexes without writing any complex script. You can then make necessary adjustments to optimize performance and ensure service stability.

How the Feature Works

Index monitoring tracks the status and performance of each index in an Elasticsearch cluster in real time, helping operations teams promptly detect and resolve performance issues. The feature works as follows:

1. Collection: A background task periodically (every 10 seconds by default, configurable) collects statistics about each target index, including the number of documents, storage size, and shard status.
2. Storage: The collected data is written to a dedicated monitoring index named **monitoring-eye-css-[yyyy-mm-dd]**.
3. Visualization: Kibana reads data from the monitoring index and visualizes it through preset dashboards or custom visualizations, enabling time series analysis, trend comparison, and fault detection.

Constraints

- Only Elasticsearch 7.6.2 and 7.10.2 support index monitoring.
- Do not create regular indexes with their names starting with **monitoring-eye-css-***. Doing so may interfere with the collection of index monitoring data.
- Do not delete the **monitoring-eye-css-[yyyy-mm-dd]** index or its associated index pattern. Otherwise, index monitoring data will be unavailable.

Logging In to Kibana

Log in to Kibana and go to the command execution page. Elasticsearch clusters support multiple access methods. This topic uses Kibana as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Enabling Index Monitoring

- Run the following command to enable monitoring for all indexes in the current cluster:

```
PUT _cluster/settings
{
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  "persistent": {
    "css.monitoring.index.enabled": "true"
  }
}
```

- Run the following command to enable monitoring for specified high-priority indexes:

```
PUT _cluster/settings
{
  "persistent": {
    "css.monitoring.index.enabled": "true",
    "css.monitoring.index.interval": "30s",
    "css.monitoring.index.indices": ["index_name"],
    "css.monitoring.history.duration": "3d"
  }
}
```

Table 2-186 Parameters for enabling index monitoring

Parameter	Type	Default Value	Description
css.monitoring.index.enabled	Boolean	false	To enable or disable index monitoring. The value can be: • **true**: To enable index monitoring. • **false**: To disable index monitoring.
css.monitoring.index.indices	String	* (indicating that all indexes will be monitored)	The names of the indexes you want to monitor. • Single index: Enter the index name, for example, **my_index**. • Multiple indexes: Enter multiple index names and use a comma (,) to separate them, for example, **my_index1,my_index2**. • Wildcard: Use the wildcard (*) to match multiple indexes. For example, **myindex*** matches all indexes whose name starts with **myindex**. By default, all indexes are monitored. If you have a large number of indexes, consider monitoring only high-priority ones.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
css.monitoring.index.interval	Time	10s	Index monitoring data collection interval. Value format: number + unit • Number: a natural number • Unit: nanos (nanosecond), micros (microsecond), ms (millisecond), s (second), m (minute), h (hour), or d (day) Minimum value: **1s** Set an appropriate data collection interval based on your monitoring needs and cluster load. Collecting monitoring data too frequently can impact performance.
css.monitoring.history.duration	Time	7d	Retention duration of index monitoring data, that is, the data in the **monitoring-eye-css-[yyyy-mm-dd]** index. Upon expiration, data will be deleted automatically and permanently. Value format: number + unit • Number: a natural number • Unit: nanos (nanosecond), micros (microsecond), ms (millisecond), s (second), m (minute), h (hour), or d (day) Minimum value: **1d** Set an appropriate data retention duration to balance your monitoring needs with storage costs. A longer retention duration increases storage costs.

Checking the Read and Write Traffic of Indexes

When index monitoring is enabled for a cluster, you can use an API to query the real-time read and write traffic of indexes.

CAUTION

System indexes (those starting with **.**) contain internal management and maintenance information. To ensure system security, their read and write traffic cannot be queried.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- Run the following command to query the real-time read and write traffic information of all indexes in the cluster:
  ```
  GET /_cat/monitoring
  ```
- Run the following command to query the real-time read and write traffic of specified indexes in the cluster:
  ```
  GET /_cat/monitoring/{index_name}
  ```
- Run the following command to query the read/write traffic of indexes in a specified period:
  ```
  GET _cat/monitoring?begin=1650099461000
  GET _cat/monitoring?begin=2022-04-16T08:57:41
  GET _cat/monitoring?begin=2022-04-16T08:57:41&end=2022-04-17T08:57:41
  ```

Table 2-187 Parameters for querying the read and write traffic of indexes

Parameter	Type	Default Value	Description
index_name	String	N/A	Specifies one or more indexes. • Single index: Enter the index name, for example, **my_index**. • Multiple indexes: Enter multiple index names and use a comma (,) to separate them, for example, **my_index1,my_index2**. • Wildcard: Use the wildcard (*) to match multiple indexes. For example, **myindex*** matches all indexes whose name starts with **myindex**.
begin	String	Last 5 minutes	Start time of the monitoring period. Use the UTC time. Supported formats: • Standard date and time format: yyyy-MM-dd or yyyy-MM-ddTHH:mm:ss • A timestamp in milliseconds
end	String	Current time	End time of the monitoring period. Use the UTC time. Supported formats: • Standard date and time format: yyyy-MM-dd or yyyy-MM-ddTHH:mm:ss • A timestamp in milliseconds

Example response:

```
index   begin                   end                     status pri rep init unassign docs.count docs.deleted store.size
pri.store.size delete.rate indexing.rate search.rate
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
test 2022-03-25T09:46:53.765Z 2022-03-25T09:51:43.767Z yellow  1   1  0    1     9        0      5.9kb
5.9kb         0/s           0/s         0/s
```

Table 2-188 Response parameters

Parameter	Description
index	Index name.
begin	Start time of the monitoring data you queried.
end	End time of the monitoring data you queried.
status	Index status within the queried period.
pri	The number of primary shards within the queried period.
rep	The number of replica shards within the queried period.
init	The number of initialized indexes within the queried period.
unassign	The number of unallocated indexes within the queried period.
docs.count	The number of documents within the queried period.
docs.deleted	The number of deleted documents within the queried period.
store.size	Index storage size within the queried period.
pri.store.size	Size of a primary index shard within the queried period.
delete.rate	Number of indexes deleted per second within the queried period.
indexing.rate	Number of indexes written per second within the queried period.
search.rate	Number of indexes queried per second within the queried period.

Visualizing and Analyzing Index Monitoring Data in Kibana

Kibana enables efficient analysis of index monitoring data, including time series analysis, trend comparison, and fault detection.

CSS provides a pre-built Kibana dashboard that allows you to quickly view index monitoring information. You can also create custom visualizations in Kibana for more flexible monitoring.

- Check index monitoring results in the pre-built dashboard.
  a. In the Kibana console, click the menu button in the upper-left corner, and click **Dashboard**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

b. Find and click **[Monitoring] Index monitoring Dashboard** to view the index monitoring results.

The pre-built dashboard displays the number of read and write operations per second in the cluster, as well as the top 10 indexes by read and write requests.

Figure 2-50 Pre-built dashboard

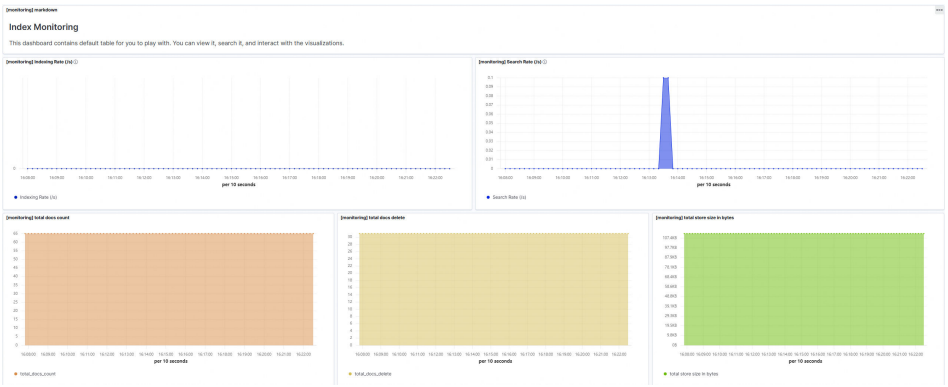


Table 2-189 Charts on the pre-built dashboard

Chart Name	Description
[monitoring] markdown	Markdown chart, which briefly describes the dashboard content.
[monitoring] Indexing Rate (/s)	Number of documents written to the cluster per second.
[monitoring] Search Rate (/s)	Average number of queries per second in the cluster.
[monitoring] indexing rate of index for top10	Top 10 indexes with the most documents written per second.
[monitoring] search rate of index for top10	Top 10 indexes with the most queries per second.
[monitoring] total docs count	Total number of documents in the cluster.
[monitoring] total docs delete	Total number of deleted documents in the cluster.
[monitoring] total store size in bytes	Total storage space occupied by documents in the cluster.
[monitoring] indices store_size for top10	Top 10 indexes that occupy the largest storage space.
[monitoring] indices docs_count for top10	Top 10 indexes that store the largest number of documents.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Chart Name	Description
[monitoring] indexing time in millis of index for top10(ms)	Top 10 indexes with the longest document write latency per time unit (ms).
[monitoring] search query time in millis of index for top10(ms)	Top 10 indexes with the longest index query time per time unit (ms).
[monitoring] segment count of index for top10	Top 10 indexes with the largest number of index segments.
[monitoring] segment memory in bytes of index for top10	Top 10 indexes with the largest heap memory usage of index segments.

- Check index monitoring results in custom visualizations.

  The following procedure describes how to create a custom visualization to monitor the changes in the document counts of indexes.

  a. Click **Visualize** in the upper-left menu of the Kibana console.

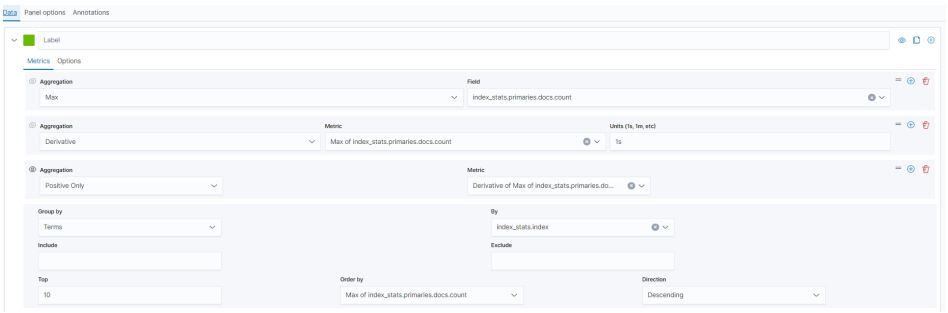
  b. Click **Create visualization**, and select **TSVB**.

  c. Set the chart parameters.

  On the **Data** tab shown in **Figure 2-51**, set the parameters as needed.

  - Select **Max** for **Aggregation**, and select **index_stats.primaries.docs.count** in **Field**, indicating the number of documents in a primary shard.
  - Select **Derivative** from **Aggregation** to indicate differences between aggregation buckets. Set **Units** to **1s** to visualize network rates as "per second".
  - Set **Aggregation** to **Positive Only** to prevent negative numbers after resetting.
  - To show statistics by index, set **Group by** to **Terms** and **By** to **index_stats.index**. Statistics will be grouped by index name.

  Figure 2-51 TSVB page

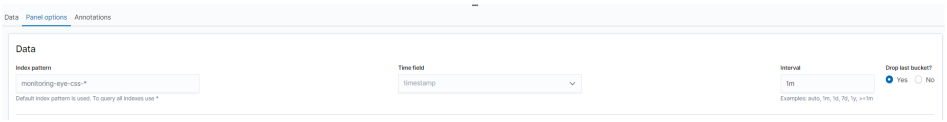
  

  To view data in different time segments, set the aggregation interval, or the displayed data will be incomplete. On the page shown in **Figure 2-52**,

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

select the **Panel options** tab, and set **Interval** to **1m** or **30m** to adjust the granularity of **timestamp**.

Figure 2-52 Setting the interval



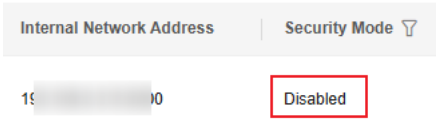
FAQ: What Do I Do If Index Monitoring Charts Are Not Displayed?

Symptom: The cluster's preset dashboards and visualizations are not displayed.

Solution:

1. In the cluster list, check whether the security mode is enabled for the cluster.

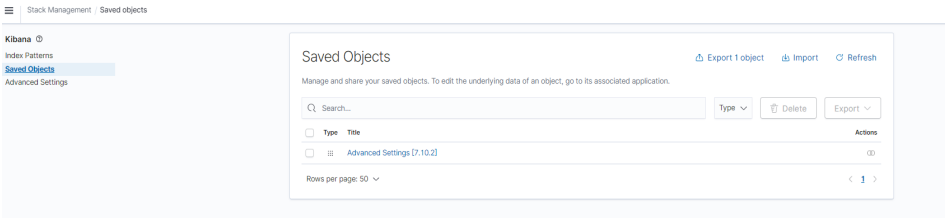
   Figure 2-53 Checking whether the security mode is enabled

   - If **Security Mode** is **Enabled**, switch to the Global space.

     On the Kibana page, click the username in the upper right corner and choose **Switch tenants**. On the **Select your tenant** page, switch the space and click **Confirm**. Then, check whether the charts are now displayed properly. If they are still not, go to the next step.
   - If **Security Mode** is **Disabled**, go to the next step.
2. Import the charts again.
   a. Create the **monitoring-kibana.ndjson** file by referring to **kibana-monitor Configuration File**.
   b. On the Kibana console, choose **Management > Stack Management > Saved objects**.

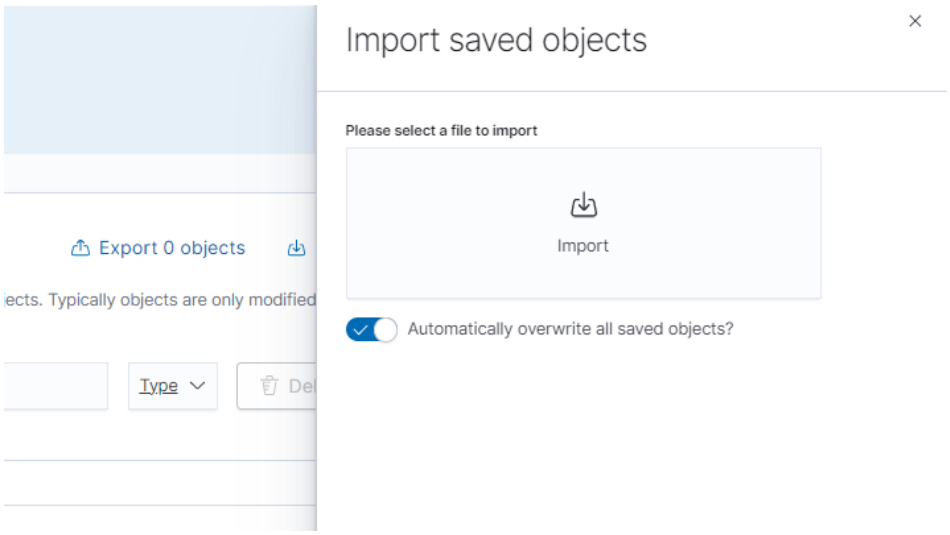
      Figure 2-54 Selecting Saved Objects

   c. Click **Import** and upload the created **monitoring-kibana.ndjson** file.

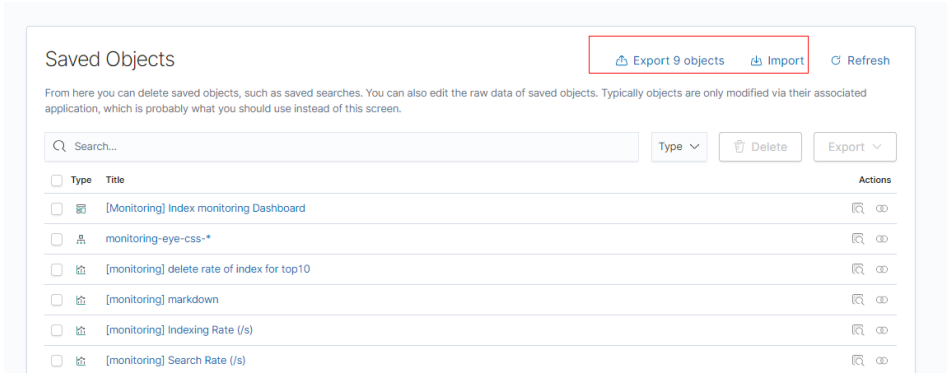
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 2-55 Uploading a file



d. After the file is uploaded, click **done**. The index monitoring charts are imported successfully.

Figure 2-56 Index monitoring charts imported successfully



If the problem persists, contact technical support.

kibana-monitor Configuration File

The kibana-monitor configuration files are different for Elasticsearch 7.6.2 and Elasticsearch 7.10.2. You are advised to use **monitoring-kibana.ndjson** as the file name.

- **Elasticsearch 7.10.2 cluster**

{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring] segment memory in bytes of index for top10","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] segment memory in bytes of index for top10\",\"type\":\"metrics\",\"aggs\":[],\"params\":{\"id\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"type\":\"timeseries\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-af02-69e470af7417\",\"color\":\"#68BC00\",\"split_mode\":\"terms\",\"split_color_mode\":\"kibana\",\"metrics\":[{\"id\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"max\",\"field\":\"index_stats.total.segments.memory_in_bytes\"}],\"separate_axis\":0,\"axis_position\":\"right\",\"formatter\":\"bytes\",\"chart_type\":\"line\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none\",\"label\":\"segments memory in bytes \",\"type\":\"timeseries\",\"terms_field\":\"index_stats.index\",\"terms_order_by\":\"61ca57f2-469d-11e7-af02-69e470af7417\"}],\"time_field\":\"timestamp\",\"index_pattern\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter\":\"number\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid\":1,\"tooltip_mode\":\"show_all\",\"default_index_pattern\":\"monitoring-eye-css-*

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

\",\"default_timefield\":\"timestamp\",\"isModelInvalid
\":false}}"},"id":"3ae5d820-6628-11ed-8cd7-973626cf6f70","migrationVersion":
{"visualization":"7.10.0"},"references":
[],"type":"visualization","updated_at":"2022-03-17T13:26:38.009Z","version":"WzY0NywxMF0="}
{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring] segment count of index for
top10","uiStateJSON":"{}","version":1,"visState":"{\"aggs\":[],\"params\":{\"axis_formatter\":\"number
\",\"axis_position\":\"left\",\"axis_scale\":\"normal\",\"default_index_pattern\":\"monitoring-eye-css-*
\",\"default_timefield\":\"timestamp\",\"filter\":{\"language\":\"kuery\",\"query\":\"\"},\"id
\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"index_pattern\":\"monitoring-eye-css-*\",\"interval
\":\"\",\"isModelInvalid\":false,\"series\":[{\"axis_position\":\"right\",\"chart_type\":\"line\",\"color
\":\"rgba(231,102,76,1)\",\"fill\":0.5,\"formatter\":\"number\",\"id\":\"61ca57f1-469d-11e7-
af02-69e470af7417\",\"label\":\"segment count of index for top10\",\"line_width\":1,\"metrics\":
[{\"field\":\"index_stats.total.segments.count\",\"id\":\"61ca57f2-469d-11e7-
af02-69e470af7417\",\"type\":\"max\"}],\"point_size\":1,\"separate_axis\":0,\"split_color_mode
\":\"kibana\",\"split_mode\":\"terms\",\"stacked\":\"none\",\"terms_field\":\"index_stats.index
\",\"terms_order_by\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"timeseries
\"}],\"show_grid\":1,\"show_legend\":1,\"time_field\":\"timestamp\",\"tooltip_mode\":\"show_all
\",\"type\":\"timeseries\"},\"title\":\"[monitoring] segment count of index for top10\",\"type\":\"metrics
\"}"},"id":"45d571c0-6626-11ed-8cd7-973626cf6f70","migrationVersion":
{"visualization":"7.10.0"},"references":
[],"type":"visualization","updated_at":"2022-03-17T13:26:38.009Z","version":"WzY0NiwxMF0="}
{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring] search query time in millis of index for
top10(ms)","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] search query time in
millis of index for top10(ms)\",\"type\":\"metrics\",\"aggs\":[],\"params\":{\"axis_formatter\":\"number
\",\"axis_max\":\"\",\"axis_min\":\"\",\"axis_position\":\"left\",\"axis_scale\":\"normal
\",\"default_index_pattern\":\"monitoring-eye-css-*\",\"default_timefield\":\"timestamp\",\"id
\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"index_pattern\":\"monitoring-eye-css-*\",\"interval
\":\"\",\"isModelInvalid\":false,\"series\":[{\"axis_position\":\"right\",\"chart_type\":\"line\",\"color
\":\"#68BC00\",\"fill\":0.5,\"formatter\":\"number\",\"id\":\"61ca57f1-469d-11e7-
af02-69e470af7417\",\"label\":\"index_query_time_in_millis\",\"line_width\":1,\"metrics\":[{\"field
\":\"index_stats.total.search.query_time_in_millis\",\"id\":\"61ca57f2-469d-11e7-
af02-69e470af7417\",\"type\":\"max\"},{\"unit\":\"1s\",\"id
\":\"42c92b10-6645-11ed-925a-6de90846447d\",\"type\":\"derivative\",\"field\":\"61ca57f2-469d-11e7-
af02-69e470af7417\"}],\"point_size\":1,\"separate_axis\":0,\"split_color_mode\":\"kibana\",\"split_mode
\":\"terms\",\"stacked\":\"none\",\"terms_field\":\"index_stats.index\",\"terms_order_by
\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"timeseries\"}],\"show_grid\":1,\"show_legend
\":1,\"time_field\":\"timestamp\",\"tooltip_mode\":\"show_all\",\"type\":\"timeseries
\",\"background_color\":null,\"filter\":{\"query\":\"\",\"language\":\"kuery\"},\"legend_position\":\"right
\"}}"},"id":"c8109100-6627-11ed-8cd7-973626cf6f70","migrationVersion":
{"visualization":"7.10.0"},"references":
[],"type":"visualization","updated_at":"2022-11-17T07:53:02.743Z","version":"WzY0NSwxMF0="}
{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring] indexing time in millis of index for
top10(ms)","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] indexing time in millis
of index for top10(ms)\",\"type\":\"metrics\",\"aggs\":[],\"params\":{\"id\":\"61ca57f0-469d-11e7-
af02-69e470af7417\",\"type\":\"timeseries\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-
af02-69e470af7417\",\"color\":\"rgba(218,139,69,1)\",\"split_mode\":\"terms\",\"split_color_mode
\":\"kibana\",\"metrics\":[{\"id\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"max\",\"field
\":\"index_stats.total.indexing.index_time_in_millis\"},{\"unit\":\"1s\",\"id
\":\"8099c0d0-6645-11ed-925a-6de90846447d\",\"type\":\"derivative\",\"field\":\"61ca57f2-469d-11e7-
af02-69e470af7417\"},{\"unit\":\"\",\"id\":\"e2c583e0-6652-11ed-b184-3911d8096b2b\",\"type
\":\"positive_only\",\"field\":\"8099c0d0-6645-11ed-925a-6de90846447d\"}],\"separate_axis
\":0,\"axis_position\":\"right\",\"formatter\":\"number\",\"chart_type\":\"line\",\"line_width
\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none\",\"label\":\"indexing index time in millis of index for
top 10\",\"type\":\"timeseries\",\"terms_field\":\"index_stats.index\",\"terms_order_by
\":\"61ca57f2-469d-11e7-af02-69e470af7417\"}],\"time_field\":\"timestamp\",\"index_pattern
\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter\":\"number
\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid\":1,\"tooltip_mode\":\"show_all
\",\"default_index_pattern\":\"monitoring-eye-css-*\",\"default_timefield\":\"timestamp
\",\"isModelInvalid\":false,\"background_color_rules\":[{\"id\":\"2930b170-662a-11ed-8056-
b7a911ede866\"}],\"bar_color_rules\":[{\"id\":\"2a1a0000-662a-11ed-8056-
b7a911ede866\"}],\"gauge_color_rules\":[{\"id\":\"2adeaf90-662a-11ed-8056-
b7a911ede866\"}],\"gauge_width\":10,\"gauge_inner_width\":10,\"gauge_style\":\"half\",\"axis_min
\":\"\"}}"},"id":"17d49220-662a-11ed-8cd7-973626cf6f70","migrationVersion":
{"visualization":"7.10.0"},"references":
[],"type":"visualization","updated_at":"2022-11-17T07:53:02.743Z","version":"WzY0NCwxMF0="}
{"attributes":{"description":"","kibanaSavedObjectMeta":

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

{"searchSourceJSON":"{}"},"title":"[monitoring] total store size in
bytes","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] total store size in bytes
\",\"type\":\"metrics\",\"aggs\":[],\"params\":{\"id\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"type
\":\"timeseries\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-af02-69e470af7417\",\"color
\":\"#68BC00\",\"split_mode\":\"everything\",\"split_color_mode\":\"kibana\",\"metrics\":[{\"id
\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"max\",\"field
\":\"indices_stats._all.total.store.size_in_bytes\"}],\"separate_axis\":0,\"axis_position\":\"right
\",\"formatter\":\"bytes\",\"chart_type\":\"line\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked
\":\"none\",\"label\":\"total store size in bytes\",\"type\":\"timeseries\"}],\"time_field\":\"timestamp
\",\"index_pattern\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter
\":\"number\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid\":1,\"tooltip_mode\":\"show_all
\",\"default_index_pattern\":\"monitoring-eye-css-*\",\"default_timefield\":\"timestamp
\",\"isModelInvalid\":false,\"legend_position\":\"bottom\",\"background_color_rules\":[{\"id
\":\"7712e550-664f-11ed-8b5d-8db37e5b4cc4\"}],\"bar_color_rules\":[{\"id
\":\"77680a30-664f-11ed-8b5d-8db37e5b4cc4\"}]}}"},"id":"c7f72ae0-664e-11ed-8cd7-973626cf6f70","
migrationVersion":{"visualization":"7.10.0"},"references":
[],"type":"visualization","updated_at":"2022-11-17T07:53:02.743Z","version":"WzY0MywxMF0="}
{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring] total docs
count","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] total docs count\",\"type
\":\"metrics\",\"aggs\":[],\"params\":{\"id\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"type
\":\"timeseries\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-af02-69e470af7417\",\"color
\":\"rgba(218,139,69,1)\",\"split_mode\":\"everything\",\"split_color_mode\":\"kibana\",\"metrics\":
[{\"unit\":\"\",\"id\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"max\",\"field
\":\"indices_stats._all.total.docs.count\"}],\"separate_axis\":0,\"axis_position\":\"right\",\"formatter
\":\"number\",\"chart_type\":\"line\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none
\",\"label\":\"total_docs_count\",\"type\":\"timeseries\"}],\"time_field\":\"timestamp\",\"index_pattern
\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter\":\"number
\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid\":1,\"tooltip_mode\":\"show_all
\",\"default_index_pattern\":\"monitoring-eye-css-*\",\"default_timefield\":\"timestamp
\",\"isModelInvalid\":false,\"legend_position\":\"bottom
\"}}"},"id":"eea89780-664b-11ed-8cd7-973626cf6f70","migrationVersion":
{"visualization":"7.10.0"},"references":
[],"type":"visualization","updated_at":"2022-11-17T07:53:02.743Z","version":"WzY0MiwxMF0="}
{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring] total docs
delete","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] total docs delete\",\"type
\":\"metrics\",\"aggs\":[],\"params\":{\"id\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"type
\":\"timeseries\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-af02-69e470af7417\",\"color
\":\"rgba(214,191,87,1)\",\"split_mode\":\"everything\",\"split_color_mode\":\"kibana\",\"metrics\":
[{\"id\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"max\",\"field
\":\"indices_stats._all.total.docs.deleted\"}],\"separate_axis\":0,\"axis_position\":\"right\",\"formatter
\":\"number\",\"chart_type\":\"line\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none
\",\"label\":\"total_docs_delete\",\"type\":\"timeseries\",\"hidden\":false}],\"time_field\":\"timestamp
\",\"index_pattern\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter
\":\"number\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid\":1,\"tooltip_mode\":\"show_all
\",\"default_index_pattern\":\"monitoring-eye-css-*\",\"default_timefield\":\"timestamp
\",\"isModelInvalid\":false,\"drop_last_bucket\":1,\"legend_position\":\"bottom
\"}}"},"id":"cfbb4e20-664c-11ed-8cd7-973626cf6f70","migrationVersion":
{"visualization":"7.10.0"},"references":
[],"type":"visualization","updated_at":"2022-11-17T07:53:02.743Z","version":"WzY0MSwxMF0="}
{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring] indices docs_count for
top10","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] indices docs_count for
top10\",\"type\":\"metrics\",\"aggs\":[],\"params\":{\"id\":\"61ca57f0-469d-11e7-
af02-69e470af7417\",\"type\":\"timeseries\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-
af02-69e470af7417\",\"color\":\"#68BC00\",\"split_mode\":\"terms\",\"metrics\":[{\"unit\":\"\",\"id
\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"max\",\"field\":\"index_stats.total.docs.count
\"}],\"separate_axis\":0,\"axis_position\":\"right\",\"formatter\":\"number\",\"chart_type\":\"line
\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none\",\"label\":\"docs_count\",\"type
\":\"timeseries\",\"terms_field\":\"index_stats.index\",\"terms_order_by\":\"61ca57f2-469d-11e7-
af02-69e470af7417\",\"split_color_mode\":\"rainbow\"}],\"time_field\":\"timestamp\",\"index_pattern
\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter\":\"number
\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid\":1,\"default_index_pattern\":\"monitoring-
eye-css-*\",\"default_timefield\":\"timestamp\",\"isModelInvalid\":false,\"tooltip_mode\":\"show_all
\"}}"},"id":"225f6020-a5f1-11ec-9a68-ada9d754c566","migrationVersion":
{"visualization":"7.10.0"},"references":
[],"type":"visualization","updated_at":"2022-11-17T09:05:42.853Z","version":"WzY0MCwxMF0="}
{"attributes":{"description":"","kibanaSavedObjectMeta":

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

{"searchSourceJSON":"{}"},"title":"[monitoring] indices store_size for top10","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] indices store_size for top10\",\"type\":\"metrics\",\"aggs\":[],\"params\":{\"id\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"type\":\"timeseries\",\"series\":[{\"id\":\"38474c50-a5f5-11ec-aa10-992297d21a2e\",\"color\":\"#68BC00\",\"split_mode\":\"terms\",\"metrics\":[{\"id\":\"38474c51-a5f5-11ec-aa10-992297d21a2e\",\"type\":\"max\",\"field\":\"index_stats.total.store.size_in_bytes\"}],\"separate_axis\":0,\"axis_position\":\"right\",\"formatter\":\"bytes\",\"chart_type\":\"line\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none\",\"label\":\"store_size for index\",\"type\":\"timeseries\",\"terms_field\":\"index_stats.index\",\"terms_order_by\":\"38474c51-a5f5-11ec-aa10-992297d21a2e\",\"filter\":{\"query\":\"\",\"language\":\"kuery\"},\"split_color_mode\":\"rainbow\"}],\"time_field\":\"timestamp\",\"index_pattern\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter\":\"number\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid\":1,\"default_index_pattern\":\"monitoring-eye-css-*\",\"default_timefield\":\"timestamp\",\"isModelInvalid\":false,\"filter\":{\"query\":\"\",\"language\":\"kuery\"},\"bar_color_rules\":[{\"id\":\"7d9d3cb0-a5f5-11ec-aa10-992297d21a2e\"}],\"tooltip_mode\":\"show_all\"}}"},"id":"c78119a0-a5f5-11ec-9a68-ada9d754c566","migrationVersion":{"visualization":"7.10.0"},"references":[],"type":"visualization","updated_at":"2022-11-17T09:06:14.935Z","version":"WzYzOSwxMF0="}
{"attributes":{"description":"","kibanaSavedObjectMeta":{"searchSourceJSON":"{}"},"title":"[monitoring] search rate of index for top10(/s)","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] search rate of index for top10(/s)\",\"type\":\"metrics\",\"aggs\":[],\"params\":{\"id\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"type\":\"timeseries\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-af02-69e470af7417\",\"color\":\"rgba(99,157,12,1)\",\"split_mode\":\"terms\",\"metrics\":[{\"id\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"max\",\"field\":\"index_stats.total.search.query_total\"},{\"unit\":\"1s\",\"id\":\"fdfdfad0-a5ef-11ec-aa10-992297d21a2e\",\"type\":\"derivative\",\"field\":\"61ca57f2-469d-11e7-af02-69e470af7417\"},{\"unit\":\"\",\"id\":\"0aaa26a0-a5f0-11ec-aa10-992297d21a2e\",\"type\":\"positive_only\",\"field\":\"fdfdfad0-a5ef-11ec-aa10-992297d21a2e\"}],\"separate_axis\":0,\"axis_position\":\"right\",\"formatter\":\"number\",\"chart_type\":\"line\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none\",\"label\":\"search rate\",\"type\":\"timeseries\",\"terms_field\":\"index_stats.index\",\"terms_order_by\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"split_color_mode\":\"rainbow\"}],\"time_field\":\"timestamp\",\"index_pattern\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter\":\"number\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid\":1,\"default_index_pattern\":\"monitoring-eye-css-*\",\"default_timefield\":\"timestamp\",\"isModelInvalid\":false,\"tooltip_mode\":\"show_all\"}}"},"id":"ab503550-a5ef-11ec-9a68-ada9d754c566","migrationVersion":{"visualization":"7.10.0"},"references":[],"type":"visualization","updated_at":"2022-03-17T13:26:38.009Z","version":"WzYzOCwxMF0="}
{"attributes":{"description":"","kibanaSavedObjectMeta":{"searchSourceJSON":"{}"},"title":"[monitoring] indexing rate of index for top10(/s)","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] indexing rate of index for top10(/s)\",\"type\":\"metrics\",\"aggs\":[],\"params\":{\"id\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"type\":\"timeseries\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-af02-69e470af7417\",\"color\":\"#68BC00\",\"split_mode\":\"terms\",\"metrics\":[{\"id\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"max\",\"field\":\"index_stats.total.indexing.index_total\"},{\"unit\":\"1s\",\"id\":\"541ed8f0-a5ee-11ec-aa10-992297d21a2e\",\"type\":\"derivative\",\"field\":\"61ca57f2-469d-11e7-af02-69e470af7417\"},{\"unit\":\"\",\"id\":\"67ec1f50-a5ee-11ec-aa10-992297d21a2e\",\"type\":\"positive_only\",\"field\":\"541ed8f0-a5ee-11ec-aa10-992297d21a2e\"}],\"separate_axis\":0,\"axis_position\":\"right\",\"formatter\":\"number\",\"chart_type\":\"line\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none\",\"label\":\"indexing_rate\",\"type\":\"timeseries\",\"split_filters\":[{\"color\":\"#68BC00\",\"id\":\"81004200-a5ee-11ec-aa10-992297d21a2e\",\"filter\":{\"query\":\"\",\"language\":\"kuery\"}}],\"filter\":{\"query\":\"\",\"language\":\"kuery\"},\"terms_field\":\"index_stats.index\",\"terms_order_by\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"terms_size\":\"10\",\"terms_direction\":\"desc\",\"split_color_mode\":\"rainbow\"}],\"time_field\":\"timestamp\",\"index_pattern\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter\":\"number\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid\":1,\"default_index_pattern\":\"monitoring-eye-css-*\",\"default_timefield\":\"timestamp\",\"isModelInvalid\":false,\"tooltip_mode\":\"show_all\"}}"},"id":"943b3e00-a5ef-11ec-9a68-ada9d754c566","migrationVersion":{"visualization":"7.10.0"},"references":[],"type":"visualization","updated_at":"2022-03-17T13:36:31.344Z","version":"WzYzNywxMF0="}
{"attributes":{"description":"number of document being indexing for primary and replica shards","kibanaSavedObjectMeta":{"searchSourceJSON":"{}"},"title":"[monitoring] Indexing Rate (/s)","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] Indexing Rate (/s)\",\"type\":\"metrics\",\"params\":{\"id\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"type\":\"timeseries\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-af02-69e470af7417\",\"color\":\"rgba(0,32,188,1)\",\"split_mode\":\"everything\",\"metrics\":[{\"id\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"max\",\"field\":\"indices_stats._all.total.indexing.index_total\"},{\"unit\":\"1s\",\"id\":\"fed72db0-a5f8-11ec-aa10-992297d21a2e\",\"type\":\"derivative\",\"field

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

\":\"61ca57f2-469d-11e7-af02-69e470af7417\"},{\"unit\":\"\",\"id\":\"14b66420-a5f9-11ec-aa10-992297d21a2e\",\"type\":\"positive_only\",\"field\":\"fed72db0-a5f8-11ec-aa10-992297d21a2e\"}],\"separate_axis\":0,\"axis_position\":\"right\",\"formatter\":\"number\",\"chart_type\":\"line\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none\",\"label\":\"Indexing Rate (/s)\",\"type\":\"timeseries\",\"split_color_mode\":\"rainbow\",\"hidden\":false}],\"time_field\":\"timestamp\",\"index_pattern\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter\":\"number\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid\":1,\"default_index_pattern\":\"monitoring-eye-css-*\",\"default_timefield\":\"timestamp\",\"isModelInvalid\":false,\"legend_position\":\"bottom\"},\"aggs\":[]}"},"id":"de4f8ab0-a5f8-11ec-9a68-ada9d754c566","migrationVersion":{"visualization":"7.10.0"},"references":[],"type":"visualization","updated_at":"2022-03-18T00:52:40.515Z","version":"WzYzNiwxMF0="}
{"attributes":{"description":"number of search request being executed in primary and replica shards","kibanaSavedObjectMeta":{"searchSourceJSON":"{}"},"title":"[monitoring] Search Rate (/s)","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] Search Rate (/s)\",\"type\":\"metrics\",\"params\":{\"id\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"type\":\"timeseries\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-af02-69e470af7417\",\"color\":\"rgba(0,33,224,1)\",\"split_mode\":\"everything\",\"metrics\":[{\"id\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"max\",\"field\":\"indices_stats._all.total.search.query_total\"},{\"unit\":\"1s\",\"id\":\"b1093ac0-a5f7-11ec-aa10-992297d21a2e\",\"type\":\"derivative\",\"field\":\"61ca57f2-469d-11e7-af02-69e470af7417\"},{\"unit\":\"\",\"id\":\"c17db930-a5f7-11ec-aa10-992297d21a2e\",\"type\":\"positive_only\",\"field\":\"b1093ac0-a5f7-11ec-aa10-992297d21a2e\"}],\"separate_axis\":0,\"axis_position\":\"right\",\"formatter\":\"number\",\"chart_type\":\"line\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none\",\"split_color_mode\":\"rainbow\",\"label\":\"Search Rate (/s)\",\"type\":\"timeseries\",\"filter\":{\"query\":\"\",\"language\":\"kuery\"}}],\"time_field\":\"timestamp\",\"index_pattern\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter\":\"number\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid\":1,\"default_index_pattern\":\"monitoring-eye-css-*\",\"default_timefield\":\"timestamp\",\"isModelInvalid\":false,\"legend_position\":\"bottom\"},\"aggs\":[]}"},"id":"811df7a0-a5f8-11ec-9a68-ada9d754c566","migrationVersion":{"visualization":"7.10.0"},"references":[],"type":"visualization","updated_at":"2022-03-18T00:53:40.866Z","version":"WzYzNSwxMF0="}
{"attributes":{"description":"","kibanaSavedObjectMeta":{"searchSourceJSON":"{}"},"title":"[monitoring] markdown","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] markdown\",\"type\":\"markdown\",\"params\":{\"fontSize\":12,\"openLinksInNewTab\":false,\"markdown\":\"### Index Monitoring \\nThis dashboard contains default table for you to play with. You can view it, search it, and interact with the visualizations.\"},\"aggs\":[]}"},"id":"b2811c70-a5f1-11ec-9a68-ada9d754c566","migrationVersion":{"visualization":"7.10.0"},"references":[],"type":"visualization","updated_at":"2022-03-17T12:56:44.215Z","version":"WzYzNCwxMF0="}
{"attributes":{"fields":"[{\"count\":0,\"name\":\"_id\",\"type\":\"string\",\"esTypes\":[\"_id\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":false},{\"count\":0,\"name\":\"_index\",\"type\":\"string\",\"esTypes\":[\"_index\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":false},{\"count\":0,\"name\":\"_score\",\"type\":\"number\",\"scripted\":false,\"searchable\":false,\"aggregatable\":false,\"readFromDocValues\":false},{\"count\":0,\"name\":\"_source\",\"type\":\"_source\",\"esTypes\":[\"_source\"],\"scripted\":false,\"searchable\":false,\"aggregatable\":false,\"readFromDocValues\":false},{\"count\":0,\"name\":\"_type\",\"type\":\"string\",\"esTypes\":[\"_type\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":false},{\"count\":0,\"name\":\"cluster_uuid\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"index_stats.created\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"index_stats.index\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"index_stats.primaries.docs.count\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"index_stats.primaries.docs.deleted\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"index_stats.primaries.fielddata.evictions\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"index_stats.primaries.fielddata.memory_size_in_bytes\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"index_stats.primaries.flush.total_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"index_stats.primaries.indexing.delete_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"index_stats.primaries.indexing.delete_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"index_stats.primaries.indexing.delete_total\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.primaries.indexing.index_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.primaries.indexing.index_failed\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.primaries.indexing.index_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.primaries.indexing.index_total\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.primaries.indexing.throttle_time_in_millis\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.primaries.merges.current\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.primaries.merges.total_size_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.primaries.query_cache.cache_count\",\"type\":\"number\",\"esTypes
\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.primaries.query_cache.cache_size\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.primaries.query_cache.evictions\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.primaries.query_cache.hit_count\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.primaries.query_cache.memory_size_in_bytes\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.primaries.query_cache.miss_count\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.primaries.query_cache.total_count\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.primaries.recovery.current_as_source\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.primaries.recovery.current_as_target\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.primaries.recovery.throttle_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.primaries.refresh.external_total_time_in_millis\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.primaries.refresh.total_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.primaries.request_cache.evictions\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.primaries.request_cache.hit_count\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.primaries.request_cache.memory_size_in_bytes\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.primaries.request_cache.miss_count\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.primaries.search.fetch_current\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.primaries.search.fetch_time_in_millis\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.primaries.search.fetch_total\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.primaries.search.open_contexts\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.primaries.search.query_current\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.primaries.search.query_time_in_millis\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"index_stats.primaries.search.query_total
\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"index_stats.primaries.search.scroll_current
\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.primaries.search.scroll_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.primaries.search.scroll_total\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.primaries.search.suggest_current\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.primaries.search.suggest_time_in_millis\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.primaries.search.suggest_total\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.primaries.segments.count\",\"type\":\"number\",\"esTypes\":
[\"integer\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.primaries.segments.doc_values_memory_in_bytes\",\"type
\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.primaries.segments.fixed_bit_set_memory_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.primaries.segments.index_writer_memory_in_bytes\",\"type
\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.primaries.segments.memory_in_bytes\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.primaries.segments.norms_memory_in_bytes\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.primaries.segments.points_memory_in_bytes\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.primaries.segments.stored_fields_memory_in_bytes\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.primaries.segments.term_vectors_memory_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.primaries.segments.terms_memory_in_bytes\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.primaries.segments.version_map_memory_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.primaries.store.size_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.shards.active_primaries_shards\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.shards.active_replicas\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.shards.active_shards\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.shards.initializing_shards\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.shards.primaries_shards\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.shards.relocating_shards\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.shards.replicas_shards\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.shards.total_shards\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.shards.unassigned_shards\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.status\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"scripted\":false,\"searchable
\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.docs.count\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.docs.deleted\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.fielddata.evictions\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.fielddata.memory_size_in_bytes\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.total.flush.total_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.total.indexing.delete_current\",\"type\":\"number\",\"esTypes\":[\"long

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.total.indexing.delete_time_in_millis\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.total.indexing.delete_total\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.total.indexing.index_current\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.total.indexing.index_failed\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.total.indexing.index_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.total.indexing.index_total\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.total.indexing.throttle_time_in_millis\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.total.merges.total_size_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.total.query_cache.cache_count\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.total.query_cache.cache_size\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.total.query_cache.evictions\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.total.query_cache.hit_count\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.total.query_cache.memory_size_in_bytes\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"index_stats.total.query_cache.miss_count
\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"index_stats.total.query_cache.total_count
\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.recovery.current_as_source\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.recovery.current_as_target\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.recovery.throttle_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.total.refresh.external_total_time_in_millis\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.total.refresh.total_time_in_millis\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.total.request_cache.evictions\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.total.request_cache.hit_count\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.total.request_cache.memory_size_in_bytes\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"index_stats.total.request_cache.miss_count
\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"index_stats.total.search.fetch_current
\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.search.fetch_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.search.fetch_total\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.search.open_contexts\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.search.query_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.search.query_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.search.query_total\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.search.scroll_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.search.scroll_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

\":\"index_stats.total.search.scroll_total\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.search.suggest_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.search.suggest_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.total.search.suggest_total\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.total.segments.count\",\"type\":\"number\",\"esTypes\":[\"integer
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.total.segments.doc_values_memory_in_bytes\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.segments.fixed_bit_set_memory_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.total.segments.index_writer_memory_in_bytes\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.segments.memory_in_bytes\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.segments.norms_memory_in_bytes\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.total.segments.points_memory_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.total.segments.stored_fields_memory_in_bytes\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.segments.term_vectors_memory_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.total.segments.terms_memory_in_bytes\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"index_stats.total.segments.version_map_memory_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"index_stats.total.store.size_in_bytes\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"index_stats.uuid\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"indices_stats._all.primaries.docs.count\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"indices_stats._all.primaries.docs.deleted\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"indices_stats._all.primaries.indexing.index_current\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"indices_stats._all.primaries.indexing.index_failed\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"indices_stats._all.primaries.indexing.index_time_in_millis\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"indices_stats._all.primaries.indexing.index_total\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"indices_stats._all.primaries.indexing.throttle_time_in_millis\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"indices_stats._all.primaries.search.fetch_current\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"indices_stats._all.primaries.search.fetch_time_in_millis\",\"type\":\"number\",\"esTypes
\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"indices_stats._all.primaries.search.fetch_total\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"indices_stats._all.primaries.search.open_contexts\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"indices_stats._all.primaries.search.query_current\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count
\":0,\"name\":\"indices_stats._all.primaries.search.query_time_in_millis\",\"type\":\"number\",\"esTypes
\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"count\":0,\"name\":\"indices_stats._all.primaries.search.query_total\",\"type\":\"number\",\"esTypes
\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

{\"count\":0,\"name\":\"indices_stats._all.primaries.search.scroll_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.primaries.search.scroll_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.primaries.search.scroll_total\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.primaries.search.suggest_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.primaries.search.suggest_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.primaries.search.suggest_total\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.docs.count\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.docs.deleted\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.indexing.index_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.indexing.index_failed\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.indexing.index_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.indexing.index_total\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.indexing.throttle_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.search.fetch_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.search.fetch_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.search.fetch_total\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.search.open_contexts\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.search.query_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.search.query_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.search.query_total\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.search.scroll_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.search.scroll_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.search.scroll_total\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.search.suggest_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.search.suggest_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"indices_stats._all.total.search.suggest_total\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"interval\",\"type\":\"number\",\"esTypes\":[\"long\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"source_node.host\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"source_node.ip\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"source_node.name\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"source_node.timestamp\",\"type\":\"date\",\"esTypes\":[\"date\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"source_node.transport_address\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"source_node.uuid\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name\":\"state_uuid\",\"type\":\"string\",\"esTypes

\":[\"keyword\"],\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"count\":0,\"name\":\"timestamp\",\"type\":\"date\",\"esTypes\":[\"date\"],\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"count\":0,\"name
\":\"type\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"scripted\":false,\"searchable
\":true,\"aggregatable\":true,\"readFromDocValues
\":true}]","timeFieldName":"timestamp","title":"monitoring-eye-css-*"},"id":"153bc490-
a5a8-11ec-9a68-ada9d754c566","migrationVersion":{"index-pattern":"7.6.0"},"references":
[],"type":"index-pattern","updated_at":"2022-03-17T08:10:17.348Z","version":"WzY0OCwxMF0="}
{"attributes":{"description":"Analyze usage statistics of index
monitoring","hits":0,"kibanaSavedObjectMeta":{"searchSourceJSON":"{\"query\":{\"language\":\"kuery
\",\"query\":\"\"},\"filter\":[]}"},"optionsJSON":"{\"hidePanelTitles\":false,\"useMargins
\":true}","panelsJSON":"[{\"version\":\"7.10.2\",\"gridData\":{\"x\":0,\"y\":0,\"w\":48,\"h\":5,\"i
\":\"971ed6c6-81b9-491b-9f08-e3ae9c382abd\"},\"panelIndex\":\"971ed6c6-81b9-491b-9f08-
e3ae9c382abd\",\"embeddableConfig\":{},\"panelRefName\":\"panel_0\"},{\"version
\":\"7.10.2\",\"gridData\":{\"x\":0,\"y\":5,\"w\":24,\"h\":15,\"i\":\"5a6982e7-0c6c-4733-8a2d-
e4c57cdf7397\"},\"panelIndex\":\"5a6982e7-0c6c-4733-8a2d-e4c57cdf7397\",\"embeddableConfig\":
{},\"panelRefName\":\"panel_1\"},{\"version\":\"7.10.2\",\"gridData\":{\"x\":24,\"y\":5,\"w\":24,\"h
\":15,\"i\":\"662476f4-739c-4a05-858c-2ee8230cf410\"},\"panelIndex
\":\"662476f4-739c-4a05-858c-2ee8230cf410\",\"embeddableConfig\":{},\"panelRefName
\":\"panel_2\"},{\"version\":\"7.10.2\",\"gridData\":{\"x\":0,\"y\":20,\"w\":17,\"h\":17,\"i
\":\"d89c38e2-33f3-4592-b503-20460a6a7a57\"},\"panelIndex\":\"d89c38e2-33f3-4592-
b503-20460a6a7a57\",\"embeddableConfig\":{},\"panelRefName\":\"panel_3\"},{\"version
\":\"7.10.2\",\"gridData\":{\"x\":17,\"y\":20,\"w\":16,\"h\":17,\"i
\":\"1f693b49-79fa-4807-94e8-0c12f51e54f8\"},\"panelIndex
\":\"1f693b49-79fa-4807-94e8-0c12f51e54f8\",\"embeddableConfig\":{},\"panelRefName
\":\"panel_4\"},{\"version\":\"7.10.2\",\"gridData\":{\"x\":33,\"y\":20,\"w\":15,\"h\":17,\"i
\":\"616b143d-74e9-4dac-98ba-5849536f0fba\"},\"panelIndex
\":\"616b143d-74e9-4dac-98ba-5849536f0fba\",\"embeddableConfig\":{},\"panelRefName
\":\"panel_5\"},{\"version\":\"7.10.2\",\"gridData\":{\"x\":0,\"y\":37,\"w\":24,\"h\":11,\"i
\":\"cfa82f27-1b8d-49ba-a7b9-d8809d3b258c\"},\"panelIndex\":\"cfa82f27-1b8d-49ba-a7b9-
d8809d3b258c\",\"embeddableConfig\":{},\"panelRefName\":\"panel_6\"},{\"version
\":\"7.10.2\",\"gridData\":{\"x\":24,\"y\":37,\"w\":24,\"h\":11,\"i\":\"135d13eb-
aab6-43ca-9029-7d26e91d90e3\"},\"panelIndex\":\"135d13eb-
aab6-43ca-9029-7d26e91d90e3\",\"embeddableConfig\":{},\"panelRefName\":\"panel_7\"},{\"version
\":\"7.10.2\",\"gridData\":{\"x\":0,\"y\":48,\"w\":24,\"h\":11,\"i\":\"28a77de1-9110-49e8-
b273-724f880b1653\"},\"panelIndex\":\"28a77de1-9110-49e8-
b273-724f880b1653\",\"embeddableConfig\":{},\"panelRefName\":\"panel_8\"},{\"version
\":\"7.10.2\",\"gridData\":{\"x\":24,\"y\":48,\"w\":24,\"h\":11,\"i\":\"80ece867-cf23-4935-
bfbc-430afa51bcca\"},\"panelIndex\":\"80ece867-cf23-4935-bfbc-430afa51bcca\",\"embeddableConfig
\":{},\"panelRefName\":\"panel_9\"},{\"version\":\"7.10.2\",\"gridData\":{\"x\":0,\"y\":59,\"w\":24,\"h
\":11,\"i\":\"2ba970aa-c9c4-491b-bdd3-c1b1ee9bc8d3\"},\"panelIndex\":\"2ba970aa-c9c4-491b-bdd3-
c1b1ee9bc8d3\",\"embeddableConfig\":{},\"panelRefName\":\"panel_10\"},{\"version
\":\"7.10.2\",\"gridData\":{\"x\":24,\"y\":59,\"w\":24,\"h\":11,\"i\":\"f2e1b6ab-ddf7-492e-
aaca-9460f11aa4aa\"},\"panelIndex\":\"f2e1b6ab-ddf7-492e-aaca-9460f11aa4aa
\",\"embeddableConfig\":{},\"panelRefName\":\"panel_11\"},{\"version\":\"7.10.2\",\"gridData\":{\"x
\":0,\"y\":70,\"w\":24,\"h\":11,\"i\":\"dd14182d-d8b9-47f2-bf36-6cba3b09586c\"},\"panelIndex
\":\"dd14182d-d8b9-47f2-bf36-6cba3b09586c\",\"embeddableConfig\":{},\"panelRefName
\":\"panel_12\"},{\"version\":\"7.10.2\",\"gridData\":{\"x\":24,\"y\":70,\"w\":24,\"h\":11,\"i
\":\"a47f9333-52b7-49b7-8cac-f470cf405131\"},\"panelIndex\":\"a47f9333-52b7-49b7-8cac-
f470cf405131\",\"embeddableConfig\":{},\"panelRefName
\":\"panel_13\"}]","timeRestore":false,"title":"[Monitoring] Index monitoring
Dashboard","version":1},"id":"524eb000-a5f2-11ec-9a68-ada9d754c566","migrationVersion":
{"dashboard":"7.9.3"},"references":[{"id":"b2811c70-a5f1-11ec-9a68-
ada9d754c566","name":"panel_0","type":"visualization"},{"id":"de4f8ab0-a5f8-11ec-9a68-
ada9d754c566","name":"panel_1","type":"visualization"},{"id":"811df7a0-a5f8-11ec-9a68-
ada9d754c566","name":"panel_2","type":"visualization"},
{"id":"eea89780-664b-11ed-8cd7-973626cf6f70","name":"panel_3","type":"visualization"},
{"id":"cfbb4e20-664c-11ed-8cd7-973626cf6f70","name":"panel_4","type":"visualization"},
{"id":"c7f72ae0-664e-11ed-8cd7-973626cf6f70","name":"panel_5","type":"visualization"},
{"id":"943b3e00-a5ef-11ec-9a68-ada9d754c566","name":"panel_6","type":"visualization"},
{"id":"ab503550-a5ef-11ec-9a68-ada9d754c566","name":"panel_7","type":"visualization"},
{"id":"c78119a0-a5f5-11ec-9a68-ada9d754c566","name":"panel_8","type":"visualization"},
{"id":"225f6020-a5f1-11ec-9a68-ada9d754c566","name":"panel_9","type":"visualization"},
{"id":"17d49220-662a-11ed-8cd7-973626cf6f70","name":"panel_10","type":"visualization"},
{"id":"c8109100-6627-11ed-8cd7-973626cf6f70","name":"panel_11","type":"visualization"},
{"id":"45d571c0-6626-11ed-8cd7-973626cf6f70","name":"panel_12","type":"visualization"},
{"id":"3ae5d820-6628-11ed-8cd7-973626cf6f70","name":"panel_13","type":"visualization"}],"type":"das
hboard","updated_at":"2022-11-17T09:25:14.003Z","version":"WzY0OSwxMF0="}
{"exportedCount":16,"missingRefCount":0,"missingReferences":[]}

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- **Elasticsearch 7.6.2 cluster**

{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring]
markdown","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] markdown\",\"type
\":\"markdown\",\"params\":{\"fontSize\":12,\"openLinksInNewTab\":false,\"markdown\":\"### Index
Monitoring \\nThis dashboard contains default table for you to play with. You can view it, search it,
and interact with the visualizations.\"},\"aggs\":[]}"},"id":"b2811c70-a5f1-11ec-9a68-
ada9d754c566","migrationVersion":{"visualization":"7.4.2"},"references":
[],"type":"visualization","updated_at":"2022-03-17T12:56:44.215Z","version":"WzQsMV0="}
{"attributes":{"description":"number of document being indexing for primary and replica
shards","kibanaSavedObjectMeta":{"searchSourceJSON":"{}"},"title":"[monitoring] Indexing Rate (/
s)","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] Indexing Rate (/s)\",\"type
\":\"metrics\",\"params\":{\"id\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"type\":\"timeseries
\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-af02-69e470af7417\",\"color
\":\"rgba(0,32,188,1)\",\"split_mode\":\"everything\",\"metrics\":[{\"id\":\"61ca57f2-469d-11e7-
af02-69e470af7417\",\"type\":\"max\",\"field\":\"indices_stats._all.total.indexing.index_total\"},{\"unit
\":\"1s\",\"id\":\"fed72db0-a5f8-11ec-aa10-992297d21a2e\",\"type\":\"derivative\",\"field
\":\"61ca57f2-469d-11e7-af02-69e470af7417\"},{\"unit\":\"\",\"id\":\"14b66420-a5f9-11ec-
aa10-992297d21a2e\",\"type\":\"positive_only\",\"field\":\"fed72db0-a5f8-11ec-aa10-992297d21a2e
\"}],\"separate_axis\":0,\"axis_position\":\"right\",\"formatter\":\"number\",\"chart_type\":\"line
\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none\",\"label\":\"Indexing Rate (/s)\",\"type
\":\"timeseries\",\"split_color_mode\":\"rainbow\",\"hidden\":false}],\"time_field\":\"timestamp
\",\"index_pattern\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter
\":\"number\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid\":1,\"default_index_pattern
\":\"monitoring-eye-css-*\",\"default_timefield\":\"timestamp\",\"isModelInvalid
\":false,\"legend_position\":\"bottom\"},\"aggs\":[]}"},"id":"de4f8ab0-a5f8-11ec-9a68-
ada9d754c566","migrationVersion":{"visualization":"7.4.2"},"references":
[],"type":"visualization","updated_at":"2022-03-18T00:52:40.515Z","version":"WzYsMV0="}
{"attributes":{"description":"number of search request being executed in primary and replica
shards","kibanaSavedObjectMeta":{"searchSourceJSON":"{}"},"title":"[monitoring] Search Rate (/
s)","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] Search Rate (/s)\",\"type
\":\"metrics\",\"params\":{\"id\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"type\":\"timeseries
\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-af02-69e470af7417\",\"color
\":\"rgba(0,33,224,1)\",\"split_mode\":\"everything\",\"metrics\":[{\"id\":\"61ca57f2-469d-11e7-
af02-69e470af7417\",\"type\":\"max\",\"field\":\"indices_stats._all.total.search.query_total\"},{\"unit
\":\"1s\",\"id\":\"b1093ac0-a5f7-11ec-aa10-992297d21a2e\",\"type\":\"derivative\",\"field
\":\"61ca57f2-469d-11e7-af02-69e470af7417\"},{\"unit\":\"\",\"id\":\"c17db930-a5f7-11ec-
aa10-992297d21a2e\",\"type\":\"positive_only\",\"field\":\"b1093ac0-a5f7-11ec-aa10-992297d21a2e
\"}],\"separate_axis\":0,\"axis_position\":\"right\",\"formatter\":\"number\",\"chart_type\":\"line
\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none\",\"split_color_mode\":\"rainbow
\",\"label\":\"Search Rate (/s)\",\"type\":\"timeseries\",\"filter\":{\"query\":\"\",\"language\":\"kuery
\"}}],\"time_field\":\"timestamp\",\"index_pattern\":\"monitoring-eye-css-*\",\"interval
\":\"\",\"axis_position\":\"left\",\"axis_formatter\":\"number\",\"axis_scale\":\"normal\",\"show_legend
\":1,\"show_grid\":1,\"default_index_pattern\":\"monitoring-eye-css-*\",\"default_timefield
\":\"timestamp\",\"isModelInvalid\":false,\"legend_position\":\"bottom\"},\"aggs\":[]}"},"id":"811df7a0-
a5f8-11ec-9a68-ada9d754c566","migrationVersion":{"visualization":"7.4.2"},"references":
[],"type":"visualization","updated_at":"2022-03-18T00:53:40.866Z","version":"WzUsMV0="}
{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring] total docs
count","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] total docs count\",\"type
\":\"metrics\",\"aggs\":[],\"params\":{\"id\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"type
\":\"timeseries\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-af02-69e470af7417\",\"color
\":\"rgba(218,139,69,1)\",\"split_mode\":\"everything\",\"split_color_mode\":\"kibana\",\"metrics\":
[{\"unit\":\"\",\"id\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"max\",\"field
\":\"indices_stats._all.total.docs.count\"}],\"separate_axis\":0,\"axis_position\":\"right\",\"formatter
\":\"number\",\"chart_type\":\"line\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none
\",\"label\":\"total_docs_count\",\"type\":\"timeseries\"}],\"time_field\":\"timestamp\",\"index_pattern
\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter\":\"number
\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid\":1,\"tooltip_mode\":\"show_all
\",\"default_index_pattern\":\"monitoring-eye-css-*\",\"default_timefield\":\"timestamp
\",\"isModelInvalid\":false,\"legend_position\":\"bottom
\"}}"},"id":"eea89780-664b-11ed-8cd7-973626cf6f70","migrationVersion":
{"visualization":"7.4.2"},"references":
[],"type":"visualization","updated_at":"2022-11-17T07:53:02.743Z","version":"WzEyLDFd"}
{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring] total docs
delete","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] total docs delete\",\"type
\":\"metrics\",\"aggs\":[],\"params\":{\"id\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"type
\":\"timeseries\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-af02-69e470af7417\",\"color

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

\":\"rgba(214,191,87,1)\",\"split_mode\":\"everything\",\"split_color_mode\":\"kibana\",\"metrics\":
[{\"id\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"max\",\"field
\":\"indices_stats._all.total.docs.deleted\"}],\"separate_axis\":0,\"axis_position\":\"right\",\"formatter
\":\"number\",\"chart_type\":\"line\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none
\",\"label\":\"total_docs_delete\",\"type\":\"timeseries\",\"hidden\":false}],\"time_field\":\"timestamp
\",\"index_pattern\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter
\":\"number\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid\":1,\"tooltip_mode\":\"show_all
\",\"default_index_pattern\":\"monitoring-eye-css-*\",\"default_timefield\":\"timestamp
\",\"isModelInvalid\":false,\"drop_last_bucket\":1,\"legend_position\":\"bottom
\"}}"},"id":"cfbb4e20-664c-11ed-8cd7-973626cf6f70","migrationVersion":
{"visualization":"7.4.2"},"references":
[],"type":"visualization","updated_at":"2022-11-17T07:53:02.743Z","version":"WzExLDFd"}
{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring] total store size in
bytes","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] total store size in bytes
\",\"type\":\"metrics\",\"aggs\":[],\"params\":{\"id\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"type
\":\"timeseries\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-af02-69e470af7417\",\"color
\":\"#68BC00\",\"split_mode\":\"everything\",\"split_color_mode\":\"kibana\",\"metrics\":[{\"id
\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"max\",\"field
\":\"indices_stats._all.total.store.size_in_bytes\"}],\"separate_axis\":0,\"axis_position\":\"right
\",\"formatter\":\"bytes\",\"chart_type\":\"line\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked
\":\"none\",\"label\":\"total store size in bytes\",\"type\":\"timeseries\"}],\"time_field\":\"timestamp
\",\"index_pattern\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter
\":\"number\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid\":1,\"tooltip_mode\":\"show_all
\",\"default_index_pattern\":\"monitoring-eye-css-*\",\"default_timefield\":\"timestamp
\",\"isModelInvalid\":false,\"legend_position\":\"bottom\",\"background_color_rules\":[{\"id
\":\"7712e550-664f-11ed-8b5d-8db37e5b4cc4\"}],\"bar_color_rules\":[{\"id
\":\"77680a30-664f-11ed-8b5d-8db37e5b4cc4\"}]}}"},"id":"c7f72ae0-664e-11ed-8cd7-973626cf6f70","
migrationVersion":{"visualization":"7.4.2"},"references":
[],"type":"visualization","updated_at":"2022-11-17T07:53:02.743Z","version":"WzEzLDFd"}
{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring] indexing rate of index for top10(/
s)","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] indexing rate of index for
top10\",\"type\":\"metrics\",\"params\":{\"id\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"type
\":\"timeseries\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-af02-69e470af7417\",\"color
\":\"#68BC00\",\"split_mode\":\"terms\",\"metrics\":[{\"id\":\"61ca57f2-469d-11e7-
af02-69e470af7417\",\"type\":\"max\",\"field\":\"index_stats.total.indexing.index_total\"},{\"unit\":\"1s
\",\"id\":\"541ed8f0-a5ee-11ec-aa10-992297d21a2e\",\"type\":\"derivative\",\"field
\":\"61ca57f2-469d-11e7-af02-69e470af7417\"},{\"unit\":\"\",\"id\":\"67ec1f50-a5ee-11ec-
aa10-992297d21a2e\",\"type\":\"positive_only\",\"field\":\"541ed8f0-a5ee-11ec-aa10-992297d21a2e
\"}],\"separate_axis\":0,\"axis_position\":\"right\",\"formatter\":\"number\",\"chart_type\":\"line
\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none\",\"label\":\"indexing_rate\",\"type
\":\"timeseries\",\"split_filters\":[{\"color\":\"#68BC00\",\"id\":\"81004200-a5ee-11ec-
aa10-992297d21a2e\",\"filter\":{\"query\":\"\",\"language\":\"kuery\"}}],\"filter\":{\"query
\":\"\",\"language\":\"kuery\"},\"terms_field\":\"index_stats.index\",\"terms_order_by
\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"terms_size\":\"10\",\"terms_direction\":\"desc
\",\"split_color_mode\":\"rainbow\"}],\"time_field\":\"timestamp\",\"index_pattern\":\"monitoring-eye-
css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter\":\"number\",\"axis_scale\":\"normal
\",\"show_legend\":1,\"show_grid\":1,\"default_index_pattern\":\"monitoring-eye-css-*
\",\"default_timefield\":\"timestamp\",\"isModelInvalid\":false},\"aggs\":[]}"},"id":"943b3e00-
a5ef-11ec-9a68-ada9d754c566","migrationVersion":{"visualization":"7.4.2"},"references":
[],"type":"visualization","updated_at":"2022-03-17T13:36:31.344Z","version":"WzcsMV0="}
{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring] search rate of index for top10(/
s)","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] search rate of index for
top10\",\"type\":\"metrics\",\"params\":{\"id\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"type
\":\"timeseries\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-af02-69e470af7417\",\"color
\":\"rgba(99,157,12,1)\",\"split_mode\":\"terms\",\"metrics\":[{\"id\":\"61ca57f2-469d-11e7-
af02-69e470af7417\",\"type\":\"max\",\"field\":\"index_stats.total.search.query_total\"},{\"unit\":\"1s
\",\"id\":\"fdfdfad0-a5ef-11ec-aa10-992297d21a2e\",\"type\":\"derivative\",\"field
\":\"61ca57f2-469d-11e7-af02-69e470af7417\"},{\"unit\":\"\",\"id\":\"0aaa26a0-a5f0-11ec-
aa10-992297d21a2e\",\"type\":\"positive_only\",\"field\":\"fdfdfad0-a5ef-11ec-aa10-992297d21a2e
\"}],\"separate_axis\":0,\"axis_position\":\"right\",\"formatter\":\"number\",\"chart_type\":\"line
\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none\",\"label\":\"search rate\",\"type
\":\"timeseries\",\"terms_field\":\"index_stats.index\",\"terms_order_by\":\"61ca57f2-469d-11e7-
af02-69e470af7417\",\"split_color_mode\":\"rainbow\"}],\"time_field\":\"timestamp\",\"index_pattern
\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter\":\"number
\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid\":1,\"default_index_pattern\":\"monitoring-
eye-css-*\",\"default_timefield\":\"timestamp\",\"isModelInvalid\":false},\"aggs\":[]}"},"id":"ab503550-

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

a5ef-11ec-9a68-ada9d754c566","migrationVersion":{"visualization":"7.4.2"},"references":
[],"type":"visualization","updated_at":"2022-03-18T13:25:57.434Z","version":"WzgsMV0="}
{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring] indices store_size for
top10","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] indices store_size for
top10\",\"type\":\"metrics\",\"aggs\":[],\"params\":{\"id\":\"61ca57f0-469d-11e7-
af02-69e470af7417\",\"type\":\"timeseries\",\"series\":[{\"id\":\"38474c50-a5f5-11ec-
aa10-992297d21a2e\",\"color\":\"#68BC00\",\"split_mode\":\"terms\",\"metrics\":[{\"id\":\"38474c51-
a5f5-11ec-aa10-992297d21a2e\",\"type\":\"max\",\"field\":\"index_stats.total.store.size_in_bytes
\"}],\"separate_axis\":0,\"axis_position\":\"right\",\"formatter\":\"bytes\",\"chart_type\":\"line
\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none\",\"label\":\"store_size for index
\",\"type\":\"timeseries\",\"terms_field\":\"index_stats.index\",\"terms_order_by\":\"38474c51-
a5f5-11ec-aa10-992297d21a2e\",\"filter\":{\"query\":\"\",\"language\":\"kuery\"},\"split_color_mode
\":\"rainbow\"}],\"time_field\":\"timestamp\",\"index_pattern\":\"monitoring-eye-css-*\",\"interval
\":\"\",\"axis_position\":\"left\",\"axis_formatter\":\"number\",\"axis_scale\":\"normal\",\"show_legend
\":1,\"show_grid\":1,\"default_index_pattern\":\"monitoring-eye-css-*\",\"default_timefield
\":\"timestamp\",\"isModelInvalid\":false,\"filter\":{\"query\":\"\",\"language\":\"kuery
\"},\"bar_color_rules\":[{\"id\":\"7d9d3cb0-a5f5-11ec-aa10-992297d21a2e\"}],\"tooltip_mode
\":\"show_all\"}}"},"id":"c78119a0-a5f5-11ec-9a68-ada9d754c566","migrationVersion":
{"visualization":"7.4.2"},"references":
[],"type":"visualization","updated_at":"2022-03-17T13:27:25.732Z","version":"WzksMV0="}
{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring] indices docs_count for
top10","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] indices docs_count for
top10\",\"type\":\"metrics\",\"aggs\":[],\"params\":{\"id\":\"61ca57f0-469d-11e7-
af02-69e470af7417\",\"type\":\"timeseries\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-
af02-69e470af7417\",\"color\":\"#68BC00\",\"split_mode\":\"terms\",\"metrics\":[{\"unit\":\"\",\"id
\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"max\",\"field\":\"index_stats.total.docs.count
\"}],\"separate_axis\":0,\"axis_position\":\"right\",\"formatter\":\"number\",\"chart_type\":\"line
\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none\",\"label\":\"docs_count\",\"type
\":\"timeseries\",\"terms_field\":\"index_stats.index\",\"terms_order_by\":\"61ca57f2-469d-11e7-
af02-69e470af7417\",\"split_color_mode\":\"rainbow\"}],\"time_field\":\"timestamp\",\"index_pattern
\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter\":\"number
\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid\":1,\"default_index_pattern\":\"monitoring-
eye-css-*\",\"default_timefield\":\"timestamp\",\"isModelInvalid\":false,\"tooltip_mode\":\"show_all
\"}}"},"id":"225f6020-a5f1-11ec-9a68-ada9d754c566","migrationVersion":
{"visualization":"7.4.2"},"references":
[],"type":"visualization","updated_at":"2022-03-17T13:26:38.009Z","version":"WzEwLDFd"}
{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring] indexing time in millis of index for
top10(ms)","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] indexing time in millis
of index for top10(ms)\",\"type\":\"metrics\",\"aggs\":[],\"params\":{\"id\":\"61ca57f0-469d-11e7-
af02-69e470af7417\",\"type\":\"timeseries\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-
af02-69e470af7417\",\"color\":\"rgba(218,139,69,1)\",\"split_mode\":\"terms\",\"split_color_mode
\":\"kibana\",\"metrics\":[{\"id\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"max\",\"field
\":\"index_stats.total.indexing.index_time_in_millis\"},{\"unit\":\"1s\",\"id
\":\"8099c0d0-6645-11ed-925a-6de90846447d\",\"type\":\"derivative\",\"field\":\"61ca57f2-469d-11e7-
af02-69e470af7417\"},{\"unit\":\"\",\"id\":\"e2c583e0-6652-11ed-b184-3911d8096b2b\",\"type
\":\"positive_only\",\"field\":\"8099c0d0-6645-11ed-925a-6de90846447d\"}],\"separate_axis
\":0,\"axis_position\":\"right\",\"formatter\":\"number\",\"chart_type\":\"line\",\"line_width
\":1,\"point_size\":1,\"fill\":0.5,\"stacked\":\"none\",\"label\":\"indexing index time in millis of index for
top 10\",\"type\":\"timeseries\",\"terms_field\":\"index_stats.index\",\"terms_order_by
\":\"61ca57f2-469d-11e7-af02-69e470af7417\"}],\"time_field\":\"timestamp\",\"index_pattern
\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left\",\"axis_formatter\":\"number
\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid\":1,\"tooltip_mode\":\"show_all
\",\"default_index_pattern\":\"monitoring-eye-css-*\",\"default_timefield\":\"timestamp
\",\"isModelInvalid\":false,\"background_color_rules\":[{\"id\":\"2930b170-662a-11ed-8056-
b7a911ede866\"}],\"bar_color_rules\":[{\"id\":\"2a1a0000-662a-11ed-8056-
b7a911ede866\"}],\"gauge_color_rules\":[{\"id\":\"2adeaf90-662a-11ed-8056-
b7a911ede866\"}],\"gauge_width\":10,\"gauge_inner_width\":10,\"gauge_style\":\"half\",\"axis_min
\":\"\"}}"},"id":"17d49220-662a-11ed-8cd7-973626cf6f70","migrationVersion":
{"visualization":"7.4.2"},"references":
[],"type":"visualization","updated_at":"2022-11-17T07:53:02.743Z","version":"WzE0LDFd"}
{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring] search query time in millis of index for
top10(ms)","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] search query time in
millis of index for top10(ms)\",\"type\":\"metrics\",\"aggs\":[],\"params\":{\"axis_formatter\":\"number
\",\"axis_max\":\"\",\"axis_min\":\"\",\"axis_position\":\"left\",\"axis_scale\":\"normal
\",\"default_index_pattern\":\"monitoring-eye-css-*\",\"default_timefield\":\"timestamp\",\"id

\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"index_pattern\":\"monitoring-eye-css-*\",\"interval
\":\"\",\"isModelInvalid\":false,\"series\":[{\"axis_position\":\"right\",\"chart_type\":\"line\",\"color
\":\"#68BC00\",\"fill\":0.5,\"formatter\":\"number\",\"id\":\"61ca57f1-469d-11e7-
af02-69e470af7417\",\"label\":\"index_query_time_in_millis\",\"line_width\":1,\"metrics\":[{\"field
\":\"index_stats.total.search.query_time_in_millis\",\"id\":\"61ca57f2-469d-11e7-
af02-69e470af7417\",\"type\":\"max\"},{\"unit\":\"1s\",\"id
\":\"42c92b10-6645-11ed-925a-6de90846447d\",\"type\":\"derivative\",\"field\":\"61ca57f2-469d-11e7-
af02-69e470af7417\"}],\"point_size\":1,\"separate_axis\":0,\"split_color_mode\":\"kibana\",\"split_mode
\":\"terms\",\"stacked\":\"none\",\"terms_field\":\"index_stats.index\",\"terms_order_by
\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"timeseries\"}],\"show_grid\":1,\"show_legend
\":1,\"time_field\":\"timestamp\",\"tooltip_mode\":\"show_all\",\"type\":\"timeseries
\",\"background_color\":null,\"filter\":{\"query\":\"\",\"language\":\"kuery\"},\"legend_position\":\"right
\"}}"},"id":"c8109100-6627-11ed-8cd7-973626cf6f70","migrationVersion":
{"visualization":"7.4.2"},"references":
[],"type":"visualization","updated_at":"2022-11-17T07:53:02.743Z","version":"WzE1LDFd"}
{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring] segment count of index for
top10","uiStateJSON":"{}","version":1,"visState":"{\"aggs\":[],\"params\":{\"axis_formatter\":\"number
\",\"axis_position\":\"left\",\"axis_scale\":\"normal\",\"default_index_pattern\":\"monitoring-eye-css-*
\",\"default_timefield\":\"timestamp\",\"filter\":{\"language\":\"kuery\",\"query\":\"\"},\"id
\":\"61ca57f0-469d-11e7-af02-69e470af7417\",\"index_pattern\":\"monitoring-eye-css-*\",\"interval
\":\"\",\"isModelInvalid\":false,\"series\":[{\"axis_position\":\"right\",\"chart_type\":\"line\",\"color
\":\"rgba(231,102,76,1)\",\"fill\":0.5,\"formatter\":\"number\",\"id\":\"61ca57f1-469d-11e7-
af02-69e470af7417\",\"label\":\"segment count of index for top10\",\"line_width\":1,\"metrics\":
[{\"field\":\"index_stats.total.segments.count\",\"id\":\"61ca57f2-469d-11e7-
af02-69e470af7417\",\"type\":\"max\"}],\"point_size\":1,\"separate_axis\":0,\"split_color_mode
\":\"kibana\",\"split_mode\":\"terms\",\"stacked\":\"none\",\"terms_field\":\"index_stats.index
\",\"terms_order_by\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"timeseries
\"}],\"show_grid\":1,\"show_legend\":1,\"time_field\":\"timestamp\",\"tooltip_mode\":\"show_all
\",\"type\":\"timeseries\"},\"title\":\"[monitoring] segment count of index for top10\",\"type\":\"metrics
\"}"},"id":"45d571c0-6626-11ed-8cd7-973626cf6f70","migrationVersion":
{"visualization":"7.4.2"},"references":
[],"type":"visualization","updated_at":"2022-03-17T13:26:38.009Z","version":"WzE2LDFd"}
{"attributes":{"description":"","kibanaSavedObjectMeta":
{"searchSourceJSON":"{}"},"title":"[monitoring] segment memory in bytes of index for
top10","uiStateJSON":"{}","version":1,"visState":"{\"title\":\"[monitoring] segment memory in bytes of
index for top10\",\"type\":\"metrics\",\"aggs\":[],\"params\":{\"id\":\"61ca57f0-469d-11e7-
af02-69e470af7417\",\"type\":\"timeseries\",\"series\":[{\"id\":\"61ca57f1-469d-11e7-
af02-69e470af7417\",\"color\":\"#68BC00\",\"split_mode\":\"terms\",\"split_color_mode\":\"kibana
\",\"metrics\":[{\"id\":\"61ca57f2-469d-11e7-af02-69e470af7417\",\"type\":\"max\",\"field
\":\"index_stats.total.segments.memory_in_bytes\"}],\"separate_axis\":0,\"axis_position\":\"right
\",\"formatter\":\"bytes\",\"chart_type\":\"line\",\"line_width\":1,\"point_size\":1,\"fill\":0.5,\"stacked
\":\"none\",\"label\":\"segments memory in bytes \",\"type\":\"timeseries\",\"terms_field
\":\"index_stats.index\",\"terms_order_by\":\"61ca57f2-469d-11e7-af02-69e470af7417\"}],\"time_field
\":\"timestamp\",\"index_pattern\":\"monitoring-eye-css-*\",\"interval\":\"\",\"axis_position\":\"left
\",\"axis_formatter\":\"number\",\"axis_scale\":\"normal\",\"show_legend\":1,\"show_grid
\":1,\"tooltip_mode\":\"show_all\",\"default_index_pattern\":\"monitoring-eye-css-*
\",\"default_timefield\":\"timestamp\",\"isModelInvalid
\":false}}"},"id":"3ae5d820-6628-11ed-8cd7-973626cf6f70","migrationVersion":
{"visualization":"7.4.2"},"references":
[],"type":"visualization","updated_at":"2022-03-17T13:26:38.009Z","version":"WzE3LDFd"}
{"attributes":{"description":"","hits":0,"kibanaSavedObjectMeta":{"searchSourceJSON":"{\"query\":
{\"language\":\"kuery\",\"query\":\"\"},\"filter\":[]}"},"optionsJSON":"{\"hidePanelTitles
\":false,\"useMargins\":true}","panelsJSON":"[{\"version\":\"7.6.2\",\"gridData\":{\"x\":0,\"y\":0,\"w
\":48,\"h\":5,\"i\":\"971ed6c6-81b9-491b-9f08-e3ae9c382abd\"},\"panelIndex
\":\"971ed6c6-81b9-491b-9f08-e3ae9c382abd\",\"embeddableConfig\":{},\"panelRefName
\":\"panel_0\"},{\"version\":\"7.6.2\",\"gridData\":{\"x\":0,\"y\":5,\"w\":24,\"h\":15,\"i
\":\"5a6982e7-0c6c-4733-8a2d-e4c57cdf7397\"},\"panelIndex\":\"5a6982e7-0c6c-4733-8a2d-
e4c57cdf7397\",\"embeddableConfig\":{},\"panelRefName\":\"panel_1\"},{\"version
\":\"7.6.2\",\"gridData\":{\"x\":24,\"y\":5,\"w\":24,\"h\":15,\"i
\":\"662476f4-739c-4a05-858c-2ee8230cf410\"},\"panelIndex
\":\"662476f4-739c-4a05-858c-2ee8230cf410\",\"embeddableConfig\":{},\"panelRefName
\":\"panel_2\"},{\"version\":\"7.6.2\",\"gridData\":{\"x\":0,\"y\":20,\"w\":17,\"h\":17,\"i
\":\"d89c38e2-33f3-4592-b503-20460a6a7a57\"},\"panelIndex\":\"d89c38e2-33f3-4592-
b503-20460a6a7a57\",\"embeddableConfig\":{},\"panelRefName\":\"panel_3\"},{\"version
\":\"7.6.2\",\"gridData\":{\"x\":17,\"y\":20,\"w\":16,\"h\":17,\"i
\":\"1f693b49-79fa-4807-94e8-0c12f51e54f8\"},\"panelIndex
\":\"1f693b49-79fa-4807-94e8-0c12f51e54f8\",\"embeddableConfig\":{},\"panelRefName
\":\"panel_4\"},{\"version\":\"7.6.2\",\"gridData\":{\"x\":33,\"y\":20,\"w\":15,\"h\":17,\"i

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

\":\"616b143d-74e9-4dac-98ba-5849536f0fba\"},\"panelIndex
\":\"616b143d-74e9-4dac-98ba-5849536f0fba\",\"embeddableConfig\":{},\"panelRefName
\":\"panel_5\"},{\"version\":\"7.6.2\",\"gridData\":{\"x\":0,\"y\":37,\"w\":24,\"h\":11,\"i
\":\"cfa82f27-1b8d-49ba-a7b9-d8809d3b258c\"},\"panelIndex\":\"cfa82f27-1b8d-49ba-a7b9-
d8809d3b258c\",\"embeddableConfig\":{},\"panelRefName\":\"panel_6\"},{\"version
\":\"7.6.2\",\"gridData\":{\"x\":24,\"y\":37,\"w\":24,\"h\":11,\"i\":\"135d13eb-
aab6-43ca-9029-7d26e91d90e3\"},\"panelIndex\":\"135d13eb-
aab6-43ca-9029-7d26e91d90e3\",\"embeddableConfig\":{},\"panelRefName\":\"panel_7\"},{\"version
\":\"7.6.2\",\"gridData\":{\"x\":0,\"y\":48,\"w\":24,\"h\":11,\"i\":\"28a77de1-9110-49e8-
b273-724f880b1653\"},\"panelIndex\":\"28a77de1-9110-49e8-
b273-724f880b1653\",\"embeddableConfig\":{},\"panelRefName\":\"panel_8\"},{\"version
\":\"7.6.2\",\"gridData\":{\"x\":24,\"y\":48,\"w\":24,\"h\":11,\"i\":\"80ece867-cf23-4935-
bfbc-430afa51bcca\"},\"panelIndex\":\"80ece867-cf23-4935-bfbc-430afa51bcca\",\"embeddableConfig
\":{},\"panelRefName\":\"panel_9\"},{\"version\":\"7.6.2\",\"gridData\":{\"x\":0,\"y\":59,\"w\":24,\"h
\":11,\"i\":\"2ba970aa-c9c4-491b-bdd3-c1b1ee9bc8d3\"},\"panelIndex\":\"2ba970aa-c9c4-491b-bdd3-
c1b1ee9bc8d3\",\"embeddableConfig\":{},\"panelRefName\":\"panel_10\"},{\"version
\":\"7.6.2\",\"gridData\":{\"x\":24,\"y\":59,\"w\":24,\"h\":11,\"i\":\"f2e1b6ab-ddf7-492e-
aaca-9460f11aa4aa\"},\"panelIndex\":\"f2e1b6ab-ddf7-492e-aaca-9460f11aa4aa
\",\"embeddableConfig\":{},\"panelRefName\":\"panel_11\"},{\"version\":\"7.6.2\",\"gridData\":{\"x
\":0,\"y\":70,\"w\":24,\"h\":11,\"i\":\"dd14182d-d8b9-47f2-bf36-6cba3b09586c\"},\"panelIndex
\":\"dd14182d-d8b9-47f2-bf36-6cba3b09586c\",\"embeddableConfig\":{},\"panelRefName
\":\"panel_12\"},{\"version\":\"7.6.2\",\"gridData\":{\"x\":24,\"y\":70,\"w\":24,\"h\":11,\"i
\":\"a47f9333-52b7-49b7-8cac-f470cf405131\"},\"panelIndex\":\"a47f9333-52b7-49b7-8cac-
f470cf405131\",\"embeddableConfig\":{},\"panelRefName
\":\"panel_13\"}]","timeRestore":false,"title":"[Monitoring] Index monitoring
Dashboard","version":1},"id":"524eb000-a5f2-11ec-9a68-ada9d754c566","migrationVersion":
{"dashboard":"7.3.0"},"references":[{"id":"b2811c70-a5f1-11ec-9a68-
ada9d754c566","name":"panel_0","type":"visualization"},{"id":"de4f8ab0-a5f8-11ec-9a68-
ada9d754c566","name":"panel_1","type":"visualization"},{"id":"811df7a0-a5f8-11ec-9a68-
ada9d754c566","name":"panel_2","type":"visualization"},
{"id":"eea89780-664b-11ed-8cd7-973626cf6f70","name":"panel_3","type":"visualization"},
{"id":"cfbb4e20-664c-11ed-8cd7-973626cf6f70","name":"panel_4","type":"visualization"},
{"id":"c7f72ae0-664e-11ed-8cd7-973626cf6f70","name":"panel_5","type":"visualization"},
{"id":"943b3e00-a5ef-11ec-9a68-ada9d754c566","name":"panel_6","type":"visualization"},
{"id":"ab503550-a5ef-11ec-9a68-ada9d754c566","name":"panel_7","type":"visualization"},
{"id":"c78119a0-a5f5-11ec-9a68-ada9d754c566","name":"panel_8","type":"visualization"},
{"id":"225f6020-a5f1-11ec-9a68-ada9d754c566","name":"panel_9","type":"visualization"},
{"id":"17d49220-662a-11ed-8cd7-973626cf6f70","name":"panel_10","type":"visualization"},
{"id":"c8109100-6627-11ed-8cd7-973626cf6f70","name":"panel_11","type":"visualization"},
{"id":"45d571c0-6626-11ed-8cd7-973626cf6f70","name":"panel_12","type":"visualization"},
{"id":"3ae5d820-6628-11ed-8cd7-973626cf6f70","name":"panel_13","type":"visualization"}],"type":"das
hboard","updated_at":"2022-03-22T08:37:51.756Z","version":"WzE5LDFd"}
{"attributes":{"fields":"[{\"name\":\"_id\",\"type\":\"string\",\"esTypes\":[\"_id\"],\"count\":0,\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":false},{\"name\":\"_index
\",\"type\":\"string\",\"esTypes\":[\"_index\"],\"count\":0,\"scripted\":false,\"searchable
\":true,\"aggregatable\":true,\"readFromDocValues\":false},{\"name\":\"_score\",\"type\":\"number
\",\"count\":0,\"scripted\":false,\"searchable\":false,\"aggregatable\":false,\"readFromDocValues
\":false},{\"name\":\"_source\",\"type\":\"_source\",\"esTypes\":[\"_source\"],\"count\":0,\"scripted
\":false,\"searchable\":false,\"aggregatable\":false,\"readFromDocValues\":false},{\"name\":\"_type
\",\"type\":\"string\",\"esTypes\":[\"_type\"],\"count\":0,\"scripted\":false,\"searchable
\":true,\"aggregatable\":true,\"readFromDocValues\":false},{\"name\":\"cluster_uuid\",\"type\":\"string
\",\"esTypes\":[\"keyword\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"index_stats.created\",\"type\":\"string\",\"esTypes\":
[\"keyword\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"index_stats.index\",\"type\":\"string\",\"esTypes\":
[\"keyword\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"index_stats.primaries.docs.count\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"index_stats.primaries.docs.deleted\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"index_stats.primaries.fielddata.evictions\",\"type
\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"index_stats.primaries.fielddata.memory_size_in_bytes
\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable
\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.primaries.flush.total_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.primaries.indexing.delete_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

\":\"index_stats.primaries.indexing.delete_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.indexing.delete_total\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.indexing.index_current\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.indexing.index_failed\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.indexing.index_time_in_millis\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.primaries.indexing.index_total\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.indexing.throttle_time_in_millis\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.primaries.merges.current\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.merges.total_size_in_bytes\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.query_cache.cache_count\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.query_cache.cache_size\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.query_cache.evictions\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.query_cache.hit_count\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.query_cache.memory_size_in_bytes\",\"type\":\"number\",\"esTypes
\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"index_stats.primaries.query_cache.miss_count\",\"type
\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"index_stats.primaries.query_cache.total_count\",\"type
\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"index_stats.primaries.recovery.current_as_source
\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable
\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.primaries.recovery.current_as_target\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.recovery.throttle_time_in_millis\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.primaries.refresh.external_total_time_in_millis\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"index_stats.primaries.refresh.total_time_in_millis
\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable
\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.primaries.request_cache.evictions\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.primaries.request_cache.hit_count\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.primaries.request_cache.memory_size_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.primaries.request_cache.miss_count\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.primaries.search.fetch_current\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.primaries.search.fetch_time_in_millis\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.primaries.search.fetch_total\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.search.open_contexts\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.search.query_current\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.search.query_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.search.query_total\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.search.scroll_current\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.search.scroll_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.search.scroll_total\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.search.suggest_current\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.search.suggest_time_in_millis\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.primaries.search.suggest_total\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.primaries.segments.count\",\"type\":\"number\",\"esTypes\":[\"integer
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.segments.doc_values_memory_in_bytes\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.primaries.segments.fixed_bit_set_memory_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.primaries.segments.index_writer_memory_in_bytes\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"index_stats.primaries.segments.memory_in_bytes
\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable
\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.primaries.segments.norms_memory_in_bytes\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.primaries.segments.points_memory_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.primaries.segments.stored_fields_memory_in_bytes\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.primaries.segments.term_vectors_memory_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.primaries.segments.terms_memory_in_bytes\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.primaries.segments.version_map_memory_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.primaries.store.size_in_bytes\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.shards.active_primaries_shards\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.shards.active_replicas\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.shards.active_shards\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.shards.initializing_shards\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.shards.primaries_shards\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.shards.relocating_shards\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.shards.replicas_shards\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.shards.total_shards\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.shards.unassigned_shards\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.status\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"count\":0,\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.docs.count\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.docs.deleted\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.fielddata.evictions\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.fielddata.memory_size_in_bytes\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.flush.total_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.indexing.delete_current\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

{\"name\":\"index_stats.total.indexing.delete_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.indexing.delete_total\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.indexing.index_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.indexing.index_failed\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.indexing.index_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.indexing.index_total\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.indexing.throttle_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.merges.total_size_in_bytes\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.query_cache.cache_count\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.query_cache.cache_size\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.query_cache.evictions\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.query_cache.hit_count\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.query_cache.memory_size_in_bytes\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.query_cache.miss_count\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.query_cache.total_count\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.recovery.current_as_source\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.recovery.current_as_target\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.recovery.throttle_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.refresh.external_total_time_in_millis\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.total.refresh.total_time_in_millis\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.total.request_cache.evictions\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.request_cache.hit_count\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.request_cache.memory_size_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.total.request_cache.miss_count\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.total.search.fetch_current\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.search.fetch_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.search.fetch_total\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.search.open_contexts\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.search.query_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.search.query_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.search.query_total\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.search.scroll_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.search.scroll_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.search.scroll_total\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.search.suggest_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.search.suggest_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.search.suggest_total\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.segments.count\",\"type\":\"number\",\"esTypes\":[\"integer\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.segments.doc_values_memory_in_bytes\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.segments.fixed_bit_set_memory_in_bytes\",\"type\":\"number\",\"esTypes
\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.segments.index_writer_memory_in_bytes\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.segments.memory_in_bytes\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.total.segments.norms_memory_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.total.segments.points_memory_in_bytes\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.segments.stored_fields_memory_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.total.segments.term_vectors_memory_in_bytes\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"index_stats.total.segments.terms_memory_in_bytes
\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable
\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"index_stats.total.segments.version_map_memory_in_bytes\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"index_stats.total.store.size_in_bytes\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"index_stats.uuid\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"count\":0,\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"indices_stats._all.primaries.docs.count\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"indices_stats._all.primaries.docs.deleted\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"indices_stats._all.primaries.indexing.index_current\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"indices_stats._all.primaries.indexing.index_failed\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"indices_stats._all.primaries.indexing.index_time_in_millis\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"indices_stats._all.primaries.indexing.index_total\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"indices_stats._all.primaries.indexing.throttle_time_in_millis\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"indices_stats._all.primaries.search.fetch_current
\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable
\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"indices_stats._all.primaries.search.fetch_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"indices_stats._all.primaries.search.fetch_total\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"indices_stats._all.primaries.search.open_contexts\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"indices_stats._all.primaries.search.query_current\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"indices_stats._all.primaries.search.query_time_in_millis\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"indices_stats._all.primaries.search.query_total\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"indices_stats._all.primaries.search.scroll_current\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"indices_stats._all.primaries.search.scroll_time_in_millis\",\"type\":\"number
\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"indices_stats._all.primaries.search.scroll_total\",\"type
\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

\":true,\"readFromDocValues\":true},{\"name\":\"indices_stats._all.primaries.search.suggest_current
\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable
\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"indices_stats._all.primaries.search.suggest_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"indices_stats._all.primaries.search.suggest_total\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"indices_stats._all.total.docs.count\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"indices_stats._all.total.docs.deleted\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"indices_stats._all.total.indexing.index_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"indices_stats._all.total.indexing.index_failed\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"indices_stats._all.total.indexing.index_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"indices_stats._all.total.indexing.index_total\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"indices_stats._all.total.indexing.throttle_time_in_millis\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"indices_stats._all.total.search.fetch_current\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"indices_stats._all.total.search.fetch_time_in_millis\",\"type\":\"number\",\"esTypes
\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"indices_stats._all.total.search.fetch_total\",\"type
\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"indices_stats._all.total.search.open_contexts\",\"type
\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"indices_stats._all.total.search.query_current\",\"type
\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"indices_stats._all.total.search.query_time_in_millis
\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted\":false,\"searchable
\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"indices_stats._all.total.search.query_total\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"indices_stats._all.total.search.scroll_current\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count
\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"indices_stats._all.total.search.scroll_time_in_millis\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"indices_stats._all.total.search.scroll_total\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"indices_stats._all.total.search.suggest_current\",\"type\":\"number\",\"esTypes\":[\"long
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},
{\"name\":\"indices_stats._all.total.search.suggest_time_in_millis\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"indices_stats._all.total.search.suggest_total\",\"type\":\"number\",\"esTypes\":
[\"long\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true},{\"name\":\"interval\",\"type\":\"number\",\"esTypes\":[\"long\"],\"count\":0,\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"source_node.host\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"count\":0,\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"source_node.ip\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"count\":0,\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"source_node.name\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"count\":0,\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"source_node.timestamp\",\"type\":\"date\",\"esTypes\":[\"date\"],\"count\":0,\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"source_node.transport_address\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"count\":0,\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name
\":\"source_node.uuid\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"count\":0,\"scripted
\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name\":\"state_uuid
\",\"type\":\"string\",\"esTypes\":[\"keyword\"],\"count\":0,\"scripted\":false,\"searchable
\":true,\"aggregatable\":true,\"readFromDocValues\":true},{\"name\":\"timestamp\",\"type\":\"date
\",\"esTypes\":[\"date\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable
\":true,\"readFromDocValues\":true},{\"name\":\"type\",\"type\":\"string\",\"esTypes\":[\"keyword
\"],\"count\":0,\"scripted\":false,\"searchable\":true,\"aggregatable\":true,\"readFromDocValues
\":true}]","timeFieldName":"timestamp","title":"monitoring-eye-css-*"},"id":"153bc490-
a5a8-11ec-9a68-ada9d754c566","migrationVersion":{"index-pattern":"7.6.0"},"references":

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
[],"type":"index-pattern","updated_at":"2022-03-17T08:10:17.348Z","version":"WzE4LDFd"}
{"exportedCount":16,"missingRefCount":0,"missingReferences":[]}
```

2.12.4 Supported Metrics

You can use Cloud Eye to monitor CSS cluster metrics in real time and quickly handle exceptions.

Overview

This section describes CSS metrics that can be monitored by Cloud Eye as well as their namespaces and dimensions. You can query metrics and alerts generated by CSS on the **Cloud Eye console** or using an **API**.

NOTE

Cloud Eye can monitor dimensions (or objects) nested to a maximum depth of four levels (levels 0 to 3). 3 is the deepest level. For example, if the monitored dimension of a metric is **cluster_id,instance_id**, **cluster_id** indicates level 0 and **instance_id** indicates level 1.

Namespace

SYS.ES

Cluster Monitoring Metrics

NOTE

Accumulated value: The value is accumulated from when a node is started. After the node is restarted, the value is reset to zero and starts accumulating again.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-190 Elasticsearch cluster monitoring metrics

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
status	Cluster Health Status	Health status of the monitored object • **0**: The cluster is 100% available. • **1**: All primary shards are allocated, but some replicas are missing. There is an availability risk. Pay close attention to the cluster status. • **2**: Data is missing and the cluster cannot work properly. • **3**: Failed to obtain the cluster status.	0, 1, 2, 3	N/A	N/A	cluster_id	1 minute
disk_util	Max. Disk Usage	Maximum disk usage of nodes in a CSS cluster	0~100	%	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
max_jvm_heap_usage	Max. JVM Heap Usage	Maximum JVM heap usage of nodes in a CSS cluster.	0~100	%	N/A	cluster_id	1 minute
max_jvm_young_gc_time	Max. JVM Young GC Duration	Maximum accumulated JVM Young GC duration of nodes in a CSS cluster.	≥ 0	ms	N/A	cluster_id	1 minute
max_jvm_young_gc_count	Max. JVM Young GC Count	Maximum accumulated JVM Young GC count of nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_jvm_old_gc_time	Max. JVM Old GC Duration	Maximum accumulated JVM Old GC duration of nodes in a CSS cluster.	≥ 0	ms	N/A	cluster_id	1 minute
max_jvm_old_gc_count	Max. JVM Old GC Count	Maximum accumulated JVM Old GC count of nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
total_fs_size	Total Size of File Systems	Total size of file systems in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute
free_fs_size	Available Size of File Systems	Available size of file systems in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute
max_cpu_usage	Max. CPU Usage	Maximum node CPU usage in a CSS cluster.	0~100	%	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
max_cpu_time_of_jvm_process	Max. CPU Time of JVM Process	Maximum accumulated CPU usage duration of node JVM processes in a CSS cluster.	≥ 0	ms	N/A	cluster_id	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.	≥ 0	byte	1024	cluster_id	1 minute
max_current_opened_http_count	Current Max. Opened HTTP Connections	Maximum number of HTTP connections that are currently open for nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_total_opened_http_count	Total Max. Opened HTTP Connections	Maximum number of HTTP connections that were open for nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
indices_count	Indexes	Number of indexes in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute
total_shards_count	Shards	Number of shards in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute
primary_shards_count	Primary Shards	Number of primary shards in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
docs_count	Documents	Number of documents in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute
docs_deleted_count	Deleted Documents	Number of documents deleted in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute
nodes_count	Nodes	Number of nodes in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute
data_nodes_count	Data Nodes	The number of nodes that function as data nodes in a CSS cluster. This includes cold data nodes.	≥ 0	Count	N/A	cluster_id	1 minute
master_nodes_count	Master Nodes	The number of nodes that function as master nodes in a CSS cluster. Without dedicated master nodes, it indicates the total number of non-client nodes. With dedicated master nodes, it indicates the number of dedicated master nodes.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
ingest_nodes_count	Ingest Nodes	The number of nodes that function as ingest nodes in a CSS cluster. (Ingest nodes preprocess, clean, and transform data before indexing.)	≥ 0	Count	N/A	cluster_id	1 minute
max_load_average	Max. Node Load	Maximum number of average queued tasks per minute on nodes in a cluster	≥ 0	Count	N/A	cluster_id	1 minute
avg_cpu_usage	Avg. CPU Usage	Average node CPU usage in a CSS cluster.	0~100	%	N/A	cluster_id	1 minute
avg_load_average	Avg. Node Load	Average number of queued tasks per minute on nodes in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute
avg_jvm_heap_usage	Avg. JVM Heap Usage	Average node JVM heap usage in a CSS cluster.	0~100	%	N/A	cluster_id	1 minute
max_open_file_descriptors	Max. Open File Descriptors	Maximum number of node file descriptors that are currently open in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_open_file_descriptors	Avg. Open File Descriptors	Average number of node file descriptors that are currently open in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_max_file_descriptors	Max. Allowed File Descriptors	Maximum number of allowed node file descriptors in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_open_file_descriptors	Open File Descriptors	Number of node file descriptors that are currently open in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_write_queue	Tasks in Write Queue	Number of tasks in queue for the write thread pool.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_search_queue	Tasks in Search Queue	Total number of tasks in queue for the search thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_force_merge_queue	Tasks in Force Merge Queue	Total number of tasks in queue for the force merge thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_write_rejected	Rejected Tasks in Write Queue	Total number of tasks rejected by the write thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_search_rejected	Rejected Tasks in Search Queue	Total number of tasks rejected by the search thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_force_merge_rejected	Rejected Tasks in Force Merge Queue	Total number of tasks rejected by the force merge thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_write_queue	Max. Tasks in Write Queue	Maximum number of tasks in queue for the write thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_search_queue	Max. Tasks in Search Queue	Maximum number of tasks in queue for the search thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
max_thread_pool_force_merge_queue	Max. Tasks in Force Merge Queue	Maximum number of tasks in queue for the force merge thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_write_threads	Size of Write Thread Pool	Total size of the write thread pools on nodes in the CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_search_threads	Size of Search Thread Pool	Total size of the search thread pools on nodes in the CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_force_merge_threads	Size of Force Merge Thread Pool	Total size of the force merge thread pools on nodes in the CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_write_queue	Avg. Tasks in Write Queue	Average number of tasks in queue for the write thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_thread_pool_search_queue	Avg. Tasks in Search Queue	Average number of tasks in queue for the search thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_force_merge_queue	Avg. Tasks in Force Merge Queue	Average number of tasks in queue for the force merge thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_search_threads	Avg. Size of Search Thread Pool	Average size of the search thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_write_threads	Avg. Size of Write Thread Pool	Average size of the write thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_force_merge_threads	Avg. Size of Force Merge Thread Pool	Average size of the force merge thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_write_rejected	Avg. Rejected Tasks in Write Queue	Average number of tasks rejected by the write thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_flush_queue	Tasks in Flush Queue	Total number of tasks in queue for the flush thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_flush_rejected	Rejected Tasks in Flush Queue	Total number of tasks rejected by the flush thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_flush_queue	Max. Tasks in Flush Queue	Maximum number of tasks in queue for the flush thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_flush_threads	Size of Flush Thread Pool	Total size of the flush thread pools on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_flush_queue	Avg. Tasks in Flush Queue	Average number of tasks in queue for the flush thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_flush_threads	Avg. Size of Flush Thread Pool	Average size of the flush thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_generic_queue	Tasks in Generic Queue	Total number of tasks in queue for the generic thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_generic_rejected	Rejected Tasks in Generic Queue	Total number of tasks rejected by the generic thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_generic_queue	Max. Tasks in Generic Queue	Maximum number of tasks in queue for the generic thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_generic_threads	Size of Generic Thread Pool	Total size of the generic thread pools on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_generic_queue	Avg. Tasks in Generic Queue	Average number of tasks in queue for the generic thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_generic_threads	Avg. Size of Generic Thread Pool	Average size of the generic thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_management_queue	Tasks in Management Queue	Total number of tasks in queue for the management thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_management_rejected	Rejected Tasks in Management Queue	Total number of tasks rejected by the management thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_management_queue	Max. Tasks in Management Queue	Maximum number of tasks in queue for the management thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_management_threads	Size of Management Thread Pool	Total size of the management thread pools on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_management_queue	Avg. Tasks in Management Queue	Average number of tasks in queue for the management thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_thread_pool_management_threads	Avg. Size of Management Thread Pool	Average size of the management thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_refresh_queue	Tasks in Refresh Queue	Total number of tasks in queue for the refresh thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_refresh_rejected	Rejected Tasks in Refresh Queue	Total number of tasks rejected by the refresh thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_refresh_queue	Max. Tasks in Refresh Queue	Maximum number of tasks in queue for the refresh thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_refresh_threads	Size of Refresh Thread Pool	Total size of the refresh thread pools on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_refresh_queue	Avg. Tasks in Refresh Queue	Average number of tasks in queue for the refresh thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_thread_pool_refresh_threads	Avg. Size of Refresh Thread Pool	Average size of the refresh thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_searcher_queue	Tasks in OBS Searcher Queue	Total number of tasks in queue for the OBS Searcher thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_searcher_rejected	Rejected Tasks in OBS Searcher Queue	Total number of tasks rejected by the OBS Searcher thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_obs_searcher_queue	Max. Tasks in OBS Searcher Queue	Maximum number of tasks in queue for the OBS Searcher thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_searcher_threads	Size of OBS Searcher Thread Pool	Total size of the OBS Searcher thread pools on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_thread_pool_obs_searcher_queue	Avg. Tasks in OBS Searcher Queue	Average number of tasks in queue for the OBS Searcher thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_obs_searcher_threads	Avg. Size of OBS Searcher Thread Pool	Average size of the OBS Searcher thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_queue	Tasks in OBS Queue	Total number of tasks in queue for the OBS thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_rejected	Rejected Tasks in OBS Queue	Total number of tasks rejected by the OBS thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_obs_queue	Max. Tasks in OBS Queue	Maximum number of tasks in queue for the OBS thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_threads	Size of OBS Thread Pool	Total size of the OBS thread pools on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_thread_pool_obs_queue	Avg. Tasks in OBS Queue	Average number of tasks in queue for the OBS thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_obs_threads	Avg. Size of OBS Thread Pool	Average size of the OBS thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_upload_queue	Tasks in OBS Upload Queue	Total number of tasks in queue for the OBS Upload thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_upload_rejected	Rejected Tasks in OBS Upload Queue	Total number of tasks rejected by the OBS Upload thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_obs_upload_queue	Max. Tasks in OBS Upload Queue	Maximum number of tasks in queue for the OBS Upload thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_obs_upload_threads	Size of OBS Upload Thread Pool	Total size of the OBS Upload thread pools on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_obs_upload_queue	Avg. Tasks in OBS Upload Queue	Average number of tasks in queue for the OBS Upload thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_obs_upload_threads	Avg. Size of OBS Upload Thread Pool	Average size of the OBS Upload thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_download_queue	Tasks in OBS Download Queue	Total number of tasks in queue for the OBS Download thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_download_rejected	Rejected Tasks in OBS Download Queue	Total number of tasks rejected by the OBS Download thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
max_thread_pool_obs_download_queue	Max. Tasks in OBS Download Queue	Maximum number of tasks in queue for the OBS Download thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_download_threads	Size of OBS Download Thread Pool	Total size of the OBS Download thread pools on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_obs_download_queue	Avg. Tasks in OBS Download Queue	Average number of tasks in queue for the OBS Download thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_obs_download_threads	Avg. Size of OBS Download Thread Pool	Average size of the OBS Download thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
min_free_fs_size	Min. Available Storage Space	Minimum available storage space of nodes in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_jvm_old_gc_count	Avg. GCs of Old-Generation JVM	Average number of old-generation garbage collections of nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_jvm_old_gc_time	Avg. GC Duration of Old-Generation JVM	Average old-generation garbage collection duration of nodes in a CSS cluster.	≥ 0	ms	N/A	cluster_id	1 minute
avg_jvm_young_gc_count	Avg. GCs of Young-Generation JVM	Average number of young-generation garbage collections of nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_jvm_young_gc_time	Avg. GC Duration of Young-Generation JVM	Average young-generation garbage collection duration of nodes in a CSS cluster.	≥ 0	ms	N/A	cluster_id	1 minute
avg_max_file_descriptors	Avg. Maximum Allowed File Descriptors	Average value of the maximum number of allowed file descriptors on each node in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_mem_free_in_bytes	Avg. Available Memory	Average unused memory capacity of nodes in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute
avg_mem_free_percent	Avg. Available Memory Percentage	Average percentage of unused memory of nodes in a CSS cluster.	0~100	%	N/A	cluster_id	1 minute
avg_mem_used_in_bytes	Avg. Used Memory	Average used memory of nodes in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute
avg_mem_used_percent	Avg. Used Memory Percentage	Average percentage of used memory of nodes in a CSS cluster.	0~100	%	N/A	cluster_id	1 minute
max_mem_free_in_bytes	Max. Available Memory	Maximum unused memory of nodes in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute
max_mem_free_percent	Max. Available Memory Percentage	Maximum percentage of unused memory of nodes in a CSS cluster.	0~100	%	N/A	cluster_id	1 minute
max_mem_used_in_bytes	Max. Used Memory	Maximum used memory of nodes in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
max_mem_used_percent	Max. Used Memory Percentage	Maximum percentage of used memory of nodes in a CSS cluster.	0~100	%	N/A	cluster_id	1 minute
sum_jvm_old_gc_count	Total GCs of Old-Generation JVM	Number of old-generation garbage collections of nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_jvm_old_gc_time	Total GC Duration of Old-Generation JVM	Total old-generation garbage collection duration of nodes in the CSS cluster.	≥ 0	ms	N/A	cluster_id	1 minute
sum_jvm_young_gc_count	Total GCs of Young-Generation JVM	Number of young-generation garbage collections of nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_jvm_young_gc_time	Total GC Duration of Young-Generation JVM	Total young-generation garbage collection duration of nodes in the CSS cluster.	≥ 0	ms	N/A	cluster_id	1 minute
sum_current_opened_http_count	Currently Open HTTP Connections	Number of HTTP connections that are open on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_total_opened_http_count	Historical Open HTTP Connections	Number of HTTP connections that were open on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
IndexingLatency	Average Index Latency	Average time required for a shard to complete an indexing operation.	≥ 0	ms	N/A	cluster_id	1 minute
IndexingRate	Average Index Rate	Average number of index operations per second in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
SearchLatency	Average Search Latency	Average time required for a shard to complete a search operation.	≥ 0	ms	N/A	cluster_id	1 minute
SearchRate	Average QPS	Average queries per second (QPS) in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
task_max_running_time	Max. Task Runtime	Duration of the most time-consuming search or write task in the cluster.	≥ 0	ms	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
number_of_pending_tasks	Number of Pending Tasks in Queue	Number of pending tasks to be processed by the master node in the CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_disk_read_requests_rate	Total Disk Read IOPS	Number of disk read requests to all nodes in the CSS cluster per second.	≥ 0	request/s	N/A	cluster_id	1 minute
sum_disk_write_requests_rate	Total Disk Write IOPS	Number of disk write requests to all nodes in the CSS cluster per second.	≥ 0	request/s	N/A	cluster_id	1 minute
sum_disk_read_bytes_rate	Total Disk Read Rate	Number of bytes of data read from the disks of all nodes in the CSS cluster per second.	≥ 0	Byte/s	1024	cluster_id	1 minute
sum_disk_write_bytes_rate	Total Disk Write Rate	Number of bytes of data written into the disks of all nodes in the CSS cluster per second.	≥ 0	Byte/s	1024	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_vector_index_off_heap_used_in_bytes	Average Off-Heap Memory Usage of Vector Indexes	Average off-heap memory (bytes) used for querying vector indexes on each node in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute
avg_vector_index_off_heap_usage	Average Off-Heap Memory Utilization (%) of Vector Indexes	Average off-heap memory (in %) used for querying vector indexes on each node in a CSS cluster. This metric is calculated only when vector index queries occur.	0~100	%	N/A	cluster_id	1 minute
max_vector_index_off_heap_used_in_bytes	Maximum Off-Heap Memory Usage of Vector Indexes	Maximum off-heap memory (bytes) used for querying vector indexes on each node in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
max_vector_index_off_heap_usage	Maximum Off-Heap Memory Utilization (%) of Vector Indexes	Maximum off-heap memory (in %) used for querying vector indexes on each node in a CSS cluster. This metric is calculated only when vector index queries occur.	0~100	%	N/A	cluster_id	1 minute
vector_index_circuit_breaker_status	Vector Index Circuit Breaker Status	Circuit breaker status for vector indexes. ● 0: Not enabled. ● 1: Enabled. Check whether the cluster's off-heap memory usage exceeds the threshold.	0, 1	N/A	N/A	cluster_id	1 minute
request_count	Total Requests	Total number of requests sent to all nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
successfully_request_count	Successful requests	Number of requests successfully handled by all nodes in a CSS cluster. Status code: 200.	≥ 0	Count	N/A	cluster_id	1 minute
failed_request_count	Failed requests	Number of failed requests for all nodes in a CSS cluster. Status code: non-200.	≥ 0	Count	N/A	cluster_id	1 minute
limited_request_count	Throttled requests	Number of throttled requests for all nodes in a CSS cluster. Status code: 429.	≥ 0	Count	N/A	cluster_id	1 minute
not_found_request_count	Requests Not Found	Total occurrences of the Not Found error in a CSS cluster. Status code: 404. Only Elasticsearch 7.6.2 and 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
cold_data_storage	The size of cold data stored	The total size of cold data stored in a CSS cluster, which is the total size of the data of frozen indexes stored in an OBS bucket. This metric is measured hourly. Users cannot check this OBS bucket on the OBS console.	≥ 0	byte	1024	cluster_id	1 minute
number_of_index_creation_failures	Number of index creation failures	Number of index creation failures in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
shard_doc_exceed_threshold_count	Number of shards where the number of documents exceeds the threshold	Number of shards where the number of documents exceeds the threshold in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
write_fail_alias_error	Number of Write Failures Due to Alias Exceptions	Number of write failures caused by abnormal index aliases in a CSS cluster. Only Elasticsearch 7.6.2 and 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
write_fail_documents_count_exceed	Number of Write Failures Due to Excessive Documents	Number of write failures caused by too many index documents in a CSS cluster. Only Elasticsearch 7.6.2 and 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute
write_fail_fields_count_exceed	Number of Write Failures Due to Excessive Fields	Number of write failures caused by too many index fields in a CSS cluster. Only Elasticsearch 7.6.2 and 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute
write_fail_shards_count_exceed	Number of Write Failures Due to Excessive Shards	Number of write failures caused by too many index shards in a CSS cluster. Only Elasticsearch 7.6.2 and 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
write_fail_index_blocks_write	Number of Write Failures Due to Index Write Prohibition	Number of write failures caused by the setting of write prohibition in index configuration in a CSS cluster. Only Elasticsearch 7.6.2 and 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute
write_fail_mapping_malformed	Number of Write Failures Due to Incorrect Field Types	Number of write failures caused by incorrect index field types in a CSS cluster. Only Elasticsearch 7.6.2 and 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute
remote_sync_task_count	Number of Primary/ Secondary Synchronization Jobs	Number of primary/ secondary synchronization jobs for a CSS cluster. Only Elasticsearch 7.6.2 and 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
remote_sync_failed_task_count	Number of Failed Primary/Secondary Synchronization Jobs	Number of failed primary/secondary synchronization jobs for a CSS cluster. Only Elasticsearch 7.6.2 and 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute
remote_sync_diff_bytes_of_primary_shards	Size of Data to Be Synchronized via Primary/Secondary Synchronization Jobs for Primary Shards	Total size of data to be synchronized via primary/secondary synchronization jobs for all primary shards in a CSS cluster. Only Elasticsearch 7.6.2 and 7.10.2 clusters support this metric.	≥ 0	byte	1024	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
remote_sync_finished_bytes_of_primary_shards	Size of Data Synchronized via Primary/ Secondary Synchronization Jobs for Primary Shards	Total size of data already synchronized via primary/ secondary synchronization jobs for all primary shards in a CSS cluster. Only Elasticsearch 7.6.2 and 7.10.2 clusters support this metric.	≥ 0	byte	1024	cluster_id	1 minute
remote_sync_diff_docs_of_primary_shards	Number of Documents to Be Synchronized via Primary/ Secondary Synchronization Jobs for Primary Shards	Total number of documents to be synchronized via primary/ secondary synchronization jobs for all primary shards in a CSS cluster. Only Elasticsearch 7.6.2 and 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
remote_sync_diff_bytes	Size of Data to Be Synchronized via Primary/ Secondary Synchronization Jobs	Total size of data to be synchronized via primary/ secondary synchronization jobs for all shards in a CSS cluster. Only Elasticsearch 7.6.2 and 7.10.2 clusters support this metric.	≥ 0	byte	1024	cluster_id	1 minute
remote_sync_finished_bytes	Size of Data Synchronized via Primary/ Secondary Synchronization Jobs	Total size of data already synchronized via primary/ secondary synchronization jobs for all shards in a CSS cluster. Only Elasticsearch 7.6.2 and 7.10.2 clusters support this metric.	≥ 0	byte	1024	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
remote_sync_diff_docs	Number of Documents to Be Synchronized via Primary/ Secondary Synchronization Jobs	Total number of documents to be synchronized via primary/ secondary synchronization jobs for all shards in a CSS cluster. Only Elasticsearch 7.6.2 and 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute
index_storage	Index Data Volume	Total data size of all indexes in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute
backup_failed_task_count	Number of Failed Backup Tasks	Number of failed snapshot backup tasks in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
ism_fail_count	Number of Failed Index Lifecycle Tasks	Number of failed index lifecycle tasks in a CSS cluster. Only Elasticsearch 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute
disk_total	Total Disk Space	Total data disk capacity of all nodes in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
number_of_unassigned_shards	Number of Unallocated Shards	Total number of unallocated shards in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
shard_quota_usage	Shard Quota Usage	Ratio of allocated shards to the upper limit of the shard quantity.	0~100	%	N/A	cluster_id	1 minute
disk_iops_qos_num	Disk IOPS Overload Times	Total number of times the disk IOPS exceeds the predefined threshold.	≥ 0	Count	N/A	cluster_id	1 minute
disk_iobw_qos_num	Disk I/O Bandwidth Overload Times	Total number of times the disk I/O bandwidth usage exceeds the predefined threshold.	≥ 0	Count	N/A	cluster_id	1 minute
max_disk_io_util	Maximum Disk I/O Usage	Maximum disk I/O usage among all disks.	0~100	%	N/A	cluster_id	1 minute
avg_disk_io_util	Average Disk I/O Usage	Average disk I/O usage among all disks.	0~100	%	N/A	cluster_id	1 minute
network_bandwidth_in	Inbound Bandwidth	Total inbound traffic per second.	≥ 0	Byte/s	1024	cluster_id	1 minute
network_bandwidth_out	Outbound Bandwidth	Total outbound traffic per second.	≥ 0	Byte/s	1024	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_business_worker_queue	Total Queued Tasks in business_worker Thread Pool	Total number of tasks in queue for the business_worker thread pool on nodes in a cluster. Only Elasticsearch 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_business_worker_active	Total Ongoing Tasks in business_worker Thread Pool	Total number of ongoing tasks in the business_worker thread pool on nodes in a cluster. Only Elasticsearch 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_business_worker_queue	Max. Queued Tasks in business_worker Thread Pool	Maximum number of tasks in queue for the business_worker thread pool on nodes in a cluster. Only Elasticsearch 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
max_thread_pool_business_worker_active	Max. Ongoing Tasks in business_worker Thread Pool	Maximum number of ongoing tasks in the business_worker thread pool on nodes in a cluster. Only Elasticsearch 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_business_worker_queue	Avg. Queued Tasks in business_worker Thread Pool	Average number of tasks in queue for the business_worker thread pool on nodes in a cluster. Only Elasticsearch 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_business_worker_active	Avg. Ongoing Tasks in business_worker Thread Pool	Average number of ongoing tasks in the business_worker thread pool on nodes in a cluster. Only Elasticsearch 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_transport_worker_queue	Total Queued Tasks in transport_worker Thread Pool	Total number of tasks in queue for the transport_worker thread pool on nodes in a cluster. Only Elasticsearch 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_transport_worker_queue	Max. Queued Tasks in transport_worker Thread Pool	Maximum number of tasks in queue for the transport_worker thread pool on nodes in a cluster. Only Elasticsearch 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_transport_worker_queue	Avg. Queued Tasks in transport_worker Thread Pool	Average number of tasks in queue for the transport_worker thread pool on nodes in a cluster. Only Elasticsearch 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_flush_active	Total Ongoing Tasks in Flush Queue	Total number of ongoing tasks in the flush queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_flush_active	Max. Ongoing Tasks in Flush Queue	Maximum number of ongoing tasks in the flush queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_flush_active	Avg. Ongoing Tasks in Flush Queue	Average number of ongoing tasks in the flush queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_generic_active	Total Ongoing Tasks in Generic Queue	Total number of ongoing tasks in the generic queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_generic_active	Max. Ongoing Tasks in Generic Queue	Maximum number of ongoing tasks in the generic queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_generic_active	Avg. Ongoing Tasks in Generic Queue	Average number of ongoing tasks in the generic queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_management_active	Total Ongoing Tasks in Management Queue	Total number of ongoing tasks in the management queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_management_active	Max. Ongoing Tasks in Management Queue	Maximum number of ongoing tasks in the management queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_management_active	Avg. Ongoing Tasks in Management Queue	Average number of ongoing tasks in the management queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_refresh_active	Total Ongoing Tasks in Refresh Queue	Total number of ongoing tasks in the refresh queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_refresh_active	Max. Ongoing Tasks in Refresh Queue	Maximum number of ongoing tasks in the refresh queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_thread_pool_refresh_active	Avg. Ongoing Tasks in Refresh Queue	Average number of ongoing tasks in the refresh queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_active	Total Ongoing Tasks in OBS Queue	Total number of ongoing tasks in the OBS queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_obs_active	Max. Ongoing Tasks in OBS Queue	Maximum number of ongoing tasks in the OBS queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_obs_active	Avg. Ongoing Tasks in OBS Queue	Average number of ongoing tasks in the OBS queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_searcher_active	Total Ongoing Tasks in OBS Searcher Queue	Total number of ongoing tasks in the OBS Searcher queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
max_thread_pool_obs_searcher_active	Max. Ongoing Tasks in OBS Searcher Queue	Maximum number of ongoing tasks in the OBS Searcher queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_obs_searcher_active	Avg. Ongoing Tasks in OBS Searcher Queue	Average number of ongoing tasks in the OBS Searcher queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_write_active	Total Ongoing Tasks in Write Queue	Total number of ongoing tasks in the write queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_write_active	Max. Ongoing Tasks in Write Queue	Maximum number of ongoing tasks in the write queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_write_active	Avg. Ongoing Tasks in Write Queue	Average number of ongoing tasks in the write queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_search_active	Total Ongoing Tasks in Search Queue	Total number of ongoing tasks in the search queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_search_active	Max. Ongoing Tasks in Search Queue	Maximum number of ongoing tasks in the search queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_search_active	Avg. Ongoing Tasks in Search Queue	Average number of ongoing tasks in the search queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_force_merge_active	Total Ongoing Tasks in Force Merge Queue	Total number of ongoing tasks in the force merge queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_force_merge_active	Max. Ongoing Tasks in Force Merge Queue	Maximum number of ongoing tasks in the force merge queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_force_merge_active	Avg. Ongoing Tasks in Force Merge Queue	Average number of ongoing tasks in the force merge queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_get_queue	Tasks in GET Queue	Total number of tasks in queue for the GET thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_get_rejected	Rejected Tasks in GET Queue	Total number of tasks rejected by the GET thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_get_active	Total Ongoing Tasks in Get Queue	Total number of ongoing tasks in the GET thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_get_queue	Max. Tasks in GET Queue	Maximum number of tasks in queue for the GET thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_get_active	Max. Ongoing Tasks in GET Queue	Maximum number of ongoing tasks in the GET thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_thread_pool_get_queue	Avg. Tasks in GET Queue	Average number of tasks in queue for the GET thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_get_active	Avg. Ongoing Tasks in GET Queue	Average number of ongoing tasks in the GET thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_left_node_count	Disconnected Nodes	The number of nodes that are disconnected in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

Node Monitoring Metrics

Table 2-191 Elasticsearch cluster node monitoring metrics

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
jvm_heap_usage	JVM Heap Usage	JVM heap memory usage of a node.	0~100	%	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
cpu_usage	CPU Usage	CPU usage.	0~100	%	N/A	cluster_id,instance_id	1 minute
disk_util	Max. Disk Usage	Disk usage of a node	0~100	%	N/A	cluster_id,instance_id	1 minute
load_average	Average Load	Average number of queued tasks per minute on a node	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
open_file_descriptors	Open File Descriptors	Number of opened file descriptors on a node	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
max_file_descriptors	Max. Allowed File Descriptors	Maximum number of allowed file descriptors	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_write_queue	Tasks in Write Queue	Number of tasks in queue for the write thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_search_queue	Tasks in Search Queue	Number of tasks in queue for the search thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_force_merge_queue	Tasks in Force Merge Queue	Number of tasks in queue for the force merge thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
thread_pool_write_rejected	Rejected Tasks in Write Queue	Number of tasks rejected by the write thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_search_rejected	Rejected Tasks in Search Queue	Number of tasks rejected by the search thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_force_merge_rejected	Rejected Tasks in Force Merge Queue	Number of tasks rejected by the force merge thread pool	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_write_threads	Size of Write Thread Pool	Size of a write thread pool	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_search_threads	Size of Search Thread Pool	Size of a search thread pool	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_force_merge_threads	Size of Force Merge Thread Pool	Size of a force merge thread pool	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_flush_queue	Tasks in Flush Queue	Number of tasks in queue for the flush thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_flush_rejected	Rejected Tasks in Flush Queue	Number of tasks rejected by the flush thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_flush_threads	Size of Flush Thread Pool	Size of the flush thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
thread_pool_generic_queue	Tasks in Generic Queue	Number of tasks in queue for the generic thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_generic_rejected	Rejected Tasks in Generic Queue	Number of tasks rejected by the generic thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_generic_threads	Size of Generic Thread Pool	Size of the generic thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_management_queue	Tasks in Management Queue	Number of tasks in queue for the management thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_management_rejected	Rejected Tasks in Management Queue	Number of tasks rejected by the management thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_management_threads	Size of Management Thread Pool	Size of the management thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_refresh_queue	Tasks in Refresh Queue	Number of tasks in queue for the refresh thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_refresh_rejected	Rejected Tasks in Refresh Queue	Number of tasks rejected by the refresh thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
thread_pool_refresh_threads	Size of Refresh Thread Pool	Size of the refresh thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_searcher_queue	Tasks in OBS Searcher Queue	Number of tasks in queue for the OBS Searcher thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_searcher_rejected	Rejected Tasks in OBS Searcher Queue	Number of tasks rejected by the OBS Searcher thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_searcher_threads	Size of OBS Searcher Thread Pool	Size of the OBS Searcher thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_queue	Tasks in OBS Queue	Number of tasks in queue for the OBS thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_rejected	Rejected Tasks in OBS Queue	Number of tasks rejected by the OBS thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_threads	Size of OBS Thread Pool	Size of the OBS thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_upload_queue	Tasks in OBS Upload Queue	Number of tasks in queue for the OBS Upload thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
thread_pool_obs_upload_rejected	Rejected Tasks in OBS Upload Queue	Number of tasks rejected by the OBS Upload thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_upload_threads	Size of OBS Upload Thread Pool	Size of the OBS Upload thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_download_queue	Tasks in OBS Download Queue	Number of tasks in queue for the OBS Download thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_download_rejected	Rejected Tasks in OBS Download Queue	Number of tasks rejected by the OBS Download thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_download_threads	Size of OBS Download Thread Pool	Size of the OBS Download thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
free_fs_size	Available Size of File Systems	Available size of file systems in a CSS cluster.	≥ 0	byte	1024	cluster_id,instance_id	1 minute
total_fs_size	Total Size of File Systems	Total size of file systems in a CSS cluster.	≥ 0	byte	1024	cluster_id,instance_id	1 minute
jvm_old_gc_count	Total GCs of Old-Generation JVM	Number of old-generation garbage collection times	≥ 0	Count	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
jvm_old_gc_time	Total GC Duration of Old-Generation JVM	Time spent on old-generation garbage collection	≥ 0	ms	N/A	cluster_id,instance_id	1 minute
jvm_young_gc_count	Total GCs of Young-Generation JVM	Number of young-generation garbage collection times	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
jvm_young_gc_time	GC Duration of Young-Generation JVM	Time spent on young-generation garbage collection	≥ 0	ms	N/A	cluster_id,instance_id	1 minute
mem_free_in_bytes	Available Memory	Unused memory space of a node.	≥ 0	byte	1024	cluster_id,instance_id	1 minute
mem_free_percent	Available Memory Percentage	Percentage of unused memory space on a node.	0~100	%	N/A	cluster_id,instance_id	1 minute
mem_used_in_bytes	Used Memory	Used memory space of a node.	≥ 0	byte	1024	cluster_id,instance_id	1 minute
current_opened_http_count	Currently Open HTTP Connections	Number of HTTP connections currently open on a node	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
total_opened_http_count	Total Open HTTP Connections	Total number of HTTP connections on a node	≥ 0	Count	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
disk_read_requests_rate	Disk Read IOPS	Number of disk read requests to a node in the CSS cluster per second.	≥ 0	request/s	N/A	cluster_id,instance_id	1 minute
disk_write_requests_rate	Disk Write IOPS	Number of disk write requests to a node in the CSS cluster per second.	≥ 0	request/s	N/A	cluster_id,instance_id	1 minute
disk_read_bytes_rate	Disk Read Rate	Number of bytes of data read from the disks of a node in the CSS cluster per second.	≥ 0	Byte/s	1024	cluster_id,instance_id	1 minute
disk_write_bytes_rate	Disk Write Rate	Number of bytes of data written into the disks of a node in the CSS cluster per second.	≥ 0	Byte/s	1024	cluster_id,instance_id	1 minute
shards_count	Shards	Number of shards per node in the CSS cluster.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
shard_quota_usage	Shard Quota Usage	Ratio of allocated shards to the upper limit of the shard quantity.	0~100	%	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
vector_index_off_heap_used_in_bytes	Off-Heap Memory Usage of Vector Indexes	Off-heap memory (bytes) used for querying vector indexes on nodes in a CSS cluster.	≥ 0	byte	1024	cluster_id,instance_id	1 minute
vector_index_off_heap_usage	Off-Heap Memory Utilization (%) of Vector Indexes	Off-heap memory (in %) used for querying vector indexes on nodes in a CSS cluster.	0~100	%	N/A	cluster_id,instance_id	1 minute
disk_iops_qos_num	Disk IOPS Overload Times	Total number of times the disk IOPS exceeds the predefined threshold.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
disk_iobw_qos_num	Disk I/O Bandwidth Overload Times	Total number of times the disk I/O bandwidth usage exceeds the predefined threshold.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
max_disk_io_util	Maximum Disk I/O Usage	Maximum disk I/O usage among all disks.	0~100	%	N/A	cluster_id,instance_id	1 minute
avg_disk_io_util	Average Disk I/O Usage	Average disk I/O usage among all disks.	0~100	%	N/A	cluster_id,instance_id	1 minute
network_bandwidth_in	Inbound Bandwidth	Total inbound traffic per second on the VM.	≥ 0	Byte/s	1024	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
network_bandwidth_out	Outbound Bandwidth	Total outbound traffic per second on the VM.	≥ 0	Byte/s	1024	cluster_id,instance_id	1 minute
thread_pool_business_worker_queue	Queued Tasks in business_worker Thread Pool	Number of tasks in queue for the business_worker thread pool on the node. Only Elasticsearch 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_business_worker_active	Ongoing Tasks in business_worker Thread Pool	Number of ongoing tasks in the business_worker thread pool on the node. Only Elasticsearch 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_transport_worker_queue	Queued Tasks in transport_worker Thread Pool	Number of tasks in queue for the transport_worker thread pool on the node. Only Elasticsearch 7.10.2 clusters support this metric.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
node_indexing_rate	Node Indexing Rate	Indexing rate on the node (average number of indexing operations per second).	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
node_search_rate	Node Query Rate	Query rate on the node (average number of queries per second).	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
node_get_rate	Node Get Rate	Get rate on the node (average number of get requests per second).	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_flush_active	Ongoing Tasks in Flush Queue	Number of ongoing tasks in the Flush queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_generic_active	Ongoing Tasks in Generic Queue	Number of ongoing tasks in the generic queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_management_active	Ongoing Tasks in Management Queue	Number of ongoing tasks in the management queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_refresh_active	Ongoing Tasks in Refresh Queue	Number of ongoing tasks in the refresh queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_active	Ongoing Tasks in OBS Queue	Number of ongoing tasks in the OBS queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
thread_pool_obs_searcher_active	Ongoing Tasks in OBS Searcher Queue	Number of ongoing tasks in the OBS Searcher queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_write_active	Ongoing Tasks in Write Queue	Number of ongoing tasks in the write queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_search_active	Ongoing Tasks in Search Queue	Number of ongoing tasks in the search queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_force_merge_active	Ongoing Tasks in Force Merge Queue	Number of ongoing tasks in the Force Merge queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_get_rejected	Rejected Tasks in GET Queue	Number of rejected tasks in the GET queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_get_queue	Tasks in GET Queue	Number of queued tasks in the GET queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_get_active	Ongoing Tasks in GET Queue	Number of ongoing tasks in the GET queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
rss_usage	RSS Memory Usage	The ratio of RSS memory to total physical memory. Only Elasticsearch 7.10.2 clusters support this metric.	0~100	%	N/A	cluster_id,instance_id	1 minute

If an object is in a hierarchical system, specify the monitored dimension in hierarchical form when you use an API to query the metrics of this object.

For example, to query the CPU usage (cpu_usage) of an Elasticsearch cluster node in CSS, the dimension information of this metric is **cluster_id,instance_id**, where **cluster_id** indicates level 0 and **instance_id** level 1.

- To query a single metric by calling an API, the **instance_id** dimension is used as follows:
  ```
  dim.0=cluster_id,3d65c1ac-9a9f-4c5f-a054-35184a087bb2&dim.1=instance_id,6666cd76f96956469e7be39d750cc7d9
  ```
  where, **3d65c1ac-9a9f-4c5f-a054-35184a087bb2** and **6666cd76f96956469e7be39d750cc7d9** are the values of **cluster_id** and **instance_id**, respectively. For how to obtain these values, see **Dimensions**.
- To query multiple metrics by calling an API, the **instance_id** dimension is used as follows:
  ```
  "dimensions": [
    {
      "name": "cluster_id",
      "value": "3d65c1ac-9a9f-4c5f-a054-35184a087bb2"
    },
    {
      "name": "instance_id",
      "value": "6666cd76f96956469e7be39d750cc7d9"
    }
  ]
  ```
  where, **3d65c1ac-9a9f-4c5f-a054-35184a087bb2** and **6666cd76f96956469e7be39d750cc7d9** are the values of **cluster_id** and **instance_id**, respectively. For how to obtain the values, see **Dimensions**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Dimensions

Table 2-192 Dimension description

Key	Value
cluster_id	Cluster ID. You can obtain this value from the **clusters[].id** field in the response body of the **Querying the Cluster List** API.
cluster_id,instance_id	Cluster node ID. You can obtain this value from the **clusters[].instances[].id** field in the response body of the **Querying the Cluster List** API.

2.13 Managing Logs

2.13.1 Querying Logs

During routine O&M, when an Elasticsearch cluster becomes sluggish, returns an error for a query request, or shows a yellow status, the O&M team must be able to obtain relevant information immediately. CSS provides a log query feature that aggregates underlying server logs to the console, where you can query them by node and log level in seconds. By analyzing run logs, slow query logs, and deprecation logs, you can quickly locate the root causes of faults and issues. Additionally, you can dynamically adjust log levels and enable trace logging for debugging, allowing you to obtain the most detailed log records without restarting your cluster.

Querying Recent Logs

Query recently generated logs that are not yet archived.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Logs > Log Search**. The **Log Search** page is displayed.

   You can search records by log type, node, log level, or keyword. For a detailed description of each type of logs, see **Log Types**.

> ⚠ CAUTION
>
> When a log file reaches 128 MB or when the time reaches 00:00 UTC, the system automatically compresses and archives it. Only unarchived logs appear on the log search page, while archived logs remain accessible through the log backup function. For more information, see **Backing Up Logs to OBS**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Adjusting Log Levels Dynamically

Log4j2 is used as the log component in Elasticsearch clusters. Multiple log levels (ERROR, WARN, INFO, DEBUG, and TRACE) are supported. The default log level is INFO. To facilitate troubleshooting and debugging, you can dynamically adjust the log levels.

- INFO is the default log level. The levels, in order of increasing detail, are: ERROR, WARN, INFO, DEBUG, and TRACE. When INFO is set, you will see logs for itself and all higher-severity levels (ERROR and WARN), while more verbose levels (DEBUG and TRACE) are excluded.
- You can change the log level of a specified module in real time via an Elasticsearch API.

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

      The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.
2. Run the following command to change the log level, for example, to DEBUG:

```
PUT _cluster/settings
{
  "persistent": {
    "logger": {
      "org.elasticsearch.action": "DEBUG"
    }
  }
}
```

3. Refresh the log query page and check whether DEBUG logs have been generated. You should always restore the log level to the default setting once you finish the task at hand.
4. Run the following command to restore the default log level **INFO**:

```
PUT _cluster/settings
{
  "persistent": {
    "logger": {
      "org.elasticsearch.action": null
    }
  }
}
```

Enabling Trace Logging

To analyze communication details on the HTTP or transport layer, you can enable trace logging for the HTTP or Transport module to obtain detailed log records.

1. Log in to Kibana and go to the command execution page. Elasticsearch clusters support multiple access methods. This topic uses Kibana as an example to describe the operation procedures.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

a. Log in to the **CSS management console**.
b. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
c. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
d. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

2. Run the following command to enable trace logging:

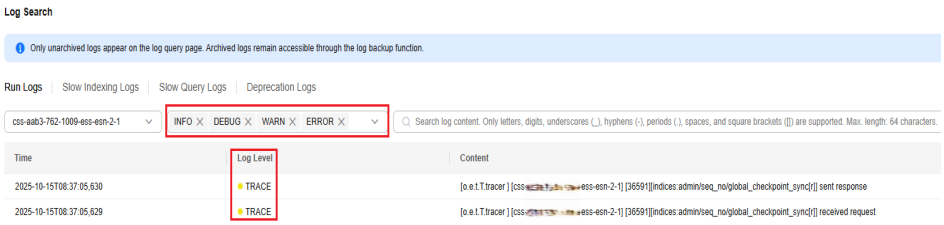
```
PUT _cluster/settings
{
  "transient": {
    "logger.org.elasticsearch.transport.TransportService.tracer": "trace",
    "transport.tracer.include": "",
    "http.tracer.include": "",
    "logger.org.elasticsearch.http.HttpTracer": "trace"
  }
}
```

CAUTION

Enabling trace logging is a non-persistent configuration and will be disabled upon a cluster restart. It is typically used for emergency packet capture and analysis during troubleshooting.

3. Go to the log details page to view trace logs.
   a. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
   b. Choose **Logs > Log Search**. The **Log Search** page is displayed.
   c. Select all log levels (mandatory) and view trace logs.

Figure 2-57 Viewing trace logs



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Log Types

Table 2-193 Introducing different log types

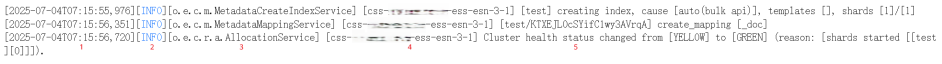
Log Type	Description	Purpose
Run logs	Run logs, or main logs, record the cluster status and key information about write and query operations. For example, write logs record operations such as index creation, index mapping update, and write queue exhaustion; and query logs record query queue status and query exceptions.	Check the status and write and query operations of each cluster node, including inter-node connectivity, full GC, index creation or deletion, and cluster-level query errors.
Slow indexing logs	Slow indexing logs record indexing operations (such as bulk, index, update, and delete) that took a long time to complete, helping you identify performance bottlenecks.	In the case of slow write performance, you can query slow indexing logs to locate the cause.
Slow query logs	Slow query logs record search requests that took a long time to complete. They help you monitor and analyze time-consuming search requests, so you can identify performance bottlenecks, optimize SQL queries, and improve overall system performance.	In the case of slow query performance, you can query slow query logs to locate the cause.
Deprecation logs	Deprecation logs record deprecation warnings. Deprecation warnings are written to this log when you use APIs, configurations, or functions that are marked for removal in future versions.	Check for APIs or features that are about to expire in future versions.
Access logs	Access logs record cluster access requests, such as the request path and source address. You cannot check access logs on the console. To check them, you need to back them up to an OBS bucket or transfer them to a target cluster first.	If there is a surge in service requests, you can analyze the request sources and paths by checking the access logs.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- **Run log description**

  Run logs record the cluster status and key information about write and query operations. For example, the log record below indicates that an index named **test** was created and later the cluster status changed from **YELLOW** to **GREEN**.

  Figure 2-58 A sample of run logs

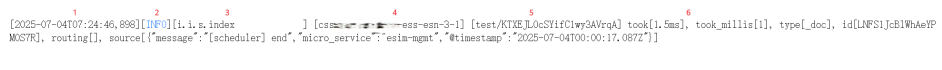
  Log content:

  - 1. Log generation time
  - 2. Log level, which can be DEBUG, INFO, WARN, or ERROR
  - 3. Log-generating module
  - 4. Name of the log-generating node
  - 5. Log content

- **Slow indexing log description**

  Slow indexing logs record indexing operations that took a long time to complete. For example, the log record below shows an indexing request that lasted longer than the preset threshold. The log contains the index name, duration, and request content.

  Figure 2-59 A sample of slow indexing logs

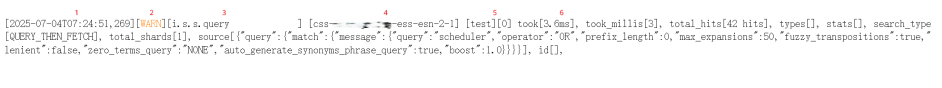
  Log content:

  - 1. Log generation time
  - 2. Log level, which can be DEBUG, INFO, WARN, or ERROR
  - 3. Log-generating module
  - 4. Name of the log-generating node
  - 5. Index name and ID
  - 6. Log content. In this example, the log recorded the request execution duration, index type, and index request body.

- **Slow query log description**

  Slow query logs record search requests that took a long time to complete. For example, the log record below shows a search request that lasted longer than the preset threshold. The log contains the index name, duration, and request content.

  Figure 2-60 A sample of slow query logs

  Log content:

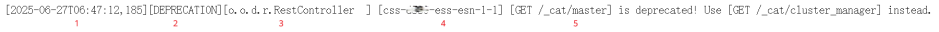
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

  - 1. Log generation time
  - 2. Log level, which can be DEBUG, INFO, WARN, or ERROR
  - 3. Log-generating module
  - 4. Name of the log-generating node
  - 5. Index name and shard ID
  - 6. Log content. In this example, the log recorded the query duration, number of hits, and query request body.

- **Deprecation log description**

  Deprecation logs record deprecation warnings. For example, the log record below indicates that **GET /_cat/master** has been deprecated and should be replaced with **GET /_cat/cluster_manager**.

  Figure 2-61 A sample of deprecation logs

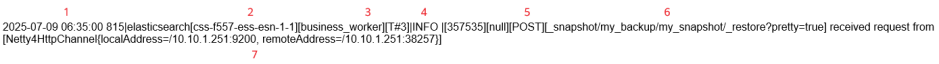
  Log content:

  - 1. Log generation time
  - 2. Log level, which can only be DEPRECATION.
  - 3. Log-generating module
  - 4. Name of the log-generating node
  - 5. Log content

- **Access log description**

  Access logs record cluster access requests and source addresses. For example, the log record below has recorded source information for the /_snapshot/my_backup/my_snapshot/_restore?pretty=true operation.

  Figure 2-62 A sample of access logs

  Log content:

  - 1. Log generation time
  - 2. Name of the log-generating node
  - 3. Name of the log-generating thread
  - 4. Log level, which can be DEBUG, INFO, WARN, or ERROR
  - 5. Log request method
  - 6. Request path
  - 7. Source and destination addresses of the request

Related Documents

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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2.13.2 Ingesting Logs into a Cluster

For large clusters that generate high volumes of logs, traditional search methods may struggle to extract comprehensive insights from the log data. To enable more advanced log analysis, you can leverage CSS's log ingestion feature to ingest logs generated by your cluster in real-time into a designated index in a specified cluster. Tools like Kibana can then be used to visualize and analyze the logs. By converting unstructured log data into searchable indexes, you can perform centralized log analysis across multiple clusters, greatly enhancing the efficiency of troubleshooting complex issues.

Constraints

Only Elasticsearch clusters whose image version is 7.10.2_24.2.0_xxx or later supports log ingestion.

Enabling Real-Time Log Ingestion

Log ingestion converts unstructured log data into structured documents stored in searchable Elasticsearch indexes.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Logs > Log Ingestion**. The **Log Ingestion** page is displayed.

   CAUTION

   If the **Log Ingestion** tab is not displayed, the cluster does not support log ingestion.

5. Enable log ingestion. If it is already enabled, skip this step.
   a. Click **Enable Log Ingestion**. In the displayed dialog box, configure necessary settings.

      Table 2-194 Log ingestion settings

Parameter	Description
Index Prefix Name	If you set a prefix for the log file indexes, the log index names will use the format **index prefix name + log ingestion date**. The unit of log ingestion is days. An index prefix name is a string of 1 to 128 characters. It can only contain digits, lowercase letters, underscores (_), and hyphens (-).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Retention Period	Set the log retention period. Ingested logs will be deleted upon expiration of this period. Value range: 1 to 3650 Unit: days
Log Storage Cluster	Select a cluster to store ingested logs. Options include **Current cluster** and **Other clusters**. • **Current cluster**: Store ingested logs in the current cluster. • **Other clusters**: Store ingested logs in another cluster. Select a cluster from the **Cluster** drop-down list to use as the log storage destination. Make sure this cluster is in the same VPC and have the same version as the current cluster. When you select another cluster, click **Check** to verify network connectivity between the current cluster and the selected other cluster. Log ingestion can start only when "The current cluster is available" is returned.

b. Click **OK** to enable log ingestion.

The configuration information will be displayed on the **Log Ingestion** tab.

6. On the **Log Ingestion** tab, if the task status changes to **Running**, log ingestion has started.

Viewing Ingested Logs

On the **Log Ingestion** tab, click **Access Kibana**. The Kibana login page for the log storage cluster is displayed. Kibana's powerful visualization features simplify log search and analytics.

Figure 2-63 Viewing ingested logs



To view details about the log storage cluster, click the cluster name in cluster list to go to the cluster overview page.

Disabling Log Ingestion

If you no longer need the log ingestion feature, you can disable it to reclaim storage resources.

On the **Log Ingestion** page, click **Disable Log Ingestion**. In the displayed dialog box, click **OK**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

> **CAUTION**
>
> Disabling log ingestion does not automatically remove ingested log data. Rather, it is deleted by the system upon expiration of its retention period. You can also manually delete it from the log storage cluster before the retention period expires.

2.13.3 Backing Up Logs to OBS

During routine O&M, querying logs in real-time through the web console is a commonly used method for diagnosing issues. However, due to limited local storage capacity, the system automatically archives historical logs when they exceed 128 MB or at 00:00 UTC. Long-term log retention is essential for compliance and operational purposes, but it must also be cost-effective. To address this, CSS provides a log backup feature that incrementally synchronizes run logs, slow query logs, and deprecation logs to OBS based on predefined policies. This solution takes advantage of OBS's vast, low-cost storage for long-term retention while also allowing users to download historical log files at any time for detailed offline analysis.

Impact on Billing

Standard storage is used for storing log backups in OBS. For pricing details, see **Billing**section "Billing" in *Object Storage Service User Guide*.

Prerequisites

You have created an OBS bucket for storing log backups. The OBS bucket must meet the following requirements:

- **Storage Class**: **Standard**.
- **Region**: the same as that of your CSS cluster.

Enabling Log Backup

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Logs > Log Backup**. The **Log Backup** page is displayed.

   Check whether log backup is enabled for the cluster.

   - Yes: No further action is required.
   - No: Go to the next step.
5. On the **Log Backup** page, click **Enable Backup**. In the displayed dialog box, configure the necessary settings.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-195 Log backup settings

Parameter	Description

OBS Bucket	Select an OBS bucket for storing log backups from the drop-down list box.
	If no OBS buckets meet your requirements, click **Create Bucket** to go to the OBS console and create one. For details, see **Creating a Bucket**.
	An IAM user must be granted the following OBS permissions to view available OBS buckets:
	`obs:bucket:getBucketStoragePolicy`
	`obs:bucket:listAllMyBuckets`

Backup Path	Set the log storage location in the OBS bucket. The backup path cannot: • Contain the following characters: **\:*?"<>|'{}** • Start with a slash (/). • Start or end with a period (.). • Contain more than two consecutive slashes (/) or periods (.). • Exceed 512 characters.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
IAM Agency	To back up data to an OBS bucket, you must have the write permission to it. By configuring an IAM agency, you can authorize CSS to access its OBS resources through an associated account. • If you are configuring an agency for the first time, click **Automatically Create IAM Agency** to create **css-obs-agency**. • If there is an IAM agency automatically created earlier, you can click **One-click authorization** to have the OBS Administrator permissions deleted automatically, and have the following custom policies added automatically instead to implement more refined permissions control. "obs:bucket:GetBucketLocation", "obs:object:GetObjectVersion", "obs:object:GetObject", "obs:object:DeleteObject", "obs:bucket:HeadBucket", "obs:bucket:GetBucketStoragePolicy", "obs:object:DeleteObjectVersion", "obs:bucket:ListBucketVersions", "obs:bucket:ListBucket", "obs:object:PutObject" • When OBS buckets use SSE-KMS encryption, the IAM agency must be granted KMS permissions. You can click **Automatically Create IAM Agency** and **One-click authorization** to have the following custom policies created automatically. "kms:cmk:create", "kms:dek:create", "kms:cmk:get", "kms:dek:decrypt", "kms:cmk:list" • To use **Automatically Create IAM Agency** and **One-click authorization**, the following minimum permissions are required: "iam:agencies:listAgencies", "iam:roles:listRoles", "iam:agencies:getAgency", "iam:agencies:createAgency", "iam:permissions:listRolesForAgency", "iam:permissions:grantRoleToAgency", "iam:permissions:listRolesForAgencyOnProject", "iam:permissions:revokeRoleFromAgency", "iam:roles:createRole" • To use an IAM agency, the following minimum permissions are required: "iam:agencies:listAgencies", "iam:agencies:getAgency", "iam:permissions:listRolesForAgencyOnProject", "iam:permissions:listRolesForAgency" **WARNING** The agency name can contain only letters (case-sensitive), digits, underscores (_), and hyphens (-). Otherwise, the backup will fail.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Determine whether to enable **Automatic backup** based on service needs.

- To enable automatic, periodic log backup, select **Automatic backup**, and set the backup policy. For more, see **Configuring Automatic Log Backup**.
- If manual log backup is sufficient, deselect **Automatic backup**.

6. Click **OK** to enable log backup.

   The configuration information will be displayed on the **Log Backup** tab.

Configuring Automatic Log Backup

After enabling log backup for a cluster, you can configure an auto backup policy to periodically back up logs based on a preset schedule.

1. On the **Log Backup** page, click **Modify Settings** on the right.

   Figure 2-64 Modifying Settings

2. In the **Modify Settings** dialog box, select **Automatic backup**, and configure the backup policy.

   Table 2-196 Automatic backup settings

Parameter	Description
Time Zone	Select a time zone for the backup start time.
Backup Start Time	Specify the start time of auto backup. Select a value from the drop-down list. The value range is from **00:00** to **23:00**. The backup always happens on the hour.

3. Click **OK** to enable automatic log backup.

   Automatic log backup tasks are displayed in the task list. When **Task Status** in the task list changes to **Succeeded**, the backup is successful.

4. If automatic log backup is no longer required, click **Modify Settings**. In the displayed dialog box, deselect **Automatic backup**, and click **OK**. In the displayed confirmation dialog box, click **OK**.

   Disabling log backup does not automatically delete log backups automatically generated earlier. Instead, you need to manually delete them on the OBS console.

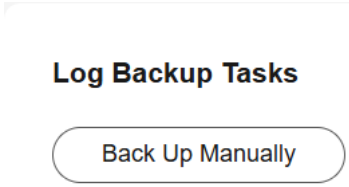
Manually Backing Up Logs

After enabling log backup for a cluster, you can manually back up its logs at any time.

1. On the **Log Backup** page, click **Back Up Manually** under **Log Backup Tasks**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 2-65 Manually backing up logs



2. In the displayed dialog box, confirm the log backup path and click **OK**.

   The log backup task is displayed in the task list. When **Task Status** in the task list changes to **Succeeded**, the backup is successful.

Checking Log Backup Files

Logs are backed up incrementally. After the backup is successful, you can access the target OBS bucket to obtain the full log files by clicking **Log Path**.

Figure 2-66 OBS bucket address

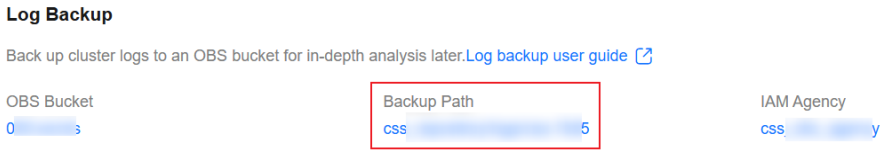


Table 2-197 lists the log types, where **clustername** indicates the cluster name.

Table 2-197 Log types

Log Name	Description
*clustername*_deprecation.log	Deprecation log file
*clustername*_index_indexing_slowlog.log	Slow indexing log file
*clustername*_index_search_slowlog.log	Slow query log file
clustername.log	Run log file
*clustername*_access.log	Access log file

Disabling Log Backup

Disable log backup if it is no longer needed.

On the **Log Backup** page, click **Disable Backup**. In the displayed dialog box, click **OK**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

> **CAUTION**
>
> Disabling log backup does not automatically delete existing backups. Instead, you need to manually delete them on the OBS console.

2.13.4 Viewing Audit Logs

To ensure data security and compliance, enterprises must routinely record and audit operations performed on their cloud service resources. With Cloud Trace Service (CTS), you can record operations associated with CSS Elasticsearch clusters for later query, auditing, and backtracking.

Prerequisites

CTS has been enabled.

Key Operations That Can Be Audited Using CTS

Table 2-198 Key operations that can be audited using CTS

Operation	Resource Type	Event Name
Querying product dynamics on the Overview page	cluster	listByGraph
Querying cluster and disk information on the Dashboards page	cluster	showDashboard
Querying supported engine types	cluster	listDatastores
Querying flavors supported by a specified engine type	cluster	getSingleDatabaseFlavors
Querying supported AZs	user	getAZInfo
Querying available disk types	cluster	listDiskTypes
Obtaining the node flavor list	cluster	listFlavors
Querying details about a specified flavor	cluster	showFlavorDetail
Querying feature availability	user	showFunctionSwitch
Querying available resource quotas	user	queryQuota
Creating a cluster	cluster	createCluster
Querying the cluster list	cluster	listClusters
Querying cluster details	cluster	showClusterDetail

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Operation	Resource Type	Event Name
Querying the storage capacity of a cluster	cluster	getClusterDiskInfos
Updating resource status	cluster	updateServiceStatus
Updating resource metadata	cluster	updateServiceMetadata
Changing the cluster name	cluster	updateClusterName
Modifying the cluster description	cluster	updateClusterDesc
Deleting a failed snapshot task	cluster	deleteFailedTaskByTaskId
Deleting all failed snapshot tasks	cluster	deleteAllFailedTask
Querying all failed snapshot tasks	cluster	listAllFailedTasks
Clearing the status of a specified task for a cluster	cluster	clearClusterAction
Quickly restarting a cluster	cluster	restartCluster
Performing a rolling restart on a cluster	cluster	rollingRestart
Deleting a cluster	cluster	deleteCluster
Creating a cluster (V2)	cluster	createClusterV2
Querying the cluster list	cluster	listClustersDetails
Querying the cluster list (V2)	cluster	listClustersDetailsV2
Changing the security group	cluster	changeSecurityGroup
Changing the password of a cluster	cluster	resetPassword
Downloading a security certificate	cluster	downloadCert
Adding data nodes to a cluster	cluster	updateExtendCluster
Adding more nodes and storage capacity	cluster	updateExtendInstanceStorage
Changing the specifications of data nodes in a cluster	cluster	updateFlavor
Changing the flavor of a specified node type	cluster	updateFlavorByType

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Operation	Resource Type	Event Name
Querying available flavors	cluster	queryResizeFlavors
Removing specified nodes	cluster	updateShrinkNodes
Removing nodes of a specific type	cluster	updateShrinkCluster
Querying information about nodes to be deleted	cluster	queryNeedDeleteInstances
Changing the security mode	cluster	changeMode
Querying the system indexes of a security-mode cluster	cluster	getSecurityIndex
Adding new node types	cluster	addIndependentNode
Enabling master nodes	cluster	addIndependentMasterNode
Enabling client nodes	cluster	addIndependentClientNode
Enabling cold data nodes	cluster	addIndependentColdNode
Replacing a specified node	cluster	updateInstance
Changing AZs for cluster nodes	cluster	updateAzByInstanceType
Upgrading the cluster kernel	cluster	upgradeCore
Obtaining information about the target image to upgrade to	cluster	listImages
Obtaining upgrade details	cluster	upgradeDetail
Retrying a failed kernel upgrade or AZ change task	cluster	retryAction
Terminating a failed kernel upgrade or AZ change task	cluster	abortAction
Querying all tags	cluster	listClustersTags
Querying the tags of a specified cluster	cluster	showClusterTag
Adding tags for a cluster	cluster	createClustersTags
Deleting the tags of a cluster	cluster	deleteClustersTags
Adding or deleting cluster tags in batches	cluster	updateBatchClustersTags
Enabling logging	cluster	startLogs

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Operation	Resource Type	Event Name
Disabling logging	cluster	stopLogs
Querying the log backup task list	cluster	listLogsJob
Querying basic log configurations	cluster	showGetLogSetting
Modifying basic log configurations	cluster	updateLogSetting
Enabling automatic log backup	cluster	startLogAutoBackupPolicy
Disabling automatic log backup	cluster	stopLogAutoBackupPolicy
Manually backing up logs	cluster	createLogBackup
Configuring automatic log backup	cluster	autoCreateLogBackup
Querying logs	cluster	showLogBackup
Checking the network connectivity between the current cluster and the destination cluster	cluster	detectClusterConnectivity
Automatically configuring basic cluster snapshot settings	cluster	startAutoSetting
Configuring basic cluster snapshot settings	cluster	openSnapshotFunction
Modifying the basic settings of a cluster snapshot	cluster	updateSnapshotSetting
Manually creating a snapshot	snapshot	createSnapshot
Configuring automatic snapshot creation	snapshot	autoCreateSnapshot
Restoring a snapshot	snapshot	restoreSnapshot
Deleting a specified snapshot	snapshot	deleteSnapshot
Setting an automatic snapshot creation policy	cluster	createAutoCreatePolicy
Querying cluster snapshot policies	cluster	queryClustersBackupPolicy
Querying automatic snapshot creation policies	cluster	showAutoCreatePolicy

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Operation	Resource Type	Event Name
Querying the cluster snapshot list	cluster	listSnapshots
Querying the cluster snapshot list	cluster	listClusterBackups
Disabling snapshots	cluster	stopSnapshot
Enabling automatic snapshot creation	cluster	startAutoCreateSnapshots
Disabling automatic snapshot creation	cluster	stopAutoCreateSnapshots
Querying the destination cluster for snapshot restoration	cluster	queryTargetRestoreCluster
Enabling or disabling VPC Endpoint	cluster	bindOrUnbindEIP
Enabling VPC Endpoint	cluster	startVpecp
Disabling VPC Endpoint	cluster	stopVpecp
Obtaining a VPCEP connection	cluster	showVpcepConnection
Updating a VPCEP connection	cluster	updateVpcepConnection
Modifying the VPCEP whitelist	cluster	updateVpcepWhitelist
Obtaining the parameter settings list	cluster	listYmls
Obtaining the parameter settings task list	cluster	listYmlsJob
Modifying parameter settings	cluster	updateYmls
Querying dedicated load balancers supported by a cluster	cluster	listElbs
Enabling or disabling a cluster load balancer	cluster	enableOrDisableElb
Configuring a cluster load balancer listener	cluster	createElbListener
Updating load balancer listeners for a cluster	cluster	updateEsListener
Obtaining information about a cluster's load balancers	cluster	showElbDetail

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Operation	Resource Type	Event Name
Querying certificates supported by a load balancer	cluster	listElbCerts
Querying historical reports and details for intelligent diagnostics	cluster	listAiOps
Manually creating an intelligent diagnostics task	cluster	createAiOps
Automatically creating an intelligent diagnostics task	cluster	autoCreateAiOps
Deleting an intelligent diagnostics task	cluster	deleteAiOps
Obtaining SMN topics available for intelligent O&M alarms	cluster	listSmnTopics
Updating the scheduled diagnostics configuration for intelligent O&M	cluster	updateAiOpsSetting
Disabling scheduled diagnostics for intelligent O&M	cluster	closeAiOpsSetting
Viewing the scheduled diagnostics configuration for intelligent O&M	cluster	showAiOpsSetting
Viewing intelligent O&M diagnostics items	cluster	showAiOpsDetector
Querying default plug-ins	cluster	getDefaultPlugins
Creating an agency	user	createAgency
Querying the agency list	cluster	listAgencies
Querying available OBS buckets	user	getAvailableBuckets
Querying files in a specified OBS bucket	cluster	listObjects
Querying resources by tag	cluster	getResourceByTags

Querying Real-Time Traces

After a management tracker is created on the CTS console, the system starts recording operations performed on cloud service resources. After a data tracker is

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

created, the system starts recording operations performed on data in OBS buckets. CTS retains operation records generated in the latest seven days.

2.14 Managing Alarms

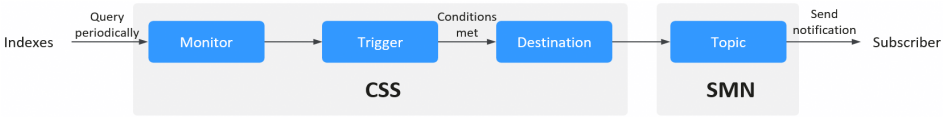
2.14.1 Configuring Open Distro Alert Notifications (via SMN)

When monitoring big data platforms in real time, O&M teams need to be promptly notified of cluster health status or any abnormal changes in service metrics. Although Elasticsearch provides powerful search capabilities, the native engine lacks deep integration with cloud-native notification services. CSS addresses this by offering a built-in Open Distro alerting plugin that integrates seamlessly with the Simple Message Notification (SMN) service. This integration enables a highly automated monitoring workflow: monitors are configured to query specified indexes periodically; triggers evaluate the query results against predefined thresholds to identify anomalies and generate alerts; CSS then calls the SMN API to push alerts to subscribers.

How the Feature Works

CSS Elasticsearch clusters provide the built-in open-source Open Distro alerting plugin, which triggers alerts and sends notifications via SMN when predefined conditions are met.

Figure 2-67 Alert notification link



1. A monitor defines the query conditions, such as which index to query and how often.
2. A trigger checks the monitor's query results to determine whether preset conditions that trigger an alert are met (for example, **count > 100**). If yes, it generates an alert and triggers a predefined action.
3. A destination is a message channel that defines the SMN topic where alerts are to be sent.
4. A CSS cluster is granted permission to access SMN through an agency.
5. SMN sends the received alert to subscribers.

For details about the Open Distro alerting plugin, see the official document **Open Distro Alerting**.

Constraints

Only Elasticsearch 7.6.2 and 7.10.2 clusters provide a built-in Open Distro alerting plugin. Only clusters of these versions support alert notification via SMN.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Preparations

1. Prepare the needed SMN topic. Create a topic on SMN and add subscribers (mobile number or email address). For details, see **Publishing a Template Message**.
2. Grant SMN access to CSS. You can configure an IAM agency to authorize CSS to access SMN.
   a. Use a CSS administrator account to perform: Log in to the **CSS management console**.

      The CSS administrator account must have the following minimum IAM permissions:

      ```
      "iam:agencies:listAgencies",
      "iam:roles:listRoles",
      "iam:agencies:getAgency",
      "iam:agencies:createAgency",
      "iam:permissions:listRolesForAgency",
      "iam:permissions:grantRoleToAgency",
      "iam:permissions:listRolesForAgencyOnProject",
      "iam:permissions:revokeRoleFromAgency",
      "iam:roles:createRole"
      ```

   b. In the navigation pane, choose **Service Authorization**.
   c. On the **Service Authorization** page, click **Create Agency** for SMN. In the dialog box displayed, confirm that the agency is successfully created.
      - If an agency has already been created, the message "The css_smn_agency already exists. Please check the agency permission." is displayed in the upper right corner, indicating SMN access has been granted to CSS.
      - If you do not have the permission to create an agency, an error message is displayed in the upper right corner indicating "no permission", in which case, check that the administrator account has been assigned the necessary IAM permissions.

Configuring Alert Notifications via SMN

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

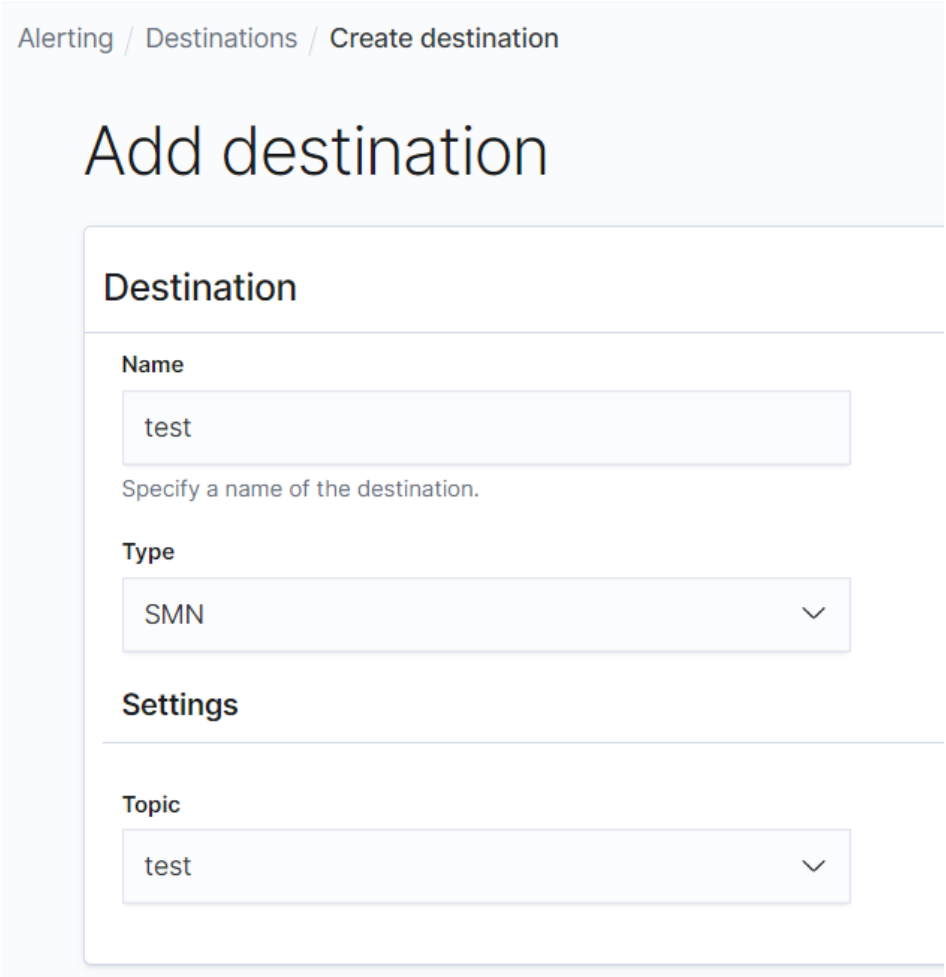
   Log in using an account with CSS permissions.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
4. On the Kibana page, choose **Open Distro for Elasticsearch** > **Alerting** in the navigation pane on the left.
5. Create an SMN destination for sending alert messages.
   a. On the **Alerting** page, click the **Destinations** tab and click **Add destination** to configure destination information.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-199 Destinations parameter description

Parameter	Description
Name	User-defined destination name. You are advised to use English letters and digits.
Type	Type of the notification. Select SMN.
Topic	An associated SMN topic. Select the SMN topic created in **Preparations**.

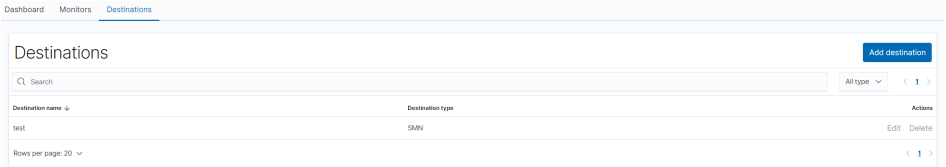
Figure 2-68 Add destination



b. Click **Create**.

c. Return to the **Destinations** page. If the new destination is displayed, it has been created.

Figure 2-69 Destination list



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

6. Create a monitor and trigger to define the alarm triggering conditions and monitor interval.
   a. Click the **Monitors** tab on the **Alerting** page, and click **Create monitor** to configure the monitor.

      Table 2-200 Monitor parameters

Parameter		Description
Configure monitor	Monitor name	User-defined monitor name
	Monitor state	Whether to disable the monitor. • Select **Disable monitor**: Disable the monitor. • (Recommended) Deselect **Disable monitor**: Enable the monitor.
Define monitor	Method of definition	Select a method to define the monitor. You are advised to use **Define using extraction query**. • **Define using visual graph**: use a visual query • **Define using extraction query**: use a specific query
	Index	Index to be monitored
	Time field	Timestamp used for time-based aggregations, such as **count**. This parameter is required only when you select **Define using visual graph** to define the monitor.
Monitor schedule	Frequency	Select the monitor frequency and set the monitor interval. The options include: • **By interval** • **Daily** • **Weekly** • **Monthly** • **Custom cron expression**

   b. Click **Create**. The **Create trigger** page is displayed.
   c. On the **Create trigger** page, set the alert triggering conditions and the actions to be triggered.

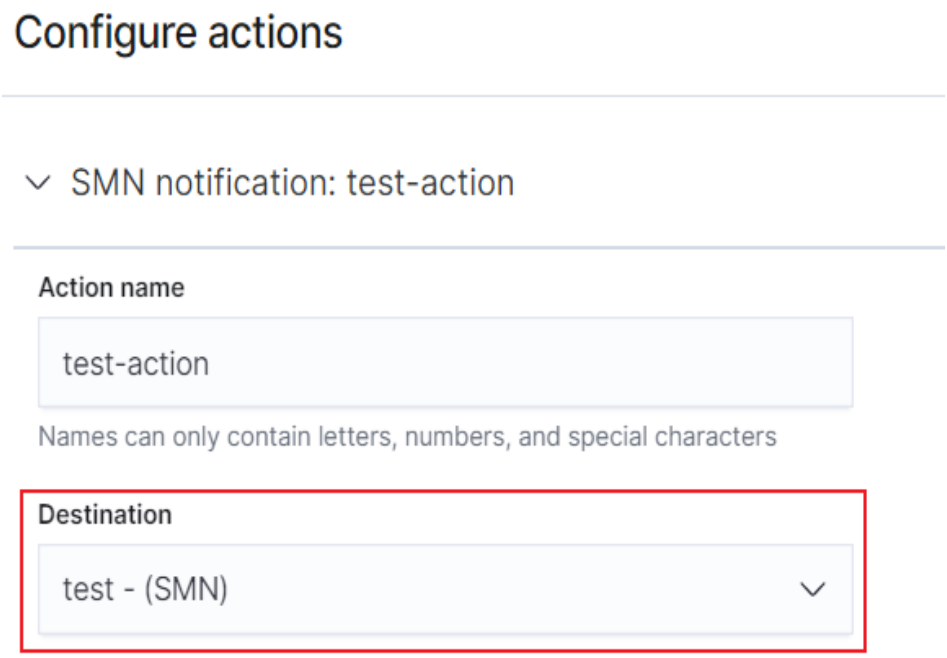
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-201 Trigger parameters

Parameter		Description
Define trigger	Trigger name	User-defined trigger name
	Severity level	Sensitivity of a trigger, that is, the number of alerts that need to be triggered before a notification is sent. **1** indicates the highest sensitivity.
	Trigger condition	Trigger condition. An alert is triggered when the trigger condition is hit.
Configure actions	Action name	Name of a trigger action
	Destination	Select the SMN destination created in **5**.
	Message subject	Message title. This parameter is required only for Elasticsearch 7.10.2 clusters.
	Message	Message body. By default, the subject and body are defined when the destination is an email address.
	Action throttling	Specify the message frequency to limit the number of notifications you receive within a given span of time. Without it, high-frequency or low-severity triggers may cause information overload or result in unexpected cloud costs. For example, if this parameter is set to 10 minutes, SMN sends only one alert notification in the next 10 minutes even if the trigger condition is hit multiple times. After 10 minutes, SMN sends another notification if the trigger condition is met again.

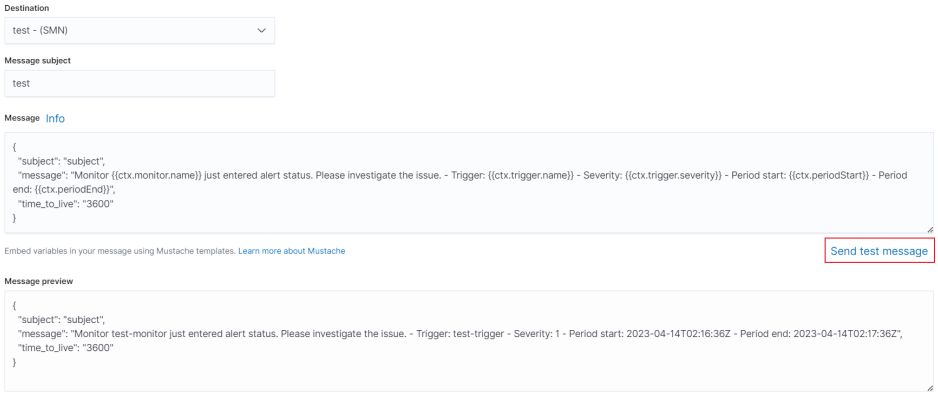
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 2-70 Setting the destination of a trigger action



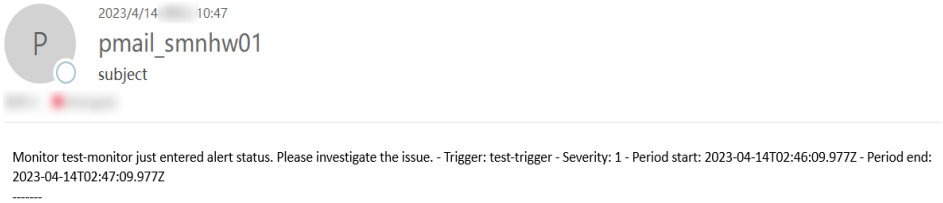
d. Click **Send test message** to test the alert notification link.

Figure 2-71 Sending a test message



e. As shown in **Figure 2-72**, if a subscriber to the SMN topic receives the message, alert notification is configured successfully.

Figure 2-72 Email notification



f. Click **Create** to go to the monitor details page.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

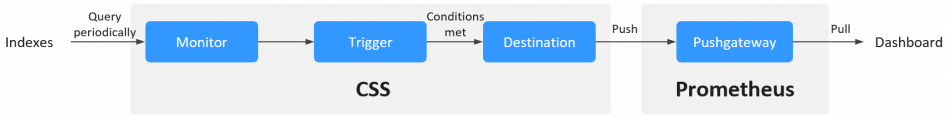
2.14.2 Synchronizing Alerts and Metrics to Prometheus

When building a full-stack monitoring system, O&M teams often use Prometheus to aggregate infrastructure and application metrics. However, business-level data stored in Elasticsearch (such as user access patterns and transaction logs) remains in document form and is not natively available as Prometheus-format time series, hindering unified analysis and alerting in Grafana dashboards. How can we expose Elasticsearch aggregation results as Prometheus-compatible time series? CSS Elasticsearch clusters address this gap using the built-in Open Distro Alerting plugin, which provides a Prometheus-compatible export channel. Using the Prometheus Pushgateway, this plugin periodically executes queries, transforms the results into structured metrics, and pushes them to the Pushgateway where they can be scraped by Prometheus. This approach enables Elasticsearch clusters to act as intelligent metric sources, supporting unified analysis of business and infrastructure data and consistent, threshold-based alerting within a single observability stack.

How the Feature Works

Elasticsearch business metrics must be pushed by the alerting plugin, whereas Prometheus uses a pull-based model by default. The Prometheus Pushgateway bridges this gap: it receives push-based metrics and exposes them via a scrape endpoint for Prometheus. The Open Distro Alerting plugin built into CSS Elasticsearch clusters provides a Prometheus-compatible export channel connected to the Pushgateway.

Figure 2-73 Alert & metric synchronization link



1. **Monitor**: Defines the aggregation query (which metrics to compute), the target indexes, and the execution schedule (query frequency).
2. **Trigger**: Evaluates query results against threshold conditions to determine if and when an alert should be generated. It also triggers an action. **Action**: Transforms Elasticsearch aggregation results (in document form) into standard Prometheus JSON format.
3. **Destination**: A message channel that sends Prometheus-compatible JSON packets to the Pushgateway.
4. **Pushgateway**: A temporary storage area that receives push-based metrics from Elasticsearch.
5. **Prometheus**: Periodically pulls data from the Pushgateway and stores it in its time-series database, where Grafana queries the data and visualizes metrics on unified dashboards.

For more about the Open Distro alerting plugin, see **Open Distro Alerting**. For more about Prometheus Pushgateway, see **Prometheus Pushgateway**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Constraints

- This feature is available only for Elasticsearch clusters whose image version is no earlier than 7.10.2_24.3.3_xxx.
- The Elasticsearch cluster must be able to access the Prometheus Pushgateway endpoint. Prometheus and Elasticsearch must be connected. Otherwise, alerts cannot be sent.
- Only Gauge-type metrics (used in dashboards) can be synchronized. They include statistical or percentage values that can both increase and decrease.

Configuring Alert Synchronization

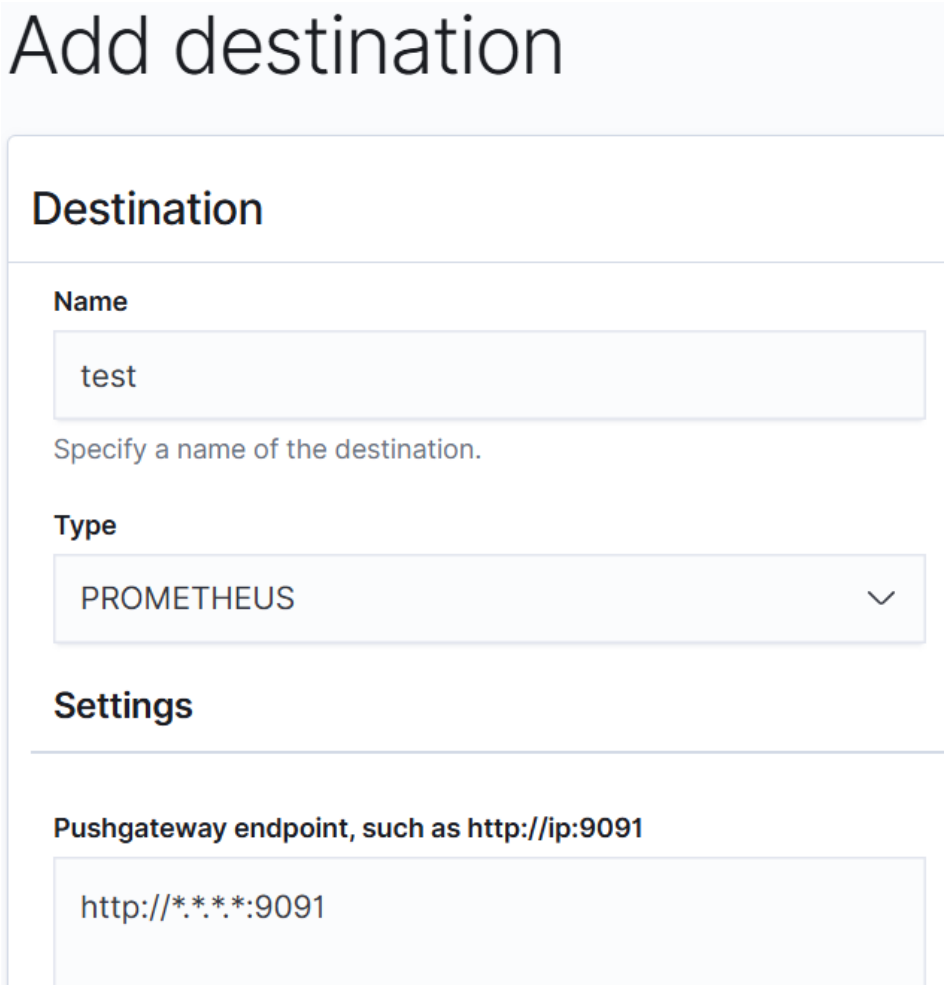
1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
4. On the Kibana page, choose **Open Distro for Elasticsearch** > **Alerting** in the navigation pane on the left.
5. Create a Prometheus destination to establish a physical connection between Elasticsearch and Pushgateway.
   a. On the **Alerting** page, click the **Destinations** tab, and click **Add destination** to configure destination information.

Table 2-202 Destinations parameters

Parameter	Description
Name	User-defined destination name
Type	Type of the notification. Select **PROMETHEUS**.
Settings	Enter the Pushgateway address of the Prometheus monitor server. • Currently, dashboards can only visualize Gauge-type metrics. These include statistical and percentage values that can both increase and decrease. Metrics are queried using specific statements and numeric values are synchronized to Pushgateway for monitoring via Prometheus. • Two types of Pushgateway addresses are supported: HTTP and HTTPS.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

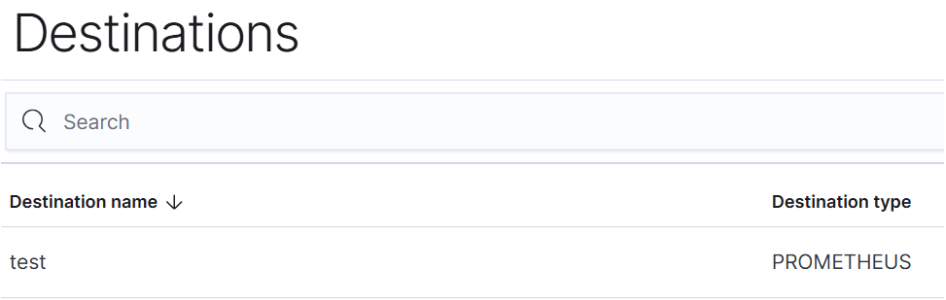
Figure 2-74 Add destination



b. Click **Create**.

c. Return to the **Destinations** page. If the new destination is displayed, it has been created.

Figure 2-75 Destinations list



6. Create a monitor and trigger to define the alarm triggering conditions and monitor interval.

   a. On the **Alerting** page, click the **Monitors** tab. Then click **Create monitor**, and configure the data source and query frequency.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-203 Monitor parameters

Parameter		Description
Configure monitor	Monitor name	User-defined monitor name
	Monitor state	Whether to disable the monitor. • Select **Disable monitor**: Disable the monitor. • (Recommended) Deselect **Disable monitor**: Enable the monitor.
Define monitor	Method of definition	Select a method to define the monitor. You are advised to use **Define using extraction query**. • **Define using visual graph**: use a visual query • **Define using extraction query**: use a specific query
	Index	Index to be monitored
	Time field	Timestamp used for time-based aggregations, such as **count**. This parameter is required only when you select **Define using visual graph** to define the monitor.
Monitor schedule	Frequency	Select the monitor frequency and set the monitor interval. The options include: • **By interval** • **Daily** • **Weekly** • **Monthly** • **Custom cron expression**

b. Click **Create**. The **Create trigger** page is displayed.

c. On the **Create trigger** page, define how to convert data into Prometheus-readable formats.

Table 2-204 Trigger parameters

Parameter		Description
Define trigger	Trigger name	User-defined trigger name.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter		Description
	Severity level	Sensitivity of the trigger, that is, how many alerts need to be triggered before an alert is actually sent. **1** indicates the highest sensitivity.
	Trigger condition	Trigger condition. An alert is triggered when the trigger condition is met. You are advised to set a trigger condition that can almost always be triggered (for example, **1 > 0**) so that the query results will always be synchronized to the Pushgateway.
Configure actions	Action name	Name of the triggered action.
	Destination	Select the Prometheus destination created in 5.

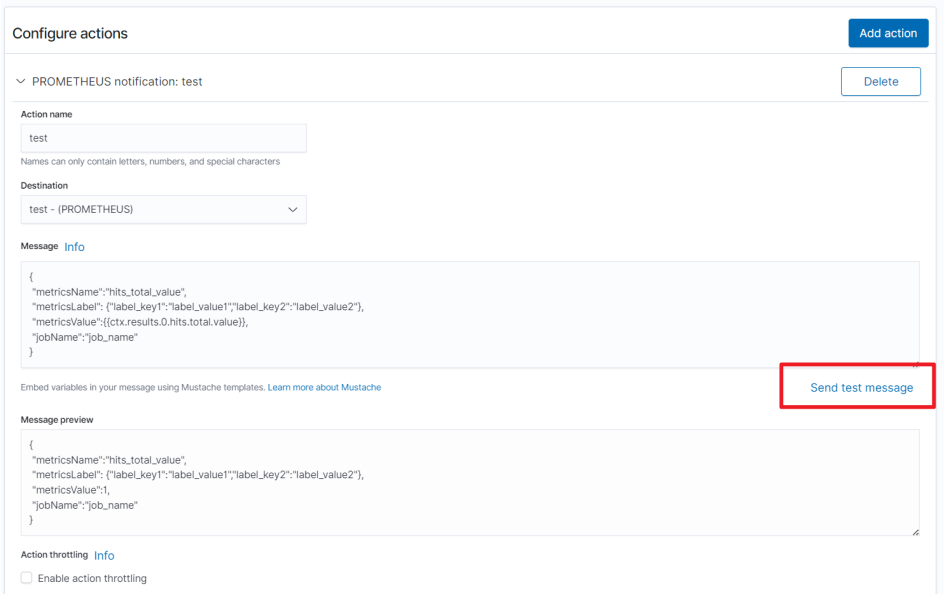
	Message	Defines the body of the message to be published, which must use the JSON format.
		Message template:
		{
		"metricsName":"hits_total_value", //Prometheus metric name
		"metricsLabel": {"label_key1":"label_value1","label_key2":"label_value2"}, //Prometheus labels
		"metricsValue":{{ctx.results.0.hits.total.value}}, //Prometheus metric values
		"jobName":"job_name", //Prometheus monitor task name
		"metricsHelp":"***" //Metric explanation. Optional.
		}

	Action throttling	Specify the message frequency to limit the number of notifications you receive within a given span of time. Without it, high-frequency or low-severity triggers may cause information overload or result in unexpected cloud costs. For example, if this parameter is set to 10 minutes, Prometheus sends only one alert notification in the next 10 minutes even if the trigger condition is met multiple times. After 10 minutes, Prometheus sends another alert notification if the trigger condition is met again.

d. Click **Send test message** to test the link to Prometheus.

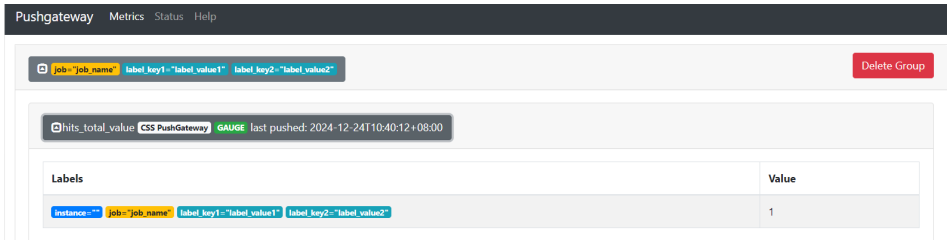
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 2-76 Sending a test message



e. As shown in **Figure 2-77**, Prometheus can receive a triggered message, meaning the trigger is set successfully.

Figure 2-77 Message received successfully



f. Click **Create** to go to the monitor details page.

2.15 Intelligent Diagnostics

During routine operations, you must periodically monitor the health of your Elasticsearch clusters to ensure service stability. However, manual health checks are time-consuming, labor-consuming, and prone to oversight. Key issues may be easily missed. To address this, CSS provides an Intelligent O&M feature. It supports both scheduled and on-demand diagnostics, covering essential checks such as data node disk usage, cluster health status, and node CPU usage. Through these checks, O&M personnel can detect and handle potential risks in a timely manner to ensure cluster reliability and availability. By default, Intelligent O&M retains the most recent diagnostic reports, so you can always review the latest results. You can also export any report for further analysis or long-term archiving.

Constraints

- Intelligent diagnostics cannot be enabled while the cluster is in one of the following **Cluster Status**: frozen, creating, or creation failed.
- While a diagnostic task is in progress, you cannot manually start another task.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- Running diagnostics on a cluster that is in the process of a configuration change (for example, being scaled up or down, or having its node specifications changed) may lead to inaccurate results. You are advised to avoid such times.
- When intelligent diagnostics is enabled, the system is allowed by default to read cluster settings and data configuration information for diagnostic or analytical purposes.

Diagnostic Modes

Two diagnostic modes are available: scheduled and manual. Choose one based on your service requirements.

Table 2-205 Comparing the two diagnostic modes

Dimension	Scheduled Diagnostics	Manual Diagnostics
Triggering method	Automatically performed daily by schedule	Manually triggered and executed right away
When to use	Routine inspection, creating a health baseline, and long-term trend observation	Quick troubleshooting in response to cluster failures or errors, or cluster status verification before and after change operations
Typical users	O&M teams (routine maintenance)	R&D or O&M personnel (troubleshooting)
Diagnostic type	• Full check • Cluster unavailability check	• Full check • Cluster unavailability check

Configuring Scheduled Diagnostics

Configuring daily scheduled diagnostics enables continuous health risk monitoring for clusters without human intervention.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Intelligent O&M > Intelligent Diagnostics**.
5. On the **Intelligent Diagnostics** page, click **Enable Scheduled Diagnostics**. In the displayed dialog box, configure a scheduled diagnostics task.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-206 Enabling scheduled diagnostics

Parameter	Description
Diagnostic Type	Select a diagnostic type. • **Full check**: All check items are supported. You can select all or only some of the check items. • **Cluster unavailability check**: When a cluster is unavailable, run this diagnosis to quickly diagnose its health status.
Diagnostic Item	Select diagnostic items to be included in the scheduled task. • When **Diagnostic Type** is set to **Full check**, you can customize the diagnostic items. • When **Diagnostic Type** is set to **Cluster unavailability check**, the cluster unavailability check items are selected by default.
Time Zone	Select your current time zone. The start time of the scheduled diagnostic task is affected by it.
Diagnosed	Select the time to start the scheduled diagnostic task every day.

6. Click **OK** to save the task information.

   The scheduled task information is displayed on the **Intelligent Diagnostics** page, and daily reports generated by the task will be displayed below on this page or on the **Historical Reports** tab.

   Figure 2-78 Viewing a scheduled diagnostics task

   Reports will show high-risk items, medium-risk items, normal items, and reminders, as well as the results and optimization suggestions for each diagnostic item. High- and medium-risk items should be prioritized.

7. On the **Intelligent Diagnostics** page, manage the scheduled diagnostic task, including modifying and disabling it.
   - Modify the scheduled diagnostic task: Click **Modify Settings**. In the displayed dialog box, modify the settings and click **OK** to update the scheduled diagnostic policy.
   - Disable the scheduled diagnostic task: Click **Disable Scheduled Diagnostics**. In the displayed dialog box, click **OK** to disable scheduled diagnostics. The scheduled task is removed from the Intelligent Diagnostics page.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Performing Manual Diagnostics

Perform a one-time manual diagnostic for quick troubleshooting or to verify cluster status. This is recommended when your cluster experiences performance issues or errors, or after changes are made to your cluster.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Intelligent O&M > Intelligent Diagnostics**.
5. On the **Intelligent Diagnostics** page, configure a diagnostic task and click **Run Diagnostics** to start the task.

Figure 2-79 Run Diagnostics

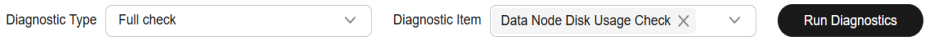


Table 2-207 Manual diagnostics

Parameter	Description
Diagnostic Type	Select a diagnostic type. • **Full check**: All check items are supported. You can select all or only some of the check items. • **Cluster unavailability check**: When a cluster is unavailable, run this diagnosis to quickly diagnose its health status.
Diagnostic Item	Select diagnostic items. • When **Diagnostic Type** is set to **Full check**, you can customize the diagnostic items. • When **Diagnostic Type** is set to **Cluster unavailability check**, the cluster unavailability check items are selected by default.

6. Wait for 1 to 3 minutes. The diagnostic report will be displayed below.

   The report will show high-risk items, medium-risk items, normal items, and reminders, as well as the results and optimization suggestions for each diagnostic item. High- and medium-risk items should be prioritized.
7. Click **Export Report** to save the diagnostic report to your local PC for further analysis or archiving.

Viewing Historical Reports

Historical diagnostic reports help you compare cluster health across different periods, which aids capacity planning and problem diagnosis.

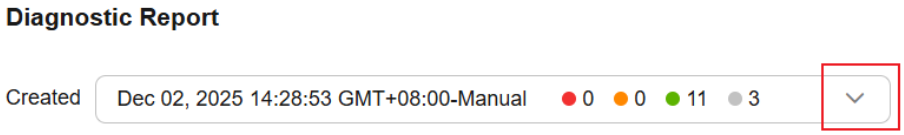
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

CAUTION

A maximum of 10 historical reports can be retained. (This is the total for both scheduled and manual diagnostic tasks.) When this limit is exceeded, the oldest reports will be deleted automatically. Export and save important reports in a timely manner.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Intelligent O&M > Historical Reports**.
5. Set **Created** to filter diagnostic reports by time range.

   Figure 2-80 Selecting historical reports

6. Click **Export Report** on the right to save the current historical report to the local PC.

Diagnostic Items

Table 2-208 lists the diagnostic items supported by Intelligent O&M.

Table 2-208 Diagnostic items

No.	Diagnostic Item	Description
1	Data Node Disk Usage Check	Identify data nodes (including cold data nodes) with the highest disk usage in the cluster. Excessive disk usage can affect cluster stability. • High risk: When a node's disk usage reaches 85%, no new index replicas will be allocated to it. • Medium risk: When a node's disk usage exceeds 80%, take immediate action to prevent further increase.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

No.	Diagnostic Item	Description
2	Data Node Disk Usage Balance Check	Check whether disk usage is balanced across all regular data nodes and cold data nodes. Unbalanced disk usage may cause performance issues. ● High risk: Maximum disk usage difference between nodes ≥ 40% ● Medium risk: 25% ≤ Maximum disk usage difference between nodes < 40%
3	Large Index Shard Check	Check for large index shards that may cause cluster load imbalance. ● High risk: Shard storage ≥ 50 GB; and the index only has one primary shard. ● Medium risk: Shard storage ≥ 50 GB; and the number of shards is not divisible by the number of data nodes or cold data nodes.
4	Cluster Node Configuration Check	Check the high availability settings of clusters. ● Single/Dual Node Risk: For clusters without a dedicated Master node, at least three Data nodes (including cold data nodes) are required to ensure cluster availability during a single node failure. ● Overloaded Master Node Risk: When there are 10 or more Data nodes, dedicated Master nodes should be configured to prevent excessive load on the Master node. ● Single Client Node Risk: A cluster containing only one dedicated Client node may result in cluster inaccessibility if that single Client node fails.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

No.	Diagnostic Item	Description
5	Cluster Health Check	Check the overall health status of all indices in the cluster. ● High risk (red): There are unallocated primary shards (i.e., some indices are unavailable). ● Medium risk (yellow): Primary shards are properly allocated, but replicas are missing (compromised data availability). ● Normal (green): All primary and replica shards are properly allocated.
6	Maximum Per-Node Shard Quantity Check	Check that the number of shards allocated to each node remains within the limit set by cluster.max_shards_per_node, preventing shard allocation failures. ● High risk: Maximum number of shards allocated to a node ≥ 90% of the per-node shard limit ● Medium risk: 85% of the per-node shard limit ≤ Maximum number of shards allocated to a node < 90% of the per-node shard limit
7	Oversized Shard Check	Check for shards that have reached or exceeded 100 GB in the cluster. Oversized shards can increase query latency and slow down fault recovery. You are advised to keep the shard size under 50 GB. ● High risk: Maximum shard size ≥ 500 GB ● Medium risk: 100 GB ≤ Maximum shard size < 500 GB
8	Hot Index Read/Write Check (Every Minute)	Identify the 10 indices with the most query and write requests within the last minute and display them in descending order. For each of these identified indices, verify that the number of its primary shards is divisible by the number of data nodes or cold data nodes to prevent load imbalance.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

No.	Diagnostic Item	Description
9	Node Disconnection Check	Check the connectivity of cluster nodes. Node disconnection will severely compromise cluster performance and therefore needs to be handled immediately.
10	Balanced Node Connections Check	Check the number of connections to port 9200 on the client nodes (or data nodes if there are no client nodes), and display nodes by their connection count in descending order.
11	Client Connection Check (Source IP Address Analysis)	Check the number of connections to port 9200 on the client nodes (or data nodes if there are no client nodes), and display all external source IP addresses by connection count in descending order as well as details about the top 10 connections.
12	Node CPU Usage Check (Interval: 1 Minute)	Measure the CPU usage of each node twice every minute, and calculate the average for each node to evaluate their load status. ● High risk: Average CPU usage ≥ 90% or no data is obtained. ● Medium risk: 80% ≤ Average CPU usage < 90%
13	JVM Heap Memory Usage Check (Interval: 1 Minute)	Measure the JVM heap memory usage of each node twice every minute, and calculate the average for each node to evaluate their load status. ● High risk: Average JVM heap memory usage ≥ 90% or no data is obtained. ● Medium risk: 80% ≤ Average JVM heap memory usage < 90%
14	Local-Disk Cluster Index Replica Check	For a cluster consisting of nodes that use local disks, check whether all indices have replicas. Otherwise, a single disk failure can lead to permanent data loss.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2.16 Managing Cluster Changes

2.16.1 Scaling Out/Up

If an Elasticsearch cluster struggles to maintain performance in the face of rapid data growth or sustained high memory usage, you may scale it horizontally by adding more nodes and more node types, or do so vertically by increasing the storage capacity of existing nodes.

Table 2-209 Scaling scenarios

Type	Scenario	Change Process
Adding new nodes	If a cluster experiences rapid data growth or sustained high memory usage, you may add more nodes to it to improve cluster availability.	Procedure for adding data nodes or cold data nodes: 1. Add new nodes and modify cluster configuration. 2. A shard redistribution is automatically triggered to reallocate certain shards to newly added nodes. 3. After the redistribution is complete, the new nodes start to handle query and indexing requests. Adding master or client nodes does not trigger data migration. During the capacity expansion, the system ensures that each shard has at least one available replica to ensure service continuity.
Adding master nodes	As your cluster's data volume grows, you can improve its stability and reliability by adding dedicated master nodes for more effective cluster node management.	Procedure for adding a new node type: Add new nodes and modify cluster configuration. Adding master nodes does not trigger data migration, and therefore does not interrupt services.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Type	Scenario	Change Process
Adding client nodes	To handle increased query and write loads as your cluster's data volume grows, you can add dedicated client nodes. This offloads request processing from your data nodes, improving overall cluster scalability.	Procedure for adding a new node type: Add new nodes and modify cluster configuration. Adding client nodes does not trigger data migration, and therefore does not interrupt services. However, when client nodes are added, the cluster's address changes from the data node address to the client node address. You must update the client configuration to use this new address. Otherwise, the client nodes cannot function.
Adding cold data nodes	As your cluster's data volume keeps growing, you may have large quantities of historical data that is infrequently accessed but still useful for analytical purposes. To optimize both storage costs and query performance, you can add cold data nodes to store such data.	Procedure for adding a new node type: Add new nodes and modify cluster configuration. Adding cold data nodes does not trigger data migration, and therefore does not interrupt services.
Increasing node storage capacity	• If a cluster experiences rapid data growth and its storage capacity becomes insufficient, you can expand the cluster nodes' storage capacity. • If the disk usage of a cluster remains high, you can expand the cluster nodes' storage capacity.	1. A request is sent to the EVS service to expand the disk capacities for all cluster nodes. 2. After disk capacity is expanded by EVS, the size of existing disk volumes is expanded on all cluster nodes. The disk capacity expansion will not interrupt ongoing services.

Impact on Billing

For a pay-per-use cluster, you can see its new price when confirming the scale-out or scale-up on the console. After the change is complete, the cluster will be billed based on the new price. For details, see **Cloud Search Service Price Calculator**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Constraints

- The node storage capacity of a cluster can only be increased—not decreased. Select an appropriate node storage capacity based on the data volume and projected data growth.
- You cannot increase the number of nodes and node storage capacity for a cluster at the same time.
- The storage capacity of the master and client nodes in a cluster cannot be expanded.
- The storage capacity of data nodes that use local disks cannot be expanded.
- For a multi-AZ cluster, the total number of client nodes must be greater than or equal to the number of AZs. (For a cluster spanning two AZs, this means a minimum of two client nodes; for a three-AZ deployment, a minimum of three client nodes.)
- Principles for adding cold data nodes to a multi-AZ cluster:
  - Node quantity requirement: A two-AZ cluster **must** have at least four cold data nodes (two in each AZ); and a three-AZ cluster **must** have at least three cold data nodes (one in each AZ).
  - Performance enhancement: To prevent uneven data distribution and ensure optimal query and ingestion performance, the number of cold data nodes should be an integer multiple of the number of AZs.
- For the range of node quantities supported by each node type, see **Table 2-210**.

Table 2-210 Node quantity ranges

Node Type	Value Range
Data nodes	• Without master nodes: 1 to 32 • With master nodes: 1 to 200 (If the combined number of data nodes and cold data nodes exceeds 100, you are advised to select a flavor with at least 16 vCPUs. This helps ensure optimal read and write performance.)
Master nodes	3, 5, 7, or 9 (must be an odd number from 3 to 9)
Client nodes	1~64
Cold data nodes	1–32

Change Impact

Before the change, learn about possible impacts and operation suggestions, and develop a plan to minimize these impacts.

Expanding the storage capacity of cluster nodes (vertical scaling) does not impact services, whereas adding new nodes or new node types (horizontal scaling) may have the following impacts:

- Impact on performance

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Adding new nodes does not interrupt services. However, after new nodes are added, data shards need to be redistributed to these nodes to balance the load, and this process will consume I/O performance. This is why you are advised to perform the operation during off-peak hours.

To minimize this impact, it is advisable to adjust the data migration rate based on the cluster's traffic cycle: increase the data migration rate during off-peak hours to shorten the task duration, and decrease it **before** peak hours arrive to ensure optimal cluster performance. The data migration rate is determined by the **indices.recovery.max_bytes_per_sec** parameter. The default value of this parameter is the number of vCPUs multiplied by 8 MB. For example, for four vCPUs, the data migration rate is 32 MB/s. You can adjust it based on the service requirements.

```
PUT /_cluster/settings
{
  "transient": {
    "indices.recovery.max_bytes_per_sec": "128MB"
  }
}
```

- Characteristics of this process

  Once started, a scaling task cannot be stopped until it succeeds or fails.

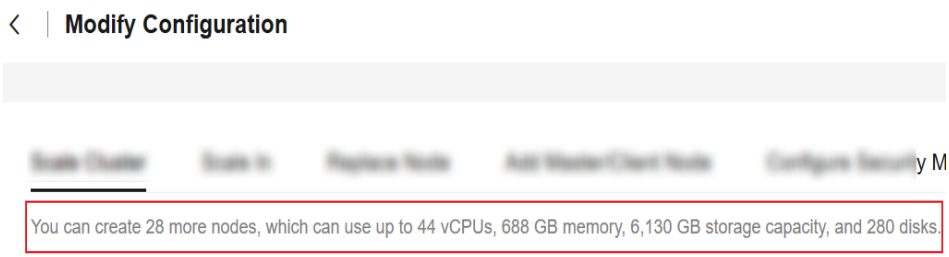
Duration

- It takes 10 to 30 minutes to add new nodes or new node types (horizontal scaling), depending on the cluster's scheduling capabilities.
- It takes 10 to 15 minutes to expand the storage capacity of cluster nodes (vertical scaling).

Prerequisites

- The cluster status is **Available**, and there are no ongoing tasks.
- Your CSS resource quotas are sufficient for the capacity expansion you are about to perform. You can check available resources on the **Modify Configuration** page.

  Figure 2-81 Checking available resources

Adding More Nodes or Increasing Node Storage Capacity

If a cluster experiences rapid data growth or sustained high memory usage, you may add more nodes to it or increase the cluster nodes' storage capacity. This helps to improve service and data availability and durability.

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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. Click the **Scale Cluster** tab.
5. Set the necessary parameters.

   Table 2-211 Adding more nodes or increasing node storage capacity

Parameter	Description
Action	Select **Scale out**.
Resources	Quantities of resources added.
Nodes	• Select the node type you want to scale in the **Node Type** column. Only one node type can be selected at a time. • In the **Nodes** column, increase the number of nodes. For the range of node quantities supported by each node type, see **Constraints**. • In the **Node Storage Type** column, increase the node storage capacity. The value range of node storage capacity is determined by the node flavor you select. The value must be divisible by 20. You cannot increase the number of nodes and node storage capacity for a cluster at the same time.

6. Click **Next**.
7. Confirm the information and click **Submit**.
8. Click **Back to Cluster List** to go back to the **Clusters** page. The **Task Status** is **Scaling out**. When **Cluster Status** changes to **Available**, the cluster has been successfully scaled out.

Adding New Node Types

For a cluster that has no dedicated master, client, or cold data nodes, as the load on its data plane increases, you may dynamically add such nodes to it to share the load of the data nodes.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. Click the **Add Node Types** tab.

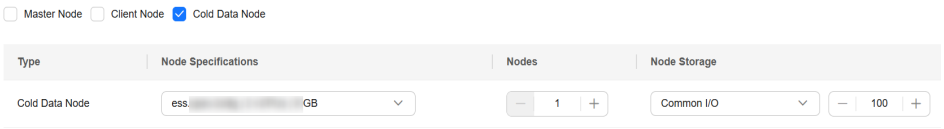
   If a cluster already has dedicated master, client, and cold data nodes, the **Add Node Types** tab is unavailable.
5. On the **Add Node Types** tab, configure the nodes.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-212 Adding new node types

Parameter	Description
Node types	Select the type of nodes you want to add. • Only the node types that the cluster does not already have are displayed. For example, if the cluster already has dedicated master nodes, only **Client Node** and **Cold Data Node** are displayed. • Only one node type can be added at a time. For example, if you need to add both master and client nodes, you need to perform this task twice.
Node Specifications	Select node specifications based on site requirements.
Nodes	Set the number of nodes to add. For the value range, see **Constraints**.
Node Storage Type	Set the node storage type. The storage capacity per master or client node is fixed to 40 GB.

Figure 2-82 Adding new node types



6. Click **Next**.
7. Confirm the information and click **Submit**.

   Return to the cluster list page. The **Task Status** of the cluster is **Scaling out**. When **Cluster Status** changes to **Available**, nodes have been added successfully.

   - When adding master nodes for a cluster whose version is earlier than 7.x, restart data nodes and cold data nodes after **Cluster Status** changes to **Available** to make the configuration of the new master nodes take effect. Otherwise, the cluster status may be displayed as unavailable due to delayed status synchronization (when in fact the cluster services are available).
   - When adding client nodes, you can choose whether to restart data nodes and cold data nodes and bring the Cerebro and Kibana processes offline after **Cluster Status** changes to **Available**. This helps avoid conflicts.
   - When adding cold data nodes, you must restart data nodes to activate the hot tag after **Cluster Status** changes to **Available**. This ensures that the cold data nodes function properly. After cold data nodes are added, manually migrate historical data to them. For details, see **Switching Between Hot and Cold Data**.

   For details about the restart operation, see **Restarting a Cluster**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Related Documents

- You can also upgrade the node specifications and change the EVS disk type of an Elasticsearch cluster. For details, see **Changing Node Specifications**.
- During a capacity expansion, if the resources in an AZ become insufficient, you can migrate the nodes in this AZ to another that has sufficient resources, and then perform the capacity expansion in the new AZ. For details, see **Switching AZs**.

2.16.2 Scaling In/Down

If an Elasticsearch cluster has excess capacity due to off-peak traffic or reduced data volumes, you can reduce its nodes to optimize costs.

Table 2-213 Scaling scenarios

Type	Scenario	Change Process
Removing nodes randomly	Randomly removes cluster nodes to optimize costs.	1. Migrate the shards on the to-be-removed nodes to the remaining nodes. 2. After data migration, bring the nodes offline and modify the cluster configuration. Nodes are removed one at a time, so as to avoid interrupting services.
Removing specified nodes	Removes specified cluster nodes to optimize costs.	

Impact on Billing

For a pay-per-use cluster, you can see its new price when confirming the scale-in on the console. After the scale-in is complete, the cluster will be billed based on the new price. For details, see **Cloud Search Service Price Calculator**.

Constraints

- During a scale-in operation, the data on the to-be-removed nodes needs to be migrated to the remaining nodes. The timeout for data migration per node is 48 hours. Scale-in will fail if this timeout expires.
- When the cluster has large quantities of data, you are advised to manually adjust the data migration rate and avoid performing the migration during peak hours.
- After a scale-in operation is complete, the cluster's disk usage must be lower than 80%.
- Rules for scaling in a cluster that has no master nodes:
  - Condition: Scale-in is allowed only when **the number of data nodes + the number of cold data nodes ≥ 3**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

  - Maximum number of nodes that can be removed during a single scale-in operation: You cannot remove half or more of the cluster's current data nodes and cold data nodes.

    For example, if a cluster has three data nodes, three client nodes, and three cold data nodes, a maximum of two nodes can be removed at a time. Formula: (3+3)/2 = 3; and the number of nodes that can be removed should be less than 3, so the correct result is 2.
  - Data reliability: After a scale-in operation, the total number of remaining data nodes and cold data nodes must be greater than the maximum number of index replicas in the cluster.

    For example, if the maximum number of replicas for all indexes in the cluster is 2, the total number of data nodes and cold data nodes after the scale-in must be at least 3.
- Rules for scaling in a cluster that has master nodes:
  - Non-master nodes (data nodes, client nodes, or cold data nodes): For each node type, the number of nodes must be at least 2 before you can proceed with a scale-in operation.
  - Master nodes: For a single scale-in operation, you can only remove less than half of the master nodes.

    For example, if a cluster has two data nodes and five master nodes, only two master nodes can be removed for the current scale-in operation. Formula: 5/2 = 2.5; and the number of nodes that can be removed should be less than 2.5, so the correct result is 2.
- Rules for removing nodes from a multi-AZ cluster:
  - Even distribution: After a scale-in operation, the difference in the number of same-type nodes across different AZs must not exceed 1.
  - Minimum number of nodes required to maintain high availability:
    - **For a two-AZ cluster**, there must be at least four data nodes (or cold data nodes), two for each AZ; and there must be at least two client nodes (if configured), one for each AZ.
    - **For a three-AZ cluster**, each available node type must have at least one node in each AZ. This means each node type must have at least three nodes in total.
  - Performance enhancement: To prevent uneven data distribution and ensure optimal query and ingestion performance, the number of data nodes or cold data nodes should be an integer multiple of the number of AZs.
- For the range of node quantities supported by each node type, see **Table 2-214**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-214 Node quantity ranges

Node Type	Value Range
Data nodes	• Without master nodes: 1 to 32 • With master nodes: 1 to 200 (If the combined number of data nodes and cold data nodes exceeds 100, you are advised to select a flavor with at least 16 vCPUs. This helps ensure optimal read and write performance.)
Master nodes	3, 5, 7, or 9 (must be an odd number from 3 to 9)
Client nodes	1~64
Cold data nodes	1–32

Change Impact

Before the change, learn about possible impacts and operation suggestions, and develop a plan to minimize these impacts.

- Impact on performance

  During a scale-in, shards on the to-be-removed nodes are migrated to the remaining nodes. This process will consume I/O performance. This is why you are advised to perform the operation during off-peak hours.

  To minimize this impact, it is advisable to adjust the data migration rate based on the cluster's traffic cycle: increase the data migration rate during off-peak hours to shorten the task duration, and decrease it **before** peak hours arrive to ensure optimal cluster performance. The data migration rate is determined by the **indices.recovery.max_bytes_per_sec** parameter. The default value of this parameter is the number of vCPUs multiplied by 8 MB. For example, for four vCPUs, the data migration rate is 32 MB/s. You can adjust it based on the service requirements.

  ```
  PUT /_cluster/settings
  {
    "transient": {
      "indices.recovery.max_bytes_per_sec": "128MB"
    }
  }
  ```

- Impact on cluster load

  After a scale-in, the remaining nodes will need to handle all of the cluster's load. This may lead to higher CPU, memory, and disk I/O usage, impacting query and ingestion performance. If shards are unevenly allocated, performance bottlenecks may occur. This is why before a scale-in, it is necessary to evaluate whether the remaining nodes have the capacity to handle the current cluster load.

- Characteristics of this process

  Once started, a scaling task cannot be stopped until it succeeds or fails.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Scale-in Duration

The following formula can be used to estimate how long a scale-in operation will take:

Scale-in duration (min) = 5 (min) x Number of nodes to be removed + Data migration duration (min)

where, 5 minutes indicates how long non-data migration operations (e.g., initialization) typically take per node. It is an empirical value.

Data migration duration (min) = Total data size of the nodes to be removed (MB) ÷ [Total number of vCPUs of the data nodes x 8 (MB/s) x 60 (s)]

where,

- 8 MB/s indicates that each vCPU can process 8 MB of data per second. It is an empirical value.
- The formulas above use estimates under ideal conditions. The actual migration speed depends on cluster load.

Prerequisites

- The cluster status is **Available**, and there are no ongoing tasks.
- All mission-critical data has been backed up. For details, see **Creating Snapshots to Back Up Data**.

Removing Nodes Randomly

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters** > **Elasticsearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. Click the **Scale Cluster** tab.
5. Click **Scale in** to set parameters.

   Table 2-215 Removing nodes randomly

Parameter	Description
Action	Select **Scale in**.
Resources	Quantities of resources reduced.
Nodes	Reduce the number of nodes in the **Nodes** column. You can change multiple node types at the same time. For the range of node quantities supported by each node type, see **Constraints**.

6. Click **Next**.
7. Confirm the information and click **Submit**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

8. Click **Back to Cluster List** to go back to the **Clusters** page. **Task Status** is **Scaling in**. When **Cluster Status** changes to **Available**, the cluster has been successfully scaled in.

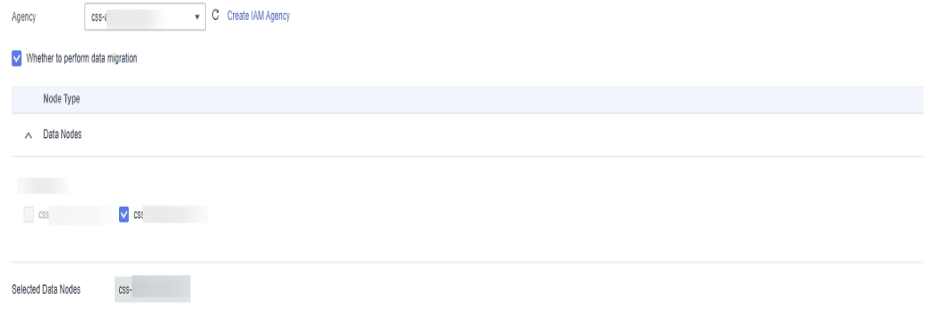
Removing Specified Nodes

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters** > **Elasticsearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. On the **Modify Configuration** page, click the **Scale In Nodes** tab.
5. Set scale-in parameters.

Table 2-216 Removing specified nodes (Scale In Nodes)

Parameter	Description
Node Type	Expand the node type that needs be changed to show all nodes under it. Select the nodes you want to remove.

Figure 2-83 Removing specified nodes



6. Click **Next**.
7. Confirm the change information and click **Submit**. In the confirm dialog box, choose to migrate data, which helps to prevent data loss, and click **OK**.

   During data migration, the system migrates all data from the to-be-removed nodes to the remaining nodes, and removes these nodes upon completion of the data migration. If the data on the to-be-removed nodes has replicas on other nodes, data migration can be skipped and the cluster change can be completed faster.
8. Click **Back to Cluster List** to go back to the **Clusters** page. **Task Status** is **Scaling in**. When **Cluster Status** changes to **Available**, the cluster has been successfully scaled in.

Related Documents

- For an Elasticsearch cluster, you can also optimize costs by changing node specifications and EVS disk types. For details, see **Changing Node Specifications**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- If you want to reduce cluster nodes even though your cluster is not eligible for a scale-in operation, you can simply create a new cluster. Then you can migrate data to the new cluster using snapshots, and delete the old cluster after data is migrated.

2.16.3 Changing Node Specifications

If the workloads on an Elasticsearch cluster's data plane change, you can scale the cluster vertically by changing its node specifications or node storage type.

Table 2-217 Change scenarios

Change Type	Scenario	Change Process
Changing node specifications	Typically, you increase node specifications instead of decreasing them. Common scenarios include: • If the allocation of new indexes or shards takes too long or the coordination and scheduling of nodes are inefficient, increase the master node specifications. • If too many requests need to be handled or too many results need to be aggregated, increase the client node specifications. • If data nodes are becoming slower in responding to data writing and query requests, increase the specifications of the data nodes. • If cold data query becomes slow, increase the cold data node specifications. • If the CPU or JVM metrics of cluster nodes show performance bottlenecks, increase node specifications to improve cluster performance. Alternatively, you may also decrease node specifications, but doing so will decrease the cluster's data processing and storage capacities. Exercise caution.	1. Bring a node offline. 2. Change the node specifications. 3. Restart the node and restore its data. 4. After the node recovers, proceed to another node and repeat the steps above. This goes on until all nodes are changed. The node specifications are changed one node at a time. This is to ensure that there are sufficient resources to keep services running.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Change Type	Scenario	Change Process
Changing the node storage type (disk type)	Change the node storage type if disk I/O has become a performance bottleneck, which impacts query and write performance.	1. Select a random node and migrate its data to other nodes. 2. Rebuild the node using the new storage type. 3. Add the node back into the cluster. The system automatically triggers a shard reallocation, migrating some of the shards to the new node. 4. After the node recovers, proceed to another node and repeat the steps above. The nodes are changed one at a time to prevent service interruptions.

Impact on Billing

For a pay-per-use cluster, you can see its new price when confirming the node specifications or storage type change on the console. After the change is complete, the cluster will be billed based on the new price. For details, see **Cloud Search Service Price Calculator**.

Constraints

- The node specifications and storage type cannot be changed for nodes that use local disks.
- The node specifications and storage type cannot be changed at the same time.
- The node storage type can be changed only for data nodes and cold data nodes.
- When you change the node storage type, data needs to be migrated between different nodes. The timeout for data migration per node is 48 hours. Upgrade will fail if this timeout expires. When the cluster has large quantities of data, you are advised to manually adjust the data migration rate and avoid performing the migration during peak hours.
- For a cluster without master nodes, the node specifications and node storage type can be changed only if the number of data nodes plus cold data nodes is at least three.
- For a cluster with master nodes, this operation is allowed only if the cluster has at least two data nodes.
- During a node storage type change, there is always one node that is unavailable. To ensure service continuity, make sure the **total number of**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

data nodes and cold data nodes is greater than the maximum number of index replicas plus 1. For a single-AZ or dual-AZ cluster, also make sure there are at least two nodes of each type for the cluster in each of its AZs.

- During a node specifications change, nodes are brought offline in order to make the changes. To ensure service continuity, make sure all shards have replicas.
- Make sure the disk usage is always less than 80% during the change.

Change Impact

Before changing a cluster's node specifications or storage type, it is essential to assess the potential impacts and review operational recommendations. This enables proper scheduling of the change, minimizing service interruptions.

- Performance impact (only when the node storage type is changed)

  Changing the node storage type does not interrupt services. However, data migration that occurs during this process consumes I/O performance, and taking individual nodes offline still has some impact on the overall cluster performance.

  To minimize this impact, it is advisable to adjust the data migration rate based on the cluster's traffic cycle: increase the data migration rate during off-peak hours to shorten the task duration, and decrease it **before** peak hours arrive to ensure optimal cluster performance. The data migration rate is determined by the **indices.recovery.max_bytes_per_sec** parameter. The default value of this parameter is the number of vCPUs multiplied by 8 MB. For example, for four vCPUs, the data migration rate is 32 MB/s. You can adjust it based on the service requirements.

  ```
  PUT /_cluster/settings
  {
    "transient": {
      "indices.recovery.max_bytes_per_sec": "128MB"
    }
  }
  ```

- Impact on request handling

  Taking nodes offline one at a time usually does not interrupt services. However, requests sent to offline nodes may fail. To mitigate this impact, the following measures may be taken:

  - Use a VPC endpoint or a dedicated load balancer to handle access requests to your cluster, which makes sure that requests are automatically routed to available nodes.
  - Enable an exponential backoff & retry mechanism on the client (configure three retries).
  - Perform this operation during off-peak hours.

- Impact on index replicas

  Shards that have no replicas will become unavailable when the nodes that store them are taken offline, causing service interruptions. You are advised to add replicas for all important indexes before making the change described in this topic.

- Impact on Kibana and Cerebro

  Changing the node storage type for a cluster will cause Kibana and Cerebro to be rebuilt. During this period, Kibana and Cerebro are temporarily unavailable.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

During a node specifications change, if Kibana and Cerebro become unavailable because the node that runs them is taken offline, refresh the web page or try to log in again, and the system will reschedule Kibana and Cerebro to an available node.

- Characteristics of this process

  Once started, a change task cannot be stopped until it succeeds or fails. A change task failure only impacts a single node, and does not interrupt services if there are data replicas, but the failed node still needs to be restored promptly.

If a change of the node specifications or storage type is urgently needed, submit a **service ticket** to contact technical support and perform a pre-change evaluation and the necessary checks.

Change Durations

- The following formulas can be used to estimate how long the node specifications change will take:

  Change duration (min) = 10 (min) x Total number of nodes to change + Data recovery duration (min)

  where,

  – 10 minutes indicates how long non-data recovery operations (e.g., initialization) typically take per node. It is an empirical value.

  – The total number of nodes is the sum of the number of data nodes, master nodes, client nodes, and cold data nodes in the cluster.

  Data recovery duration (min) = Total data size (MB)/[Total number of vCPUs of the data nodes x 8 (MB/s) x 60 (s)]

  where,

  – 8 MB/s indicates that each vCPU can process 8 MB of data per second. It is an empirical value.

  – The formulas above use estimates under ideal conditions. The actual data recovery speed depends on cluster load.

- The following formulas can be used to estimate how long the node storage type change will take:

  Change duration (min) = 15 (min) x Total number of nodes to change + Data migration duration (min)

  where,

  – 15 minutes indicates how long non-data migration operations (e.g., initialization) typically take per node. It is an empirical value.

  – The total number of nodes is the sum of the number of data nodes, master nodes, client nodes, and cold data nodes in the cluster.

  Data migration duration (min) = Total data size (MB)/[Total number of vCPUs of the data nodes x 8 (MB/s) x 60 (s)]

  where,

  – 8 MB/s indicates that each vCPU can process 8 MB of data per second. It is an empirical value.

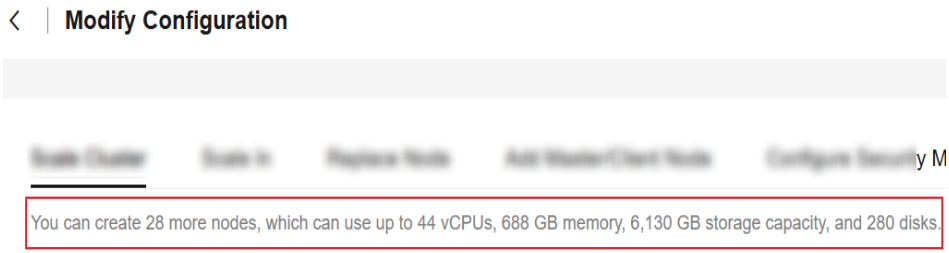
  – The formulas above use estimates under ideal conditions. The actual migration speed depends on cluster load.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Prerequisites

- The cluster status is **Available**, and there are no ongoing tasks.
- Your CSS resource quotas are sufficient for the capacity expansion you are about to perform. You can check available resources on the **Modify Configuration** page.

  Figure 2-84 Checking available resources

- All mission-critical data has been backed up before a node storage type change. This is to prevent data loss. For details, see **Creating Snapshots to Back Up Data**.

Changing the Node Specifications and Storage Type

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters** > **Elasticsearch**.
3. Check that all service data has replicas so that services will not be interrupted during the change.
   a. In the displayed cluster list, find the target cluster, and click **Access Kibana** in the **Operation** column to log in to the Kibana console.
   b. In the left navigation pane, choose **Dev Tools**.
   c. Run the **GET _cat/indices?v** command in Kibana.
      - If the returned **rep** value is greater than **0**, data replicas exist. Go to the next step.
      - If the returned **rep** value is 0, there are no data replicas. You are advised to manually create a snapshot for the cluster before moving on to the next step. For details, see **Creating Snapshots to Back Up Data**.
4. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
5. Click the **Scale Cluster** tab.
6. Click **Change specifications** to set parameters.

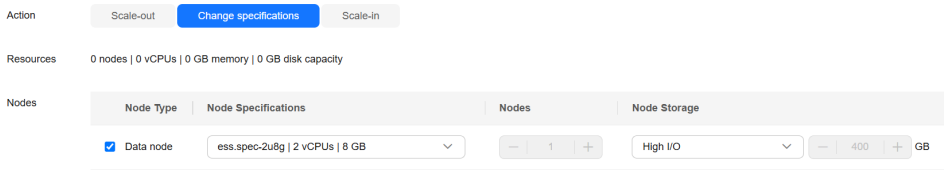
   Table 2-218 Changing the node specifications and storage type

Parameter	Description
Action	Select **Change specifications**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Resources	Shows the change of resources for this operation.
Nodes	Configure the changes you want to make. 1. Select a node type in the **Node Type** column. 2. Select a new flavor in the **Node Specifications** column, or select a new storage type in the **Node Storage** column. The node specifications and storage type cannot be changed at the same time.

Figure 2-85 Changing node specifications



7. Click **Next**.
8. Confirm the information and click **Submit**.
9. In the displayed dialog box, confirm check items, and click **OK** to start the specifications change.
   - Check items for node specifications change: **Verify index copies & nodes** and **Cluster status check**.
   - Check items for node storage type change: **Check cluster load**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-219 Check item description

Item	Description
Verify index replicas & nodes	• Verifying the existence of index replicas helps improve data security. You can skip this step, but the absence of index replicas may impact data reliability during a node specifications change. • Verifying data nodes ensures the cluster has enough capacity to handle requests. Skipping this check risks service interruptions when changing node specifications. Proceed with caution. Verification rules: • If you select **Verify index copies & nodes** and the cluster has no master nodes, each index is required to have at least one replica and the cluster must have at least three data nodes plus cold data nodes. • If you select **Verify index copies & nodes** and the cluster has a master node, each index is required to have at least one replica, but there is no requirement on how many nodes the cluster must have.
Cluster status check	During a node specifications change, the cluster status is checked by default to improve the success rate and ensure data security. The nodes are changed one at a time. For each node, the system changes its specifications, restarts it, and checks that all its processes are started successfully before moving on to the next node. In emergencies (for example, when a cluster is overloaded and services are faulty, which may prevent a specifications change request from being delivered), you can skip cluster status check so that more resources can be made available for cluster recovery. However, doing so may cause the cluster to become faulty and interrupt services. Exercise caution.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Item	Description
Check cluster load	During a node storage type change, data migration between nodes and the stopping and restarting of nodes will consume cluster resources, causing the cluster load to increase. A cluster load check can identify possible overload risks for a cluster and reduce the likelihood of an overload condition causing the node storage type change to fail. The cluster load check items are as follows: • **nodes.thread_pool.search.queue < 1000**: Check that the maximum number of requests in the search queue is less than 1000. • **nodes.thread_pool.write.queue < 200**: Check that the maximum number of requests in the write queue is less than 200. • **nodes.process.cpu.percent < 90**: Check that the maximum CPU usage is less than 90%. • **nodes.os.cpu.load_average/Number of vCPUs < 80%**: Check that the number of running processes plus the number of processes waiting for CPUs is less than 80% of the total number of vCPUs.

NOTE

If the change request fails to be submitted and a message is displayed indicating that the cluster needs to be upgraded, it means the current cluster version does not support a node storage type change. Upgrade the cluster to the latest image version and then try again. For details, see **Upgrading the Cluster Version**.

10. Click **Back to Cluster List** to go back to the **Clusters** page. The **Cluster Status** is **Configuration modified**. When **Cluster Status** changes to **Available**, the cluster node specifications have been successfully changed.

2.16.4 Replacing a Faulty Node

If a node in an Elasticsearch cluster is faulty, you can replace it to restore services.

The node replacement process is as follows:

1. Migrate data from the node that needs to be replaced to other available nodes.
2. Rebuild a new node using this node's current ID, IP address, specifications, and AZ.
3. Add the new node into the cluster. The system automatically triggers a shard reallocation, migrating some of the shards to the new node.

This process does not interrupt services because data is migrated from the replaced node to other available nodes.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Constraints

- Only one node can be replaced at a time. Each new node is rebuilt using the ID, IP address, specifications, and AZ of the node it is replacing.
- The configurations you modified manually will not be retained after node replacement. For example, if you have manually added a return route for the original node, you need to add it again for the new node after the node replacement is complete.
- If the node you want to replace is a data node or cold data node, pay attention to the following precautions:
  - When a data node or cold data node is replaced, its data is first migrated to other data nodes. This means the **total number of data nodes and cold data nodes must be greater than the maximum number of index replicas plus 1**.
  - Elasticsearch clusters whose version is earlier than 7.6.2 cannot have closed indexes. Otherwise, their data nodes or cold data nodes cannot be replaced.
  - In the AZ that contains the data node or cold data node to be replaced, there has to be at least another data node or cold data node.
  - If the cluster has no master nodes, the total number of data nodes plus cold data nodes must be at least three.
  - The precautions above do not apply if you are replacing a faulty node, regardless of its type. This is because faulty nodes are not included in **_cat/nodes**.

Change Impact

Before replacing a node, it is essential to assess the potential impacts and review operational recommendations. This enables proper scheduling of the node replacement, minimizing service interruptions.

- Impact on performance

  Replacing a node does not interrupt services. However, data migration that occurs during this process consumes I/O performance, and taking individual nodes offline still has some impact on the overall cluster performance.

  To minimize this impact, it is advisable to adjust the data migration rate based on the cluster's traffic cycle: increase the data migration rate during off-peak hours to shorten the task duration, and decrease it **before** peak hours arrive to ensure optimal cluster performance. The data migration rate is determined by the **indices.recovery.max_bytes_per_sec** parameter. The default value of this parameter is the number of vCPUs multiplied by 8 MB. For example, for four vCPUs, the data migration rate is 32 MB/s. You can adjust it based on the service requirements.

  ```
  PUT /_cluster/settings
  {
    "transient": {
      "indices.recovery.max_bytes_per_sec": "128MB"
    }
  }
  ```

- Impact on request handling

  While a node is being replaced, requests sent to it may fail. To mitigate this impact, the following measures may be taken:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

  - Use a VPC endpoint or a dedicated load balancer to handle access requests to your cluster, which makes sure that requests are automatically routed to available nodes.
  - Enable an exponential backoff & retry mechanism on the client (configure three retries).
  - Perform node replacement during off-peak hours.
- Characteristics of this process

  Once started, a node replacement task cannot be stopped until it succeeds or fails. A task failure only impacts a single node, and does not interrupt services if there are data replicas, but the failed node still needs to be restored promptly.

Node Replacement Duration

The following formula can be used to estimate how long it will take to replace a specified node of a cluster:

Change duration (min) = 15 (min) + Data migration duration (min)

where, 15 minutes indicates how long non-data migration operations (e.g., initialization) typically take per node. It is an empirical value.

Data migration duration (min) = Total data size (MB)/[Total number of vCPUs of the data nodes x 8 (MB/s) x 60 (s)]

where,

- 8 MB/s indicates that each vCPU can process 8 MB of data per second. It is an empirical value.
- The formulas above use estimates under ideal conditions. The actual migration speed depends on cluster load.

Prerequisites

- The cluster status is **Available**, and there are no ongoing tasks.
- All mission-critical data has been backed up. For details, see **Creating Snapshots to Back Up Data**.

Replacing a Specified Node

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. On the **Modify Configuration** page, click the **Replace Node** tab.
5. On the **Replace Node** tab, set the parameters as needed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-220 Replacing a specified node

Parameter	Description
Node Type	Select the node you want to replace. You can expand a node type to check all the nodes under it.

Figure 2-86 Replacing a specified node



6. Click **Submit**. In the data migration confirmation dialog box, choose to migrate data, and click **OK**.

   During data migration, the system migrates all data from the to-be-replaced node to the remaining nodes, and replaces the node upon completion of the data migration. If the data on the to-be-replaced nodes has replicas on other nodes, data migration can be skipped and the cluster change can be completed faster.

7. Click **Back to Cluster List** to go back to the **Clusters** page. The **Task Status** is **Replacing nodes**. When **Cluster Status** changes to **Available**, the node has been successfully replaced.

2.16.5 Upgrading the Cluster Version

In CSS, you can upgrade the version of an Elasticsearch cluster to use feature and performance enhancements or fix known issues.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-221 Upgrade scenarios

Upgrade Type	Scenario	Upgrade Process
Same-version upgrade	Update the cluster's kernel patches. The cluster is upgraded to the latest image of the current version to fix known issues or optimize performance. For example, if the current cluster version is **7.10.2(7.10.2_24.3.3_0102)**, upon a same-version upgrade, the cluster will be upgraded to the latest image **7.10.2(7.10.2_24.3.4_0109)** of version 7.10.2. (The version numbers used here are examples only.)	1. Bring a node offline, and migrate its data to other nodes. 2. Upgrade the node to the target version. 3. Add the node back into the cluster. The system automatically triggers a shard reallocation, migrating some of the shards to the new node. 4. Confirm that the node status is normal. Then upgrade the remaining nodes one by one. Nodes in clusters are upgraded one by one to avoid interrupting services.
Cross-version upgrade	Upgrade a cluster to the latest image of the target version to enhance functionality or normalize software versions. For example, if the current cluster version is **7.6.2(7.6.2_24.3.3_1224)**, upon a cross-version upgrade, the cluster will be upgraded to the latest image **7.10.2(7.10.2_24.3.4_0109)** of version 7.10.2. (The version numbers used here are examples only.)	1. Bring a node offline, and migrate its data to other nodes. 2. Upgrade the node to the target version. 3. Add the node back into the cluster. The system automatically triggers a shard reallocation, migrating some of the shards to the new node. 4. Confirm that the node status is normal. Then upgrade the remaining nodes one by one. Nodes in clusters are upgraded one by one to avoid interrupting services.

Version Support

The supported target versions vary depending on the current version. For details, see **Table 2-222**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-222 Version upgrade paths

Current Version	Target Version
Elasticsearch: 6.2.3	Elasticsearch: 6.5.4 or 6.8.23
Elasticsearch: 6.5.4	Elasticsearch: 6.8.23
Elasticsearch: 6.8.23	Elasticsearch: 7.6.2 or 7.10.2
Elasticsearch: 7.1.1	Elasticsearch: 7.6.2 or 7.10.2
Elasticsearch: 7.6.2	Elasticsearch: 7.10.2
Elasticsearch: 7.9.3	Elasticsearch: 7.10.2
Note: • Elasticsearch 7.10.2 is the current major cluster version. You are advised to normalize your clusters to this version. The supported target versions are displayed in the drop-down list of **Target Image**. • Elasticsearch clusters of versions 6.2.3 and 6.5.4 can be upgraded to 6.8.23 and then to 7.*X.X*.	

Constraints

- A maximum of 20 clusters can be upgraded at the same time.
- During an upgrade, the data on the upgraded nodes needs to be migrated to other nodes. The timeout for data migration per node is 48 hours. Upgrade will fail if this timeout expires. When the cluster has large quantities of data, you are advised to manually adjust the data migration rate and avoid performing the migration during peak hours.

Upgrade Impact

Before upgrading a cluster, it is essential to assess the potential impacts and review operational recommendations. This enables proper scheduling of the upgrade time, ensuring a smooth upgrade process and minimizing service interruptions.

- Impact on performance

  The nodes of a cluster are upgraded one at a time to ensure service continuity. However, data migration that occurs during the upgrade consumes I/O performance, and taking individual nodes offline still has some impact on the overall cluster performance.

  To minimize this impact, it is advisable to adjust the data migration rate based on the cluster's traffic cycle: increase the data migration rate during off-peak hours to shorten the task duration, and decrease it **before** peak hours arrive to ensure optimal cluster performance. The data migration rate is determined by the **indices.recovery.max_bytes_per_sec** parameter. The default value of this parameter is the number of vCPUs multiplied by 8 MB. For example, for four vCPUs, the data migration rate is 32 MB/s. You can adjust it based on the service requirements.

  ```
  PUT /_cluster/settings
  {
  ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  "transient": {
    "indices.recovery.max_bytes_per_sec": "128MB"
  }
}
```

- Impact on request handling

  During the upgrade, nodes are upgraded one at a time. Requests sent to the node that is being upgraded may fail. To mitigate this impact, the following measures may be taken:

  – Use a VPC endpoint or a dedicated load balancer to handle access requests to your cluster, which makes sure that requests are automatically routed to available nodes.

  – Enable an exponential backoff & retry mechanism on the client (configure three retries).

  – Perform the upgrade during off-peak hours.

- Rebuilding Kibana and Cerebro

  Kibana and Cerebro will be rebuilt during the upgrade, making them temporarily unavailable. Additionally, due to cross-version compatibility issues, Kibana may become unavailable during the upgrade. These problems will go away once the upgrade is completed.

- Characteristics of the upgrade process

  Once started, an upgrade task cannot be stopped until it succeeds or fails. An upgrade failure only impacts a single node, and does not interrupt services if there are data replicas.

Upgrade Duration

The following formula can be used to estimate how long it takes to upgrade a cluster:

Upgrade duration (min) = 15 (min) x Total number of nodes to be upgraded + Data migration duration (min)

where,

- 15 minutes is how long non-data migration operations (e.g., initialization and image upgrade) typically take per node. It is an empirical value.
- The total number of nodes is the sum of the number of data nodes, master nodes, client nodes, and cold data nodes in the cluster.

Data migration duration (min) = Total data size (MB)/[Total number of vCPUs of the data nodes x 8 (MB/s) x 60 (s) x Concurrency]

where,

- Typically, when concurrency is higher than 1, the actual data migration rate can only reach half of the throughput intended by the configured concurrency, depending on the cluster load.
- 8 MB/s indicates that each vCPU can process 8 MB of data per second. It is an empirical value.

The formulas above use estimates under ideal conditions. In practice, it is advisable to add 20% to 30% of redundancy.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Pre-Upgrade Check

To ensure a successful upgrade, you must check the items listed in the following table before performing an upgrade.

Table 2-223 Pre-upgrade checklist

Check Item	Check Method	Description	Normal Status
Cluster status	System check	After an upgrade task is started, the system automatically checks the cluster status. Clusters whose status is **green** or **yellow** can provide services properly and have no unallocated primary shards.	The cluster is available and there are no ongoing tasks.
Node quantity	System check	After an upgrade task is started, the system automatically checks node quantities. To ensure service continuity, there has to be at least two nodes of each type for a cluster in each of its AZs. For a cluster with master nodes, there has to be at least two data nodes. For a cluster without master nodes, the number of data nodes plus cold data nodes must be three at least.	• Number of nodes of each type in each AZ ≥ 2 • For a cluster with master nodes: Number of data nodes ≥ 2 • For a cluster without master nodes: Number of data nodes + Number of cold data nodes ≥ 3
Disk capacity	System check	After an upgrade task is started, the system automatically checks the disk capacity. During the upgrade, nodes are brought offline one by one and then new nodes are created. Ensure that the total disk capacity of the remaining nodes is sufficient for handling all the data of the cluster, and that the nodes' disk usage stays below 80%.	After a single node is brought offline, the total disk capacity of the remaining nodes is sufficient for handling all the data of the cluster, and the nodes' disk usage stays below 80%.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Check Item	Check Method	Description	Normal Status
Data replicas	System check	Check that the remaining data nodes and cold data nodes can handle the maximum number of primary and standby shards of indexes in the cluster. There should not be unallocated replicas during the upgrade.	Number of data nodes + Number of cold data nodes > Maximum number of index replicas + 1
Data backup	System check	Before the upgrade, back up data to prevent data loss caused by upgrade failures. When submitting an upgrade task, you can choose whether to check for full index snapshots.	Check whether data has been backed up.
Resources	System check	After an upgrade task is started, the system automatically checks resources. New resources will be created during the upgrade. Ensure that resources are available.	Resources are available and sufficient.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Check Item	Check Method	Description	Normal Status
Custom plugins	System and manual check	Perform this check only if custom plugins are installed in the source cluster. If a cluster has a custom plugin, upload all plugin packages of the target version on the plugin management page before the upgrade. During the upgrade, install the custom plugin in the new nodes. Otherwise, the custom plugins will be lost after the cluster is successfully upgraded. After an upgrade task starts, the system automatically checks whether the required custom plugin packages have been uploaded. However, you must verify that the uploaded plugin packages are correct and compatible with the target version. If a plugin package is invalid or incompatible, the upgrade will fail. **NOTE** After the upgrade is complete, the status of the custom plugin is reset to **Uploaded**.	The plugin package of the cluster to be upgraded has been uploaded to the plugin list.
Custom configurations	System check	During the upgrade, the system automatically synchronizes the content of the cluster configuration file **elasticsearch.yml**.	Clusters' custom configurations are not lost after the upgrade.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Check Item	Check Method	Description	Normal Status
Non-standard operations	Manual check	Check whether non-standard operations have been performed in the cluster. Non-standard operations refer to manual operations that are not recorded. These operations cannot be automatically passed on during the upgrade, for example, modification of the **kibana.yml** configuration file, system settings, and return routes.	Some non-standard operations are compatible. For example, the modification of a security plugin can be retained through metadata, and the modification of system configuration can be retained using images. Some non-standard operations, such as the modification of the **kibana.yml** file, cannot be retained, and you must back up the file in advance.
Compatibility check	System and manual check	After a cross-version upgrade task is started, the system automatically checks whether the source and target versions have incompatible configurations. If a custom plugin is installed for a cluster, the version compatibility of the custom plugin needs to be manually checked.	Configurations before and after the cross-version upgrade are compatible.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Check Item	Check Method	Description	Normal Status
Check Cluster Loads	System and manual check	If the cluster is heavily loaded, there is a high probability that the upgrade will get stuck or fail. You are advised to check the cluster load before the upgrade and perform the upgrade only during off-peak hours. You can also choose to check the cluster load while configuring upgrade information.	• **nodes.thread_pool.search.queue < 1000**: Check that the maximum number of requests in the search queue is less than 1000. • **nodes.thread_pool.write.queue < 200**: Check that the maximum number of requests in the write queue is less than 200. • **nodes.process.cpu.percent < 90**: Check that the maximum CPU usage is less than 90%. • **nodes.os.cpu.load_average/Number of vCPUs < 80%**: Check that the number of running processes plus the number of processes waiting for CPUs is less than 80% of the total number of vCPUs.

Performing the Upgrade Task

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Cluster Snapshots** tab, and perform a full data backup. For details, see **Manually Creating a Snapshot**.

   When creating an upgrade task, you can choose to check whether the full index data has been backed up using snapshots. This helps to prevent data loss in case of an upgrade failure.
5. Click the **Version Upgrade** tab, and set upgrade parameters.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-224 Upgrade parameters

Parameter	Description
Upgrade Type	Select an upgrade type. • Same-version upgrade: upgrade kernel patches to the latest images within the current cluster version. • Cross-version upgrade: upgrade a cluster to the latest image of the target version.
Target Image	Image of the target version. When you select an image, the image name and target version details are displayed. The supported target versions are displayed in the drop-down list of **Target Image**. If no target image is available, possible causes are as follows: • The current cluster is of the latest version. • The current cluster was created before 2023 and has vector indexes. • The new version images are unavailable in the current region. • The current cluster does not support the upgrade type you have selected.

6. Click **Submit**.
7. In the displayed dialog box, configure upgrade check and acceleration options.
   a. Select whether to enable **Check Full Index Snapshot**.

      When enabled, CSS checks for valid snapshots by matching indexes by name. Snapshots help prevent potential data loss caused by upgrade failures.

      CSS cannot check the content or backup times of snapshots. You should manually check existing snapshots. If any of them is over one month old, create the latest snapshot.

   b. Select whether to enable **Check Cluster Loads**.

      During an upgrade, data migration and node restarts will consume cluster resources and increase loads. When this option is enabled, CSS evaluates overload risks for the cluster and reduces the likelihood of a cluster upgrade failure caused by an overload.

      The check items are as follows:

      - **nodes.thread_pool.search.queue < 1000**: Check that the maximum number of requests in the search queue is less than 1000.
      - **nodes.thread_pool.write.queue < 200**: Check that the maximum number of requests in the write queue is less than 200.
      - **nodes.process.cpu.percent < 90**: Check that the maximum CPU usage of cluster nodes is less than 90%.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- **nodes.os.cpu.load_average/Number of vCPUs < 80%**: Check that the number of running processes plus the number of processes waiting for CPUs is less than 80% of the total number of vCPUs.

If any of the results is abnormal, wait until the load drops or actively optimize it before performing the upgrade.

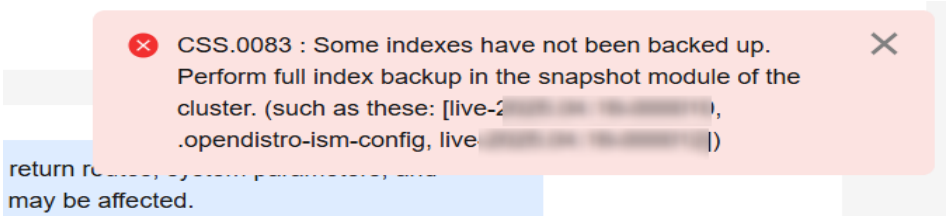
c. Set **Data migration concurrency control**.

Increasing the data migration concurrency can accelerate the upgrade process, but faster data migration leads to higher I/O usage. A higher concurrency is likely to lead to higher cluster load, which may impact cluster performance. You are advised to retain the default value **1**. The value must be less than half the number of data nodes.

8. Click **OK**. The system performs a pre-upgrade check. If the check passes, the upgrade process begins.

   If the pre-upgrade check fails, error information is displayed in the upper right corner of the console. Adjust the cluster configuration accordingly and try again.

   Figure 2-87 Upgrade check errors

   When **Task Status** in the task list below changes to **Successful**, the upgrade is completed.

9. Confirm the upgrade result.

   In the cluster list, find the target cluster, and check the cluster version in the **Version** column to see if the upgrade is successful.

Checking the Update Task

On the upgrade page, you can check the current upgrade task in the **Upgrade Records** area to learn the task progress and status.

Expand the task list and click **View Progress** to check the upgrade progress and node status.

If **Task Status** is **Failed**, you can retry the task or terminate it.

- Retry the task: Click **Retry** in the **Operation** column.
- Terminate the task: Click **Terminate** in the **Operation** column. Before terminating a cross-version upgrade task, ensure that none of the nodes have been upgraded.

  After an upgrade task is terminated, nodes that have failed to be upgraded will remain in the current state, while those that have been successfully upgraded will not be rolled back to the old version. As a result, the nodes in the same cluster may have different versions. You are advised to restart the

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

upgrade as soon as relevant issues are handled, ensuring that all nodes have the same version.

2.16.6 Switching AZs

CSS supports cross-AZ deployment. By switching AZs for a cluster, you can make more resources available to the cluster or improve the cluster's availability.

You can **Add AZ** or **Migrate AZ**.

- **Add AZ**: Add one or two AZs for a single-AZ cluster, or add one AZ for a dual-AZ cluster to enhance cluster availability.
- **Migrate AZ**: Migrate from one AZ to another that has more sufficient resources.

The AZ change process is as follows:

1. Migrate data from one node to other available nodes.
2. Rebuild the node for the target AZ. Keep the node ID, IP address, and specifications unchanged.
3. Add the new node into the cluster. The system automatically triggers a shard reallocation, migrating some of the shards to the new node. Then repeat these steps on all other nodes.

The nodes are changed one at a time to prevent service interruptions.

Constraints

- To ensure service continuity, the total number of data nodes plus cold data nodes in the cluster must be at least three.
- During the change, nodes are brought offline one by one and then new nodes are created. Ensure that the total disk capacity of the remaining nodes is sufficient for handling all the data of the cluster, and that the nodes' disk usage stays below 80%.
- To make sure that all index shards in the cluster can be allocated to the remaining data nodes and cold data nodes, the **total number of data nodes and cold data nodes must be greater than the maximum number of index replicas plus 1**.
- Before a change completes, some nodes may have already been moved to a new AZ. In this case, the AZs before and after the change are both displayed. After the change succeeds, the new AZs and their nodes will be displayed properly.
- When you **add an AZ** for a cluster, the current AZ must be retained. When adding one or two AZs for a single-AZ cluster, you must change AZs for all nodes. When adding one AZ for a dual-AZ cluster, you can choose to change AZs for specific types of nodes or all nodes in the cluster. For example, for a dual-AZ cluster, you can use three AZs for its master nodes, while still using two AZs for the rest of its nodes.
- When changing AZs, the cluster is rebuilt by minimizing the number of migrated nodes. For a specific node type, the AZ attribute is updated for migrated nodes, while the YML configurations are also updated for unmigrated nodes. Therefore, all nodes of that type must be restarted for the changes to take effect. Exception: When master nodes and data nodes co-

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

exist, modifying only data nodes requires restarting both the master and data nodes. For other node types, you only need to restart nodes of that type.

- When adding AZs for an existing cluster for high availability, follow the node allocation rules below:
  - To ensure high service availability and prevent individual node overload in a multi-AZ deployment, the following node count requirements must be met:
    - **For a two-AZ cluster**, there must be at least four data nodes (or cold data nodes), two for each AZ; and there must be at least two client nodes (if configured), one for each AZ.
    - **For a three-AZ cluster**, each available node type must have at least one node in each AZ. This means each node type must have at least three nodes in total.
    - For node types that are not deployed (for example, no cold data nodes), the minimum node count requirement does not apply.
  - Performance enhancement: To prevent uneven data distribution and ensure optimal query and ingestion performance, the number of data nodes or cold data nodes should be an integer multiple of the number of AZs.
- When migrating an AZ (**Migrate AZ**), you can select only one target AZ. For this operation, you can choose to migrate only specific types of nodes or all nodes in the cluster. For example, for a dual-AZ cluster, you may move the master nodes from one AZ to the other, while keeping the rest of the nodes unchanged. If you migrate from one AZ to another, there is no need to restart the cluster. If multiple AZs are migrated, you must restart the cluster to make the change take effect.

Change Impact

Before changing a cluster's AZs, it is essential to assess the potential impacts and review operational recommendations. This enables proper scheduling of the change, minimizing service interruptions.

- Performance

  Changing AZs for a cluster does not interrupt services. However, data migration that occurs during this process consumes I/O performance, and taking individual nodes offline still has some impact on the overall cluster performance.

  To minimize this impact, it is advisable to adjust the data migration rate based on the cluster's traffic cycle: increase the data migration rate during off-peak hours to shorten the task duration, and decrease it **before** peak hours arrive to ensure optimal cluster performance. The data migration rate is determined by the **indices.recovery.max_bytes_per_sec** parameter. The default value of this parameter is the number of vCPUs multiplied by 8 MB. For example, for four vCPUs, the data migration rate is 32 MB/s. You can adjust it based on the service requirements.

```
PUT /_cluster/settings
{
  "transient": {
    "indices.recovery.max_bytes_per_sec": "128MB"
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
 }
}
```

- Impact on request handling

  While a node is offline, requests sent to it may fail. To mitigate this impact, the following measures may be taken:

  - Use a VPC endpoint or a dedicated load balancer to handle access requests to your cluster, which makes sure that requests are automatically routed to available nodes.
  - Enable an exponential backoff & retry mechanism on the client (configure three retries).
  - Perform this operation during off-peak hours.

- Characteristics of this process

  Once started, an AZ change task cannot be stopped until it succeeds or fails.

Change Duration

The following formula can be used to estimate how long it will take to change AZs for a cluster:

Change duration (min) = 15 (min) x Total number of nodes involved + Data migration duration (min)

where, 15 minutes indicates how long non-data migration operations (e.g., initialization) typically take per node. It is an empirical value.

Data migration duration (min) = Total data size of the nodes to be migrated (MB) ÷ [Total number of vCPUs of the data nodes x 8 (MB/s) x 60 (s)]

where,

- 8 MB/s indicates that each vCPU can process 8 MB of data per second. It is an empirical value.
- The formulas above use estimates under ideal conditions. The actual migration speed depends on cluster load.

Prerequisites

- The cluster status is **Available**, and there are no ongoing tasks.
- Make sure that the cluster has not had any non-standard modifications, such as the configuration of custom return routes and system parameters. Such modifications, if ever made, will be lost after the AZ change, which may affect your services.
- All mission-critical data has been backed up. For details, see **Creating Snapshots to Back Up Data**.

Changing AZs

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

4. Click the **Change AZ** tab.
5. On the **Change AZ** page, set parameters.

CAUTION

If the number of data nodes plus cold data nodes in a cluster is not divisible by the number of AZs, data in the cluster may be unevenly distributed after the AZ change is completed. For example, if the cluster has two AZs and three data nodes, one node may need to store twice as much data as the other two nodes. This uneven distribution of data may affect the cluster's query and ingestion performance.

Table 2-225 Parameters for changing AZs

Parameter	Description
Operation Type	• **Add AZ**: Add one or two AZs for a single-AZ cluster, or add one AZ for a dual-AZ cluster. • **Migrate AZ**: Migrate data from one AZ to another.
Node Type	Select a node type for the AZ change. Select one node type or **All nodes** to change their AZ. When adding one or two AZs for a single-AZ cluster, you can only select **All nodes**.
Current AZ	Current AZ of a cluster
Target AZ	AZ to add or migrate to. • **Add AZ**: Select up to three AZs, which must include all your current AZs. • **Migrate AZ**: Select only one target AZ, which cannot be your current AZ.

6. Click **Submit**. In the displayed dialog box, choose whether to check for the presence of a full index snapshot, and click **OK** to start the change.

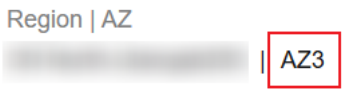
   NOTE

   You are advised to select **Check full index snapshot**. This ensures that all data has been backed up, so that in case the task fails, the data can be restored using this snapshot.

   When **Task Status** in the task list below changes to **Successful**, the AZ change is completed.
7. (Optional) Except for single-AZ migration, all other AZ change scenarios require a cluster restart to take effect.
8. Confirm the result.
   a. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
   b. On the **Overview** tab, check the cluster's AZs in the **Basic Information** area to determine whether the change is successful.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 2-88 Checking the cluster's AZs



Checking the Progress of an AZ Change Task

In the **Jobs** area on the **Change AZ** tab, find the AZ change task.

Expand the task list and click **View Progress** to check the task progress and node status.

If **Task Status** is **Failed**, you can retry the task or terminate it.

- Retry a task: Click **Retry** in the **Operation** column.
- Terminate a task: Click **Terminate** in the **Operation** column. Before terminating a task, make sure that AZ has been successfully changed for none of the nodes.

2.16.7 Changing the Security Mode

With CSS, when an Elasticsearch cluster's security needs change, you can change its security mode settings.

Configure the security mode based on the security needs of your cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-226 Cluster security modes

Cluster Type	Key Settings	Description	Applicable Scenario
Non-security mode cluster	**Security Mode**: disabled	Access to such a cluster requires no user authentication, and data will be transmitted in plaintext using HTTP.	Use when creating a cluster for internal testing or workloads that have a low security standard. • Advantage: easy to access the cluster. • Disadvantage: poor security, as anyone can access the cluster. When security mode is disabled, public network access and Kibana public network access cannot be enabled. To ensure data security, deploy the cluster in a secure environment. Do not expose the cluster's network interface to the public network.
Security-mode cluster	Security-mode cluster + HTTP: • **Security Mode**: enabled • **HTTPS Access**: disabled	Such a cluster requires user authentication. It supports access control and data encryption, and it uses HTTP to transmit data in plaintext.	Use to balance security and performance. • Advantage: User authentication improves cluster security. HTTP-based access ensures high performance of the cluster. User permissions can be configured to ensure proper isolation. • Disadvantage: Public network access is not supported. To ensure data security, deploy the cluster in a secure environment. Do not expose the cluster's network interface to the public network.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Cluster Type	Key Settings	Description	Applicable Scenario
	Security-mode cluster + HTTPS: • **Security Mode**: enabled • **HTTPS Access**: enabled	Such a cluster requires user authentication. It supports access control and data encryption, and it uses HTTPS to encrypt communication and enhance data security.	Use when security takes precedence over performance and public network access is required. • Advantage: User authentication improves cluster security. HTTPS enhances cluster security by encrypting all communication over the public network. User permissions can be configured to ensure proper isolation. • Disadvantage: When HTTPS is used, data encryption and decryption introduce computational overhead and impact the cluster's read and write performance.

Table 2-227 lists the various types of security mode changes supported for CSS clusters.

Table 2-227 Security mode change scenarios

Action	Scenario	Change Process
Switching from the non-security mode to security mode	Non-security mode → Security mode + HTTP: Change a cluster from non-security mode to security mode + HTTP.	1. Select a node and modify the Elasticsearch, Kibana, and Cerebro configuration files. 2. Restart the Elasticsearch, Kibana, and Cerebro processes and restore data. 3. After the node recovers, proceed to another node and repeat the steps above. This goes on until all nodes are changed.
	Non-security mode → Security mode + HTTPS: Change a cluster from non-security mode to security mode + HTTPS.	

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Action	Scenario	Change Process
Switching from the security mode to non-security mode	Security mode → Non-security mode: • Change a cluster from security mode + HTTP to non-security mode. • Change a cluster from security mode + HTTPS to non-security mode.	
Switching between HTTP and HTTPS in security mode	HTTP → HTTPS: Change a cluster from security mode + HTTP to security mode + HTTPS.	
	HTTPS → HTTP: Change a cluster from security mode + HTTPS to security mode + HTTP.	

Constraints

The security mode settings can be changed only for Elasticsearch clusters created after November 2022 and whose version is 6.5.4 or later.

Change Impact

Before changing the security mode for a cluster, it is essential to assess the potential impacts and review operational recommendations. This enables proper scheduling of the change, minimizing service interruptions.

Table 2-228 Impact overview

Action	Service Interruption	Authentication Mode	Performance	Public Network Access	Security Account
Non-security mode → Security mode + HTTP	Yes	Authentication required	N/A	Disallowed	N/A
Non-security mode → Security mode + HTTPS	Yes	Authentication required	Downgraded	Allowed	N/A

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Action	Service Interruption	Authentication Mode	Performance	Public Network Access	Security Account
Security mode → Non-security mode	Yes	Authentication not required	Enhanced	Automatically disabled	Permanently deleted
HTTP → HTTPS	Yes	No change	Downgraded	Allowed	N/A
HTTPS → HTTP	Yes	No change	Enhanced	Automatically disabled	N/A

Impact description:

- The security mode can be enabled or disabled for a cluster, and a security-mode cluster can use either HTTPS or HTTP.
- Service interruption: When a cluster's security mode is changed, the cluster automatically restarts, leading to temporary service unavailability.
- Authentication mode: Changing a cluster's security mode also changes its user authentication mechanism. The client authentication logic must be updated accordingly. If client adaptation cannot be completed in a timely manner, services may be interrupted.
- Performance impact: Given the same hardware configuration, switching from HTTP to HTTPS leads to a performance loss (such as throughput) of around 20% under high concurrency. Conversely, when a cluster switches from HTTPS to HTTP, cluster performance improves.
- Public network access: Only security-mode clusters that use HTTPS support public network access.
- Security account: When the security mode is disabled for a cluster, the system permanently deletes the cluster's security account.
- Tool impact: If the security mode is changed for a cluster that has an ongoing Kibana session, a session error message will be displayed on Kibana. To rectify the error, clear the cache and launch Kibana again.

Changing the security mode for a cluster changes its accessibility mode, possibly causing service interruptions. You should perform this operation before services are brought online or when service interruptions can be tolerated.

Change Duration

The following formula can be used to estimate how long it will take to change the security mode for a cluster:

Change duration (min) = 5 (min) x Total number of nodes to change + Data recovery duration (min)

where,

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- 5 minutes indicates how long non-data recovery operations (e.g., initialization) typically take per node. It is an empirical value.
- The total number of nodes is the sum of the number of data nodes, master nodes, client nodes, and cold data nodes in the cluster.

Data recovery duration (min) = Total data size (MB)/[Total number of vCPUs of the data nodes x 8 (MB/s) x 60 (s)]

where,

- 8 MB/s indicates that each vCPU can process 8 MB of data per second. It is an empirical value.
- The formulas above use estimates under ideal conditions. The actual data recovery speed depends on cluster load.

Prerequisites

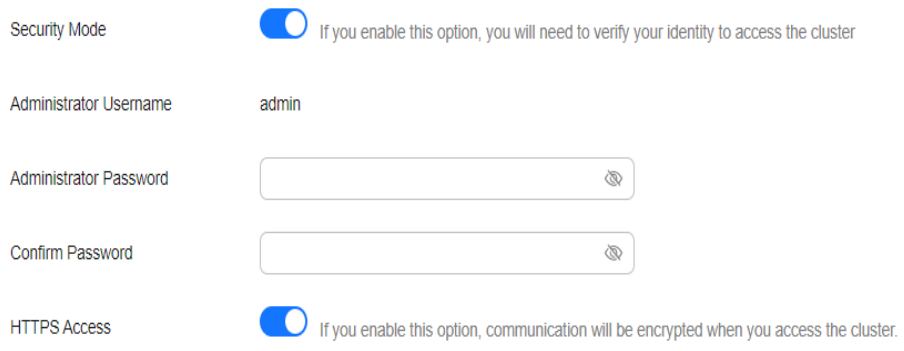
- All mission-critical data has been backed up. For details, see **Creating Snapshots to Back Up Data**.
- The cluster status is **Available**, and there are no ongoing tasks.
- Check whether load balancing is enabled for the cluster. If yes, disable load balancing first. Enable load balancing again after the security mode is changed. This prevents errors in accessing the cluster through the load balancer during the change.

Switching from the Non-Security Mode to Security Mode

You can change a non-security mode cluster to a security-mode cluster that uses HTTP or HTTPS. After a cluster's security mode is enabled, user authentication is required for accessing the cluster.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. Choose the **Change Security Mode** tab.
5. Enable the security mode. Enter and confirm the administrator password of the cluster.

Figure 2-89 Enabling the security mode



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

6. Enable or disable **HTTPS Access**.
   - If you enable **HTTPS Access**, HTTPS is used to encrypt cluster communication, and public network access can be enabled for the cluster.
   - If you disable **HTTPS Access**, HTTP is used, and public network access is disallowed for the cluster.
7. Click **Submit**, and confirm the information. The cluster list is displayed.

   The **Task Status** of the cluster is **Changing security mode**. When the cluster status changes to **Available**, the security mode has been successfully changed.

Switching from Security Mode to Non-Security Mode

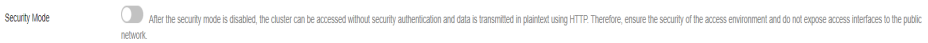
You can change a security-mode cluster that uses HTTP or HTTPS to a non-security cluster. After a cluster's security mode is disabled, user authentication is no longer required for accessing the cluster.

WARNING

- Clusters with the security mode disabled do not require user authentication, and HTTP is used to transmit data in plaintext. Make sure such clusters are deployed in a secure environment, and do not expose their network interface to the public network.
- When a cluster is switched from the security mode to non-security mode, indexes used by the security-mode cluster will be deleted. Back up data before changing the security mode.
- If a public IP address has been assigned to a security-mode cluster, unassign it before changing the security mode.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. Choose the **Change Security Mode** tab.
5. Disable the security mode.

   Figure 2-90 Disabling the security mode

6. Click **Submit**. In the displayed dialog box, confirm the information. The cluster list page is displayed.

   The **Task Status** of the cluster is **Changing security mode**. When the cluster status changes to **Available**, the security mode has been successfully changed.

Switching between HTTP and HTTPS in Security Mode

You can change the protocol of a security cluster.

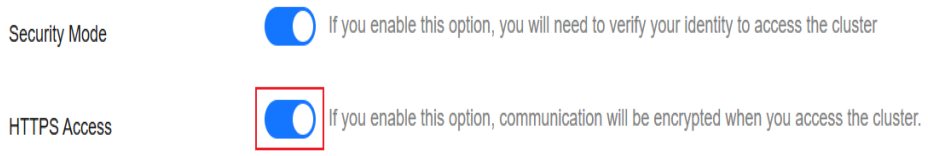
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

WARNING

If a public IP address has been assigned to a security-mode cluster, unassign it before changing from HTTPS to HTTP.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters** > **Elasticsearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. Choose the **Change Security Mode** tab.
5. Enable or disable **HTTPS Access**.

   Figure 2-91 Configuring the protocol

   - If you enable **HTTPS Access**,

     HTTPS is used to encrypt cluster communication and you can enable public network access for the cluster.
   - If you disable **HTTPS Access**, an alarm message is displayed. Click **OK** to disable the protocol.

     Cluster communication is no longer encrypted and public network access cannot be enabled.
6. Click **Submit**, and confirm the information. The cluster list is displayed.

   The **Task Status** of the cluster is **Changing security mode**. When the cluster status changes to **Available**, the security mode has been successfully changed.

2.17 Backup and Restoration

2.17.1 Creating Snapshots to Back Up Data

CSS allows you to use snapshots to back up and restore Elasticsearch cluster data. By storing a snapshot in an OBS bucket, you save a point-in-time copy of the cluster's data. By restoring this snapshot, you can restore the cluster to a previous state. There are two ways to create snapshots: automatic and manual.

- Automatic snapshot creation: Snapshots are automatically created daily or weekly based on a preset time schedule, ensuring ongoing data protection. By configuring an automatic snapshot policy, you reduce manual operations while improving backup reliability and efficiency.
- Manual snapshot creation: You create snapshots manually for special occasions, for example, prior to a high-risk operation (such as a cluster

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

upgrade). This ensures you can restore data using snapshots in case anything should go wrong. Manual snapshots provide additional flexibility.

Impact on Billing

When you use snapshots for data backup, the generated snapshots are stored in OBS buckets, which will result in additional costs. For details, see **Object Storage Service Billing**section "Billing" in *Object Storage Service User Guide*.

Constraints

- Creating cluster snapshots will increase CPU load and disk I/O. To minimize impact on system performance, you are advised to schedule it to happen during off-peak hours.
- When creating or restoring a snapshot, you can only perform certain operations, including accessing Kibana, viewing metrics, and deleting other snapshots. However, you cannot perform the following operations: modifying configuration, restarting or deleting the cluster, deleting the snapshot that is being created or restored, and creating or restoring another snapshot. While a snapshot is being created or restored for a cluster, any automatic snapshot creation task initiated for that cluster will be canceled.
- When you create the first snapshot in a cluster, a full backup is performed on all data. Subsequent snapshots are all incremental, storing only the changes since the previous snapshot. Because each incremental snapshot relies on its predecessors, you must restore them in the correct sequence.
- In a cluster where storage and compute are decoupled, an index that is both frozen and closed cannot be backed up.
- For an index with read/write splitting between leader and follower enabled, synchronization cannot be stopped while a snapshot is being created for it. To stop synchronization, you must wait until snapshot creation is complete.
- An index that is being backed up via snapshot cannot be deleted or closed. This constraint also applies to deletion or closing operations triggered by ISM policies. To perform such operations, wait until snapshot creation is complete.

Prerequisites

- The OBS bucket used for storing cluster snapshots has been created. The OBS bucket must be in the same region as the target Elasticsearch cluster, and the storage class must be **Standard** or **Warm**.
- The cluster status is **Available**.

Enabling Cluster Snapshots

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Cluster Snapshots** tab, and check whether cluster snapshots are enabled.
   - If they are enabled, skip this step.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- If they are disabled, go to the next step.

5. On the **Cluster Snapshots** tab, click **Enable Snapshot**. In the displayed dialog box, configure the necessary settings.

Figure 2-92 Enabling cluster snapshots

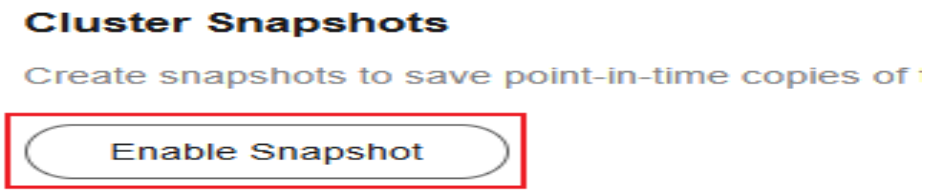


Table 2-229 Enabling cluster snapshots

Parameter	Description
OBS Bucket	From the drop-down list, select an OBS bucket for storing snapshots. If no OBS buckets meet your requirements, click **Create Bucket** to go to the OBS console and create one. For details, see **Creating a Bucket**. An IAM user must be granted the following OBS permissions to view available OBS buckets: obs:bucket:getBucketStoragePolicy obs:bucket:listAllMyBuckets
Backup Path	Snapshot storage path in the OBS bucket. The backup path cannot: • Contain the following characters: **\:*?"<>|'{}** • Start with a slash (/). • Start or end with a period (.). • Contain more than two consecutive slashes (/) or periods (.). • Exceed 512 characters. **CAUTION** Only standard OBS storage can be used to store snapshots. Do not apply OBS lifecycle rules to the backup path.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Maximum Backup Rate (per Second)	The parameter sets the maximum backup rate per node. When it is exceeded, flow control is triggered to prevent excessive resource usage and ensure system stability. **Value format**: number (0–9999) + unit (KB, MB, GB, TB, PB, or B) **Default value**: 40 MB The value 0MB means there is no limit on how fast data is backed up to snapshots. An overly high backup rate may lead to excessive resource usage, which may impact cluster stability. Therefore, set this parameter carefully. The actual backup rate may not reach the configured value, as it depends on factors such as OBS performance and disk I/O.

Maximum Recovery Rate (per Second)	The parameter sets the maximum recovery rate per node. When it is exceeded, flow control is triggered to prevent excessive resource usage and ensure system stability.
	Value format: number (0–9999) + unit (KB, MB, GB, TB, PB, or B)
	Default value: 40 MB for version 7.6.2 or earlier; and 0 MB (no limit) for versions later than 7.6.2.
	An overly high recovery rate may lead to excessive resource usage, which may impact cluster stability. Configure this parameter carefully to maintain optimal performance.
	For versions later than 7.6.2, the actual recovery rate is also affected by the value of **indices.recovery.max_bytes_per_sec**. Whichever is smaller between the two parameters will be used. Additionally, the actual recovery rate may not reach the configured value, as it depends on factors such as OBS performance and disk I/O.
	NOTE
	• To check the value of **indices.recovery.max_bytes_per_sec**, run the following command:
	`GET _cluster/settings`
	• To modify **indices.recovery.max_bytes_per_sec**, run the following command:
	`PUT _cluster/settings`
	`{`
	`  "transient": {`
	`  "indices.recovery.max_bytes_per_sec": "100mb"`
	`  }`
	`}`

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
IAM Agency	Select an IAM agency to grant the current account the permission to access and use OBS. To store snapshots to an OBS bucket, you must have the required OBS access permissions. • If you are configuring an agency for the first time, click **Automatically Create IAM Agency** to create **css-obs-agency**. • If there is an IAM agency automatically created earlier, you can click **One-click authorization** to have the OBS Administrator permissions deleted automatically, and have the following custom policies added automatically instead to implement more refined permissions control. "obs:bucket:GetBucketLocation", "obs:object:GetObjectVersion", "obs:object:GetObject", "obs:object:DeleteObject", "obs:bucket:HeadBucket", "obs:bucket:GetBucketStoragePolicy", "obs:object:DeleteObjectVersion", "obs:bucket:ListBucketVersions", "obs:bucket:ListBucket", "obs:object:PutObject" • When OBS buckets use SSE-KMS encryption, the IAM agency must be granted KMS permissions. You can click **Automatically Create IAM Agency** and **One-click authorization** to have the following custom policies created automatically. "kms:cmk:create", "kms:dek:create", "kms:cmk:get", "kms:dek:decrypt", "kms:cmk:list" • To use **Automatically Create IAM Agency** and **One-click authorization**, the following minimum permissions are required: "iam:agencies:listAgencies", "iam:roles:listRoles", "iam:agencies:getAgency", "iam:agencies:createAgency", "iam:permissions:listRolesForAgency", "iam:permissions:grantRoleToAgency", "iam:permissions:listRolesForAgencyOnProject", "iam:permissions:revokeRoleFromAgency", "iam:roles:createRole" • To use an IAM agency, the following minimum permissions are required: "iam:agencies:listAgencies", "iam:agencies:getAgency", "iam:permissions:listRolesForAgencyOnProject", "iam:permissions:listRolesForAgency" **WARNING** The agency name can contain only letters (case-sensitive), digits, underscores (_), and hyphens (-). Otherwise, the backup will fail.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Determine whether to enable automatic snapshot creation based on service needs.

- To enable it, select **Automatic Snapshot Creation**, and configure necessary settings by referring to **Configuring Automatic Snapshot Creation**.
- To disable it, unselect **Automatic Snapshot Creation**, in which case, you will need to manually create snapshots.

6. Click **OK** to enable cluster snapshots.

   The configuration information will be displayed on the **Cluster Snapshots** tab.

Configuring Automatic Snapshot Creation

You can enable automatic snapshot creation after cluster snapshots are already enabled.

1. On the **Cluster Snapshots** tab, click **Modify Settings**. In the displayed dialog box, select **Automatic Snapshot Creation**, and configure necessary settings.

Figure 2-93 Modifying Settings

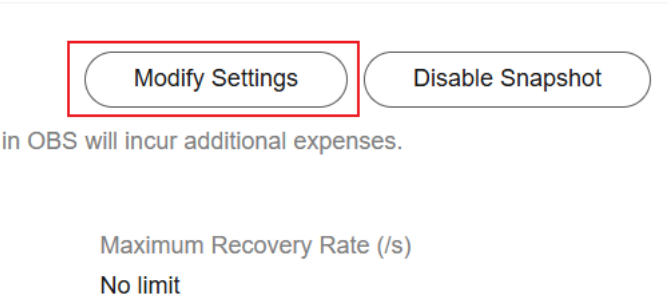


Table 2-230 Configuring automatic snapshot creation

Parameter	Description
Snapshot Name Prefix	A snapshot name consists of the snapshot name prefix and timestamp, for example, *snapshot-1566921603720*. The snapshot name prefix can contain 1 to 32 characters. It must start with a lowercase letter, and it can only contain lowercase letters, digits, hyphens (-), and underscores (_). The default value is **snapshot**.
Time Zone	Select a time zone for the backup start time.
Backup Start Time	Specify the start time of auto backup. Select a value from the drop-down list. The interval can be **Daily**, **Hourly**, or weekly (by selecting a specific day of the week), and the backup time can be set to any hour from 00:00 to 23:00 (full hours only).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Retained Snapshots	Number of automatic snapshots to be retained. The value ranges from 1 to 90. The system automatically deletes excess snapshots every half hour. (The expiration deletion policy applies only to the snapshots that were automatically created at the same frequency as the current automatic snapshot creation policy.) **NOTE** If the snapshot creation interval is short or if the data size of indexes is large, the number of automatic snapshots retained may not reach the value set using this parameter.
Index	Specify the name of the index to be backed up. • You can back up a specified index. To specify multiple indexes, use commas (,) to separate them, for example, **index1,index2,index3**. • You can use an asterisk (*) to match multiple indexes. For example, **index*** indicates that all indexes with the prefix **index** will be backed up. If you do not specify this parameter, all indexes in the cluster are backed up by default. The index name can contain 0 to 1024 characters. It cannot contain spaces, uppercase letters, or special characters **"\<|>/?** **NOTE** You can run the **GET /_cat/indices** command to query the names of all indexes in the cluster.

2. Click **OK** to enable automatic snapshot creation.

   After automatic snapshot creation is enabled, the **Automatic Snapshot Creation** parameter in the **Cluster Snapshots** area is shown as **Enabled**.

   Figure 2-94 Automatic snapshot creation enabled



   Snapshots automatically created according to the snapshot policy as well as those manually created are displayed in the cluster snapshot task list.

   When the snapshot task status changes to **Available**, the snapshot is created. If snapshot creation fails, click on the right of the snapshot list to view the number of failed tasks and learn the failure causes. A maximum of 20 failed

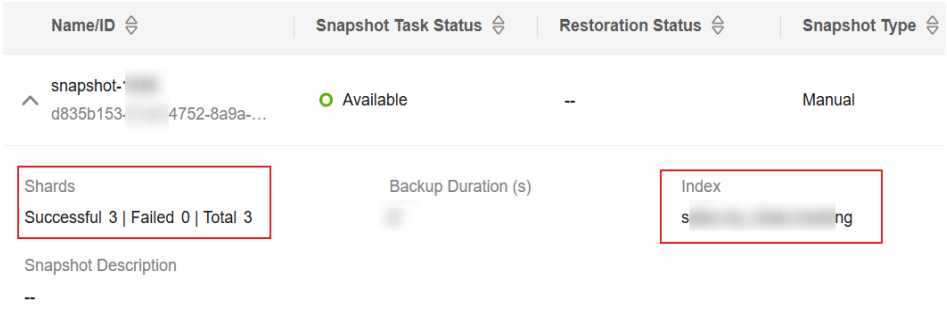
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

tasks can be displayed. When the snapshot function is disabled or the cluster is deleted, the failure records are also cleared.

3. Check whether data is successfully backed up.

   In the cluster snapshot task list, click ⌄ on the left of a snapshot name. The snapshot details are displayed. Check the shards and indexes that have been backed up to see if the backup is successful.

Figure 2-95 View Details



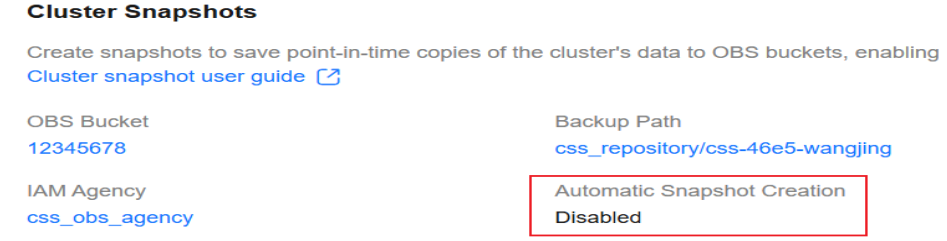
4. To disable automatic snapshot creation, click **Modify Settings**. In the displayed dialog box, deselect **Automatic Snapshot Creation**, and click **OK**. In the displayed dialog box, choose whether to delete snapshots created automatically. Then click **OK** to disable automatic snapshot creation.

CAUTION

- When disabling automatic snapshot creation, if you select **Delete snapshots created automatically**, only existing snapshots will be deleted. Snapshots that are being created will still be created.
- If automatic snapshot creation is disabled without deleting automatically created snapshots, all such snapshots in the cluster cannot be manually deleted later once automatic snapshot creation is enabled again. Instead, they can only be deleted by the system based on the snapshot lifecycle policy. For example, if **Retained Snapshots** is set to **10** in the policy, the system automatically deletes excess snapshots (oldest first) on the next half hour when the number of existing snapshots exceeds this limit.

After automatic snapshot creation is disabled, the **Automatic Snapshot Creation** parameter in the **Cluster Snapshots** area is shown as **Disabled**.

Figure 2-96 Disabling automatic snapshot creation



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Manually Creating a Snapshot

When snapshots are enabled for a cluster, you can manually create snapshots for it.

1. Under **Cluster Snapshot Tasks**, click **Manually Create Snapshot**. In the displayed dialog box, configure the snapshot policy.

   Figure 2-97 Manually creating a snapshot

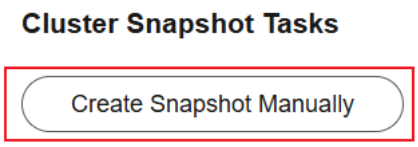
   

   Table 2-231 Parameters for manually creating a snapshot

Parameter	Description
Snapshot Name	Set the snapshot name. Unlike automatically created snapshots, manually created snapshots do not have time information automatically added to their names. The value can contain 4 to 64 characters and must start with a letter. Only lowercase letters, digits, hyphens (-), and underscores (_) are allowed.
Index	Specify the name of the index to be backed up. • You can back up a specified index. When specifying multiple indexes, use commas (,) to separate them, for example, **index1,index2,index3**. • You can use an asterisk (*) to match multiple indexes. For example, **index*** indicates that all indexes with the prefix **index** will be backed up. If you do not specify this parameter, all indexes in the cluster are backed up by default. The index name can contain 0 to 1024 characters. It cannot contain spaces, uppercase letters, or special characters **"\<|>/?** **NOTE** You can run the **GET /_cat/indices** command to query the names of all indexes in the cluster.
Snapshot Description	Add a snapshot description. The value contains 0 to 256 characters and cannot contain special characters **!<>=&\"'**

2. Click **OK** to start creating snapshots.

   Snapshots manually created as well as those automatically created according to the snapshot policy are displayed in the cluster snapshot task list.

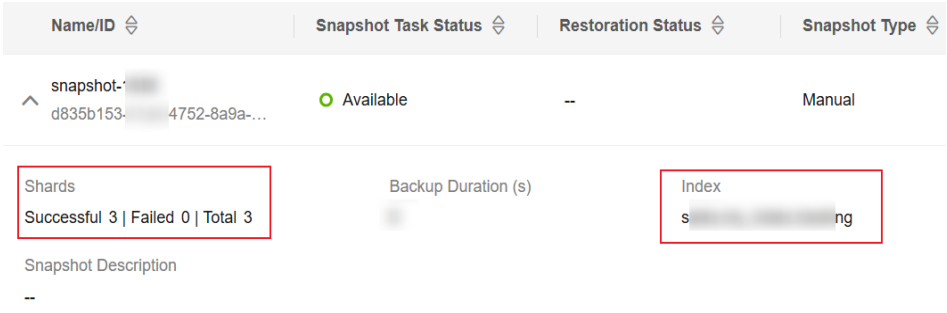
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

When the snapshot task status changes to **Available**, the snapshot is created. If snapshot creation fails, click on the right of the snapshot list to view the number of failed tasks and learn the failure causes. A maximum of 20 failed tasks can be displayed. When the snapshot function is disabled or the cluster is deleted, the failure records are also cleared.

3. Check whether data is successfully backed up.

   In the cluster snapshot task list, click ⌄ on the left of a snapshot name. The snapshot details are displayed. Check the shards and indexes that have been backed up to see if the backup is successful.

Figure 2-98 View Details

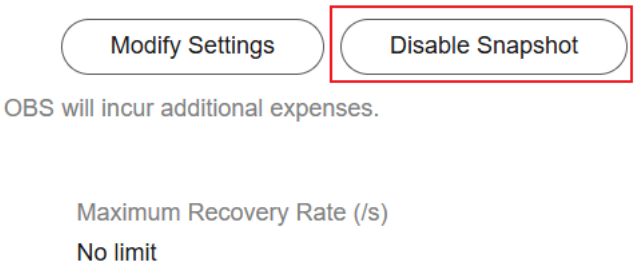


Disabling Cluster Snapshots

If you no longer need the cluster snapshot function, you can disable it to reclaim storage resources.

On the **Cluster Snapshots** page, click **Disable Snapshot**. In the displayed dialog box, click **OK**. Disabling cluster snapshots does not automatically delete existing snapshots. Instead, you need to manually delete them on the OBS console.

Figure 2-99 Disable Snapshot



Related Documents

By creating and restoring snapshots, you can migrate data between clusters. For details, see **Migrating Data Between Huawei Cloud Elasticsearch Clusters Using Backup and Restoration**.

2.17.2 Restoring Data Using Snapshots

During routine O&M, the ability to rapidly restore data upon data loss or corruption is crucial for Elasticsearch clusters. A snapshot allows you to restore a

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

point-in-time copy of a cluster to the original cluster or to another eligible cluster. This process fetches the backed-up index data, metadata, and shard structure from OBS and reconstruct them in the selected destination cluster, ensuring data consistency. This capability is essential for operations like data rollback, cluster migration, and disaster recovery. With snapshots for CSS clusters, you can restore data quickly and reliably to ensure service continuity.

Constraints

- Creating cluster snapshots will increase CPU load and disk I/O. To minimize impact on system performance, you are advised to schedule it to happen during off-peak hours.
- If the source cluster is unavailable, you can restore its snapshots to a designated destination cluster.
- When creating or restoring a snapshot, you can only perform certain operations, including accessing Kibana, viewing metrics, and deleting other snapshots. However, you cannot perform the following operations: modifying configuration, restarting or deleting the cluster, deleting the snapshot that is being created or restored, and creating or restoring another snapshot. While a snapshot is being created or restored for a cluster, any automatic snapshot creation task initiated for that cluster will be canceled.
- When you create the first snapshot in a cluster, a full backup is performed on all data. Subsequent snapshots are all incremental, storing only the changes since the previous snapshot. Because each incremental snapshot relies on its predecessors, you must restore them in the correct sequence.
- Before snapshot restoration is completed, the indexes being restored to the destination cluster cannot be queried.
- If the two clusters have indexes with the same names but different shard structures, such indexes cannot be restored.
- The version of the destination cluster must be the same as or later than that of the source cluster.

Prerequisites

In the source cluster's snapshot task list, there are snapshots whose **Snapshot Task Status** is **Available**. The status of the destination cluster is also **Available**.

Restoring Data

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Cluster Snapshots** tab.
5. In the cluster snapshot task list, select a snapshot, and click **Restore** in the **Operation** column. In the displayed dialog box, configure necessary settings.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 2-100 Restoring a snapshot

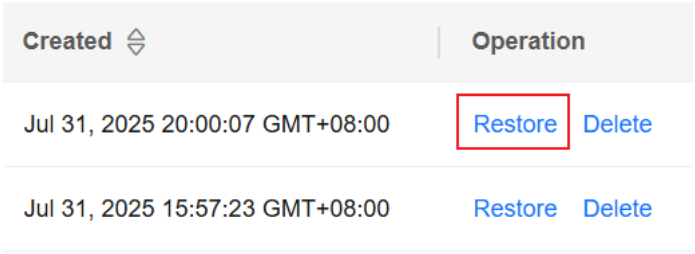


Table 2-232 Restoration settings

Parameter	Description
Index	Specify the name of the index you want to restore. • The value is a string of 0 to 1024 characters that cannot contain uppercase letters, spaces, or the following special characters: **"\<|>/?**. • When restoring an index whose name is prefixed with **.kibana**, the index name must be specified. • The **.opendistro_security** index cannot be restored. • You can use an asterisk (*) to match multiple indexes. For example, **index*** indicates that all indexes with the prefix **index** will be restored. When an asterisk (*) is used for index matching, the **.opendistro_security** index and any system indexes whose name is prefixed with **.kibana** are filtered out by default. • You can restore indexes by specifying their names, for example, **index1,index2,index3**. By default, this parameter is left blank. That is, no index name is specified, and all indexes will be restored.
Rename Pattern	Index name matching rule. Enter a regular expression. Indexes that match the regular expression will be restored. The **Rename Pattern** and **Rename Replacement** take effect only when they are both configured at the same time. For an example, see **Index, Rename Pattern, and Rename Replacement: Parameters and Examples**. The value is a string of 0 to 1024 characters that cannot contain uppercase letters, spaces, or the following special characters: **"\<|>/?,**. For example, **index_(.+)** indicates that all indexes whose name starts with **index_** will be renamed upon restoration.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Rename Replacement	Rule for index renaming. Upon restoration, matching indexes are renamed according to the defined rule. The **Rename Pattern** and **Rename Replacement** take effect only when they are both configured at the same time. For an example, see **Index, Rename Pattern, and Rename Replacement: Parameters and Examples**. The value is a string of 0 to 1024 characters that cannot contain uppercase letters, spaces, or the following special characters: **"\<|>/?,**. For example, **restored_index_$1** indicates that **restored_** will prefix the name of all restored indexes.

Cluster	Select the cluster where you want to restore the data to. You can select the current cluster or another.
	• You must specify a cluster whose status is **Available**. If the status of the current cluster is **Unavailable**, you cannot restore the snapshot to the current cluster.
	• When you choose to restore data to another cluster, make sure the target cluster has the same or later version than the current cluster, and that the two clusters are in the same region. After a frozen index is restored using a snapshot, the index is stored on the local node and is read-only by default. To make it writable, run the following command to configure the index:
	`PUT index_name/_settings`
	`{`
	`  "index.blocks.write": null`
	`}`

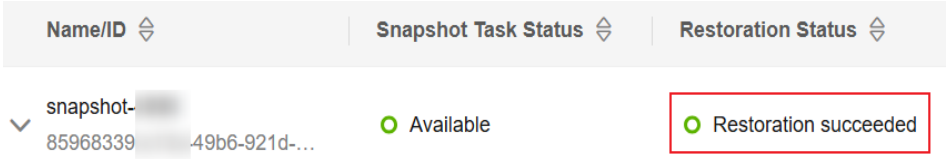
Overwrite same-name indexes in destination cluster	Whether to overwrite same-name indexes in the destination cluster. By default, this option is unselected. Restoring a snapshot will overwrite existing data in the destination cluster. When there are same-name indexes in the destination cluster, you need to select this option in order to restore same-name, same-shard structure indexes. Indexes with a different shard structure cannot be restored.

6. Click **OK** to start restoring the snapshot.

   If restoration succeeds, **Snapshot Task Status** of the snapshot in the snapshot task list will change to **Restoration succeeded**, and the index data captured in the snapshot is reconstructed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 2-101 Restoration succeeded



Verifying Snapshot Restoration

After snapshot restoration, go to the destination cluster to check the restored index information.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.
5. Run the following command to check index information.

```
GET _cat/indices
```

Index, Rename Pattern, and Rename Replacement: Parameters and Examples

This section describes the **Index**, **Rename Pattern**, and **Rename Replacement** parameters. It also provides examples to show how to configure them.

- **Retaining the original index name when restoring an index**

  Specify **Index** for the index to be replaced.
- **Renaming an index when restoring it**

  To rename an index upon restoring it, specify **Index**, **Rename Pattern**, and **Rename Replacement**. Their meanings are as follows:

  - **Index**: Specifies the name or pattern (wildcard supported) of the indexes you wish to restore. It must include the indexes that need to be renamed. This parameter defines the indexes that you wish to restore using snapshots to a designated destination cluster.
  - **Rename Pattern**: Uses a regular expression to specify indexes that you wish to rename among the indexes to be restored (specified by **Index**). Only indexes whose names match the regular expression exactly will be renamed.
  - **Rename Replacement**: Defines the naming rule for restored indexes that match **Rename Pattern**. This rule is a character string that uses a capture group (for example, $1) to reuse the text captured by **Rename Pattern**.

Example:

- **Index** is set to **index_*, old_index**: **index_*** indicates all indexes whose name starts with **index_**, and **old_index** specifies a single index.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- **Rename Pattern** is set to **index_(.+)**: This specifies that indexes whose name starts with **index_** will be renamed.
- **Rename Replacement** is set to **restored_index_$1**: This replaces the matched index names with **restored_index_** plus the text of the capture group.

The corresponding API request is as follows:

```
POST /_snapshot/{repo_name}/{snapshot_name}/_restore
{
  "indices": "index_*, old_index",
  "ignore_unavailable": true,
  "include_global_state": true,
  "rename_pattern": "index_(.+)",   // This corresponds to the Rename Pattern parameter.
  "rename_replacement": "restored_index_$1",  // This corresponds to the Rename Replacement parameter.
}
```

Restoration results for different indexes:

- The source index **index_logs** is renamed as **restored_index_logs**.
- The source index **index_users** is renamed as **restored_index_users**.
- The source index **old_index** does not match rename_pattern, and is therefore not renamed. The original name **old_index** is retained after restoration.

Related Documents

By creating and restoring snapshots, you can migrate data between clusters. For details, see **Migrating Data Between Huawei Cloud Elasticsearch Clusters Using Backup and Restoration**.

2.17.3 Deleting a Cluster Snapshot

While snapshots offer an efficient way to back up Elasticsearch clusters, retaining unnecessary snapshots leads to inflated storage costs. To free up space and manage expenses, it is essential to delete them in a timely manner.

Constraints

- Deleting a snapshot will permanently delete its data.
- The deletion of automatically generated snapshots is subject to the following constraints:
  - While automatic snapshot creation is enabled, the system automatically deletes snapshots on the nearest half hour after their retention period expires.
  - After automatic snapshot creation is disabled, existing auto-created snapshots will not be deleted automatically, but they can be deleted manually.
  - If automatic snapshot creation is re-enabled, the system resumes automatic deletion of all auto-created snapshots upon expiration, including these created before automatic snapshot creation was previously disabled.

Manually Deleting a Snapshot

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Cluster Snapshots** tab.
5. In the cluster snapshot task list, select the snapshot you want to delete, and click **Delete** in the **Operation** column. In the displayed dialog box, enter **DELETE** and click **OK**.

   A deleted snapshot will not be displayed in the snapshot list.

2.18 Managing Clusters

2.18.1 Viewing Cluster Details

On the CSS management console, you can check an Elasticsearch cluster's status, version, nodes, configurations and settings, and resource allocation on both the cluster list and the cluster overview page.

Introduction to the Cluster List

A cluster list displays all clusters of the same type under the current account. If there are a large number of clusters, the cluster list will consist of multiple pages, showing clusters of all states.

Clusters are listed in chronological order by default in the cluster list, with the most recent clusters displayed at the top.

You can click **Export** to download the full cluster list.

Table 2-233 Cluster list parameters

Parameter	Description
Name/ID	Name and ID of a cluster. The cluster ID is automatically generated by the system and uniquely identifies a cluster.
Cluster Status	Current status of a cluster. • Available: The cluster is running properly. • Creating: The cluster is being created. • Processing: The cluster is being restarted, scaled out, backed up, or restored. • Abnormal: The cluster failed to be created or is unavailable. **WARNING** If a cluster is unavailable, you can delete the cluster or use a snapshot created when the cluster was available to restore its data to another cluster. However, you cannot scale out this cluster, access it through Kibana, create snapshots for it, or use an existing snapshot to restore its data to itself. Do not try to import data into a cluster that is unavailable, as the data may get lost. For such a cluster, you can check its monitoring data or restart it to try to fix the problem. If the fault persists, contact technical support.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Task Status	Status of a task, such as cluster restart, scale-out, or scale-in.
Version	Cluster version and image version. For example, if the version is **7.10.2(7.10.2_x.x.x_x.x.x)**, **7.10.2** is the cluster version and **7.10.2_x.x.x_x.x.x** enclosed in parentheses is the image version. **CAUTION** • For a cluster created a long time ago, its image version may not be available because it is likely not recorded in the database. • If **EOS** is displayed below the version number, the current version has reached EOS and hence poses certain service availability risks. You are advised to upgrade the cluster version as soon as possible. For details, see **Upgrading the Cluster Version**.
Private Network Address	Private network address and port number of a cluster. You can use it to access the cluster. If the cluster has multiple nodes, the private network addresses and port numbers of all nodes are displayed.
VPC Endpoint Address	IP address or private domain name of the VPC endpoint associated with the cluster.
Security Mode	Whether the security mode is enabled for the cluster. Only authorized users can access a security-mode cluster.
Enterprise Project	Enterprise project that a cluster belongs to.
Tags	Displays cluster tags. You can click **Edit Tag** to add, delete, or modify cluster tags.
Cluster Description	Provides supplementary information about the cluster.
Created	Cluster creation time.
Billing Mode	Billing mode of a cluster.
Operation	Operations that can be performed on a cluster, such as checking metrics, restart, and deletion. If an operation is unavailable, the menu button is grayed out.

Viewing Basic Cluster Information

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, check the cluster's basic information, node configuration, and network information.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-234 Basic cluster information

Parameter	Description
Cluster ID	Unique ID of a cluster, which is automatically generated by the system. Each cluster ID is unique within the same region.
Cluster Name	Cluster name. You can click to change it. **Initial Cluster Name** cannot be changed.
Initial Cluster Name	This name is unique and cannot be changed. It is only used in the cluster configuration file in YAML.
Cluster Version	Cluster version. For example, if the version is **7.10.2(7.10.2_x.x.x_x.x.x)**, **7.10.2** is the cluster version and **7.10.2_x.x.x_x.x.x** enclosed in parentheses is the image version. **CAUTION** • For a cluster created a long time ago, its image version may not be available because it is likely not recorded in the database. • If **EOS** is displayed below the version number, the current version has reached EOS and hence poses certain service availability risks. You are advised to upgrade the cluster version as soon as possible. For details, see **Upgrading the Cluster Version**.
Cluster Status	Current cluster status.
Task Status	Current task status of a cluster. If there is no ongoing task, **--** is displayed.
Region | AZ	The region and AZ where the cluster is located.
Enterprise Project	Enterprise project that the cluster belongs to. You can click a project name to check the basic project information.
Cluster Description	Provides supplementary information about the cluster. You can click to modify the description.

Table 2-235 Node information

Parameter	Description
Data Node Specifications	The specifications of data nodes used in the cluster.

Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Data Nodes	The number of data nodes in the cluster.
Data Node Storage Type	The storage type of data nodes in the cluster.
Data Node Capacity (GB)	Storage capacity per data node in the cluster.
Master Node Specifications	The specifications of master nodes used in the cluster. This parameter is displayed only when dedicated master nodes are used in the cluster.
Master Nodes	The number of master nodes in the cluster. This parameter is displayed only when dedicated master nodes are used in the cluster.
Master Node Storage Type	The storage type of master nodes in the cluster. This parameter is displayed only when dedicated master nodes are used in the cluster.
Master Node Capacity (GB)	Storage capacity per master node in the cluster. This parameter is displayed only when dedicated master nodes are used in the cluster.
Client Node Specifications	The specifications of client nodes used in the cluster. This parameter is displayed only when dedicated client nodes are used in the cluster.
Client Nodes	The number of client nodes in the cluster. This parameter is displayed only when dedicated client nodes are used in the cluster.
Client Node Storage Type	The storage type of client nodes in the cluster. This parameter is displayed only when dedicated client nodes are used in the cluster.
Client Node Capacity (GB)	Storage capacity per client node in the cluster. This parameter is displayed only when dedicated client nodes are used in the cluster.
Cold Data Node Specifications	The specifications of cold data nodes used in the cluster. This parameter is displayed only when cold data nodes are used in the cluster.
Cold Data Nodes	The number of cold data nodes in the cluster. This parameter is displayed only when cold data nodes are used in the cluster.
Cold Data Node Storage Type	The storage type of cold data nodes in the cluster. This parameter is displayed only when cold data nodes are used in the cluster.
Cold Data Node Capacity (GB)	Storage capacity per cold data node in the cluster. This parameter is displayed only when cold data nodes are used in the cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Cluster Storage Capacity (GB)	Total storage capacity of the cluster. • Quota: the upper limit of the storage capacity allocated to the cluster. • Used: storage space already used by the cluster. The cluster's disk usage can be calculated as follows: Disk usage = (Used/Quota) x 100%
Node Details	View the details of all cluster nodes. Click **View**. In the displayed dialog box, view node details, including the node name/ID, status, and IP address.

Table 2-236 Network information

Parameter	Description
VPC	The VPC to which a cluster belongs.
Current Subnet	The subnet to which a cluster belongs. If the current subnet cannot meet your service requirements, click **Switch Subnet**. In the displayed dialog box, select another subnet. If no existing subnets meet your requirements, click **Create Subnet** to create a new subnet. **CAUTION** • After switching the subnet, the new subnet is used for adding nodes, including scaling, adding dedicated master/client nodes, and enabling VPC Endpoint. IP addresses will be allocated to newly added nodes from the new subnet. • A cluster can be associated with a maximum of two subnets. • Subnet switching does not affect the network settings of existing nodes.
Used Subnet	Subnet once used by the cluster.
Security Group	Security group to which a cluster belongs. To modify the security group of a cluster, click **Change Security Group** on the right. **WARNING** The selected security group must allow port 9200 in the inbound direction. Otherwise, the cluster may be inaccessible to external services. Furthermore, to enable Kibana private network access, port 5601 must also be allowed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Security Mode	Security mode of a cluster. • Enabled: The security mode is enabled for the current cluster. • Disabled: The security mode is disabled for the current cluster. You can click to change the cluster's security mode. For details, see **Changing the Security Mode**.
Reset Password	Change the **admin** password for a security-mode cluster. Click **Reset** to change the password. Requirements for admin passwords: • The password can contain 8 to 32 characters. • The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. The following special characters are allowed: **~!@#$%^&*()-_=+|[{}];:,<.>/?** • Do not use the administrator name, or the administrator name spelled backwards, as the password. • Change the password periodically. This parameter is displayed only when **Security Mode** is **Enabled**.
HTTPS Access	Whether to enable HTTPS for a cluster. • Disabled: The cluster uses HTTP. For a cluster with the security mode disabled, HTTPS is disabled by default. • Enabled: The cluster uses HTTPS. Only security-mode clusters support HTTPS. If **HTTPS Access** is enabled, you can click **Download Certificate** to obtain the certificate used for access control. Currently, the security certificate cannot be used in a public network environment. For details about how to switch the network protocol for a security-mode cluster, see **Switching between HTTP and HTTPS in Security Mode**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Public Network Access	Public network access information of the cluster. • For a security-mode cluster with public network access enabled, the configured public IP address is displayed. • For a security-mode cluster with public network access disabled, **--** is displayed. When public network access is enabled for a cluster, you are advised to use a whitelist for access control. This parameter is displayed only when **Security Mode** is **Enabled**.
Bandwidth	Public network bandwidth of the cluster. Click **Edit** to change the bandwidth size. This parameter is displayed only when the security mode and public network access are both enabled for the cluster.
Public Network Access Control	You can configure public network access control for a security-mode cluster by IP addresses or IP address ranges. Click **Modify** to modify the access control whitelist. • If a whitelist is configured, only IP addresses that are on this whitelist can access the cluster over the public network. Click **+Add**. In the displayed text box, enter IP addresses or CIDR blocks that are allowed to access the cluster from the public network. Separate them using commas (,). Each value must be unique. An example of valid values: **192.168.1.1,10.0.0.0/24**. Examples of invalid values: **0.0.0.0**, **xx.xx.xx.xx/0**, **172.16.0.0-172.16.255.255**, non-standard formats (e.g., **192.168.1**), and duplicate values. • If no whitelist is configured, all public IP addresses can access the cluster. However, this can be a security risk and should be avoided. This parameter is displayed only when the security mode and public network access are both enabled for the cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Private IPv4 Address	Private IPv4 address and port number of a cluster. If the cluster has only one node, the IP address and port number of the only node are displayed, for example, **10.62.179.32:9200**. If the cluster has multiple nodes, the IP addresses and port numbers of all nodes are displayed, for example, **10.62.179.32:9200,10.62.179.33:9200**.

Table 2-237 Additional information

Parameter	Description
Billing Mode	Billing mode of a cluster.
Created	When a cluster was successfully created.

2.18.2 Checking Default Cluster Plugins

Some of the CSS enhancements over open-source Elasticsearch are delivered as default cluster plugins. These plugins are automatically injected to each cluster node during cluster creation and are loaded when the JVM process starts. CSS allows you to check these plugins through the web console or via a standard RESTful API.

Constraints

The default cluster plugins cannot be manually uninstalled.

Checking the Plugins on the CSS Console

Quickly check all default plugins supported by the current cluster without writing any code.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Cluster Settings** > **Plugins**.
5. On the **Default Plugins** tab, check the default plugins supported by the current cluster.

Checking the Plugins via an API

Check the loading of default plugins on specific cluster nodes through an API, for example, when you need to troubleshoot the tokenizer.

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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
4. In the left navigation pane on Kibana, choose **Dev Tools**.
5. Run the following command to check the cluster's plugins:

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

Example response:

```
name                 component                    version
css-test-ess-esn-1-1 analysis-dynamic-synonym     1.0.1
css-test-ess-esn-1-1 analysis-icu                 1.1.6
css-test-ess-esn-1-1 analysis-ik                  1.0.1
......
```

Table 2-238 Response parameters

Parameter	Description
name	Node name. Use this field to determine whether each plugin has been loaded on all nodes.
component	Unique plugin ID. This name is required when you configure mappings or the tokenizer.
version	Plugin version.

2.18.3 Modifying Cluster Parameters

When using an Elasticsearch cluster, you may need to adjust cluster behavior based on service requirements. For example, frontend applications may need to access cluster APIs across domains, query performance may need to be optimized by adjusting the cache, or the Kibana session timeout may need to be extended. CSS allows you to modify configuration files **elasticsearch.yml**, **kibana.yml** and **system.config** online to flexibly adjust core cluster parameters and visualization platform parameters. You can use this feature for scenarios such as configuring cross-domain access, optimizing query performance, and customizing Kibana behavior.

Configuration File Description

A CSS cluster relies on the following configuration files. Each file has a different purpose and modification method, and changes to one file do not affect the others. You must modify them separately.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-239 Configuration files

Configuration File	Configuration Range	When Changes Take Effect
elasticsearch.yml	Core cluster configuration parameters, including cross-origin access, index reindexing, query caching, and thread pools.	After modification, all nodes load the new configuration when the cluster is manually restarted.
kibana.yml	Independent configuration parameters for Kibana, the visualization platform, such as session timeout, default redirect page, and user interface display.	After modification, only Kibana loads the new configuration when it is restarted. Cluster nodes are not affected. The changes apply only when Kibana is accessed through a public or private network address. They do not apply if Kibana is accessed in one click from the console.
system.config	Metadata file for the SAML identity provider (IdP). For details, see **SAML - Open Distro for Elasticsearch Documentation**.	After configuration, a **descriptor.xml** file is generated in the Elasticsearch config directory. All nodes load this file when the cluster is manually restarted.

How to apply the changes:

1. Modify parameters: Edit the parameter values for the required modules on the parameter configuration page.
2. Save changes: CSS writes the changes to the corresponding configuration file.
3. Restart the cluster or process: Nodes or components reload the new configuration file.
4. Apply the changes: The new configuration takes effect after the restart is complete.

Constraints

- **system.config** can be modified only for Elasticsearch 7.10.2.
- Exercise caution when configuring custom parameters. Make sure custom parameters are supported by the corresponding component. Otherwise, the cluster or the corresponding component may fail to restart.
- After modifying parameters in **elasticsearch.yml** and **system.config**, you must manually restart the cluster for the changes to take effect.
- A maximum of 20 modification records are displayed.
- You are advised to modify parameters and restart clusters during off-peak hours to minimize impact on services.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Modifying Elasticsearch Configurations (elasticsearch.yml)

Modify core Elasticsearch cluster parameters, including cross-domain access, index reindexing, query caching, and thread pools.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Cluster Settings** > **Parameter Settings** > **Elasticsearch**.
5. Click **Edit** to modify module parameters as required.

Table 2-240 Module parameters

Module Name	Parameter	Description
Cross-domain Access	http.cors.allow-credentials	Whether to carry authentication information in cross-domain requests, that is, whether to contain the Access-Control-Allow-Credentials field in the response header. The value can be: • **true**: The response header contains the Access-Control-Allow-Credentials field. • **false** (default value): The response header does not contain the Access-Control-Allow-Credentials field.
	http.cors.allow-origin	Source IP addresses or domain names allowed for cross-domain access. When there are multiple values, separate them with commas (,). Example: **192.168.122.122:9200** or **192.168.1.1:9200, 192.168.1.2:9200**
	http.cors.allow-headers	Request header fields for cross-domain access. When there are multiple values, separate them using a comma (,) and a space. Example: **X-Requested-With, Content-Type, Content-Length** The value can contain a maximum of 1024 characters. It can only contain letters, digits, hyphens (-), underscores (_), and slashes (/).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Module Name	Parameter	Description
	http.cors.max-age	Maximum retention duration of pre-check responses in the browser cache. Unit: s Value range: 0 to 31,536,000 (approximately 1 year) Default value: 1,728,000 (20 days)
	http.cors.enabled	Whether to enable cross-domain access. The value can be: • **true**: Enable cross-domain access. • **false** (default): Disable cross-domain access.
	http.cors.allow-methods	HTTP methods allowed for cross-domain access. When there are multiple values, separate them using a comma (,) and a space. Example: **OPTIONS, GET, POST** Value range: OPTIONS, HEAD, GET, POST, PUT, and DELETE
Reindexing	reindex.remote.whitelist	Which remote hosts are allowed for remote reindexing operations. Use commas (,) to separate multiple values. Example: **192.168.122.122:9200** or **192.168.1.1:9200, 192.168.1.2:9200**
Custom Cache	indices.queries.cache.size	Maximum heap space allocated to the query cache. Unit: percentage (%) Value range: 1 to 100 Default value: 10%
Queue Size in a Thread Pool	thread_pool.force_merge.size	Size of the thread pool used for force merge operations. Value range: a positive integer Default value: 1
Custom	-	Click **Add** to add or modify custom parameters supported by Elasticsearch. **WARNING** Set the values of any custom parameters to those supported by Elasticsearch. Otherwise, the cluster may fail to restart. Exercise caution when performing this operation.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

6. After modifying the parameters, click **Save**. In the displayed dialog box, confirm the parameters and select the restart option, and then click **OK**.

   If **Job Status** is **Succeeded** in the parameter settings task list below, the change has been saved.
7. Click **Restart** in the upper right corner to restart the cluster and apply the change.
8. After restart, check that **Cluster Status** is **Available** in the cluster list. This state indicates the parameter changes have taken effect.

Modifying Kibana Configuration (kibana.yml)

Modify Kibana visualization platform parameters, such as the session timeout, default redirect page, and user interface display language.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Cluster Settings** > **Parameter Settings** > **Kibana**.
5. Click **Edit** to modify module parameters as required.

Table 2-241 Module parameters

Module Name	Parameter	Description
Base Path	server.basePath	Custom value for Base Path. When you need to access Kibana through a reverse proxy (such as Nginx), this value is appended to the proxy address as a subpath. For example, if the reverse proxy address is **http://example.com** and **server.basePath** is set to **/test**, you can access Kibana through the address **http://example.com/test**. The value can be: • Must start with **/**. • Must not contain consecutive slashes (for example, **//**). • Can only contain letters, digits, hyphens (-), underscores (_), periods (.), and slashes (/). • Can contain multi-level paths, but its length cannot exceed 254 characters. • Must not end with a slash (/).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Module Name	Parameter	Description
	server.rewriteBasePath	Whether Kibana or a reverse proxy will remove the server.basePath prefix from the request path. This parameter is optional. If no reverse proxy is used, ignore this parameter. The value can be: • **true**: Kibana will automatically remove the server.basePath prefix from the request path. • **false**: The reverse proxy (such as Nginx) will remove the server.basePath prefix from the request path. In this case, you can only access Kibana through the reverse proxy. The default value is **false**. The default value can only be changed for Elasticsearch clusters no earlier than 6.5.4.
Custom	-	Click **Add** to add or modify custom parameters supported by Kibana. **WARNING** Set the values of custom parameters to those supported by Kibana. Otherwise, Kibana may fail to restart. Proceed with caution.

6. After modifying the parameters, click **Save**. In the displayed dialog box, confirm the parameters and select the restart option, and then click **OK**. Modifying Kibana settings will restart Kibana, making it temporarily unavailable.

   If **Job Status** is **Succeeded** in the parameter settings task list below, the parameter changes have taken effect.

Modifying System Configuration (system.config)

CAUTION

system.config can be modified only for Elasticsearch 7.10.2.

Modify system configuration parameters. A **descriptor.xml** file is generated or modified in the same directory as the **elasticsearch.yml** configuration file.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

4. Choose **Cluster Settings** > **Parameter Settings** > **System**.
5. Click **Edit** and enter the content of the **descriptor.xml** file in the **System** area.

   The file size cannot exceed 100 KB (102,400 bytes or characters, including line breaks).
6. After modifying the file, click **Save**. In the displayed dialog box, confirm the parameters and select the restart option, and then click **OK**.

   If **Job Status** is **Succeeded** in the parameter settings task list below, the change has been saved.
7. Click **Restart** in the upper right corner to restart the cluster and apply the change.
8. After restart, check that **Cluster Status** is **Available** in the cluster list. This state indicates the new **descriptor.xml** file has taken effect.

Related Documents

- After configuration changes, you must restart the cluster to apply the changes. For details, see **Restarting a Cluster**.
- For more parameters supported by Elasticsearch, see **Elasticsearch official documentation**.
- For more parameters supported by Kibana, see **Kibana official documentation**.

FAQ

- **Q1: What should I do if the cluster status shows Configuration error after modifying parameters?**

  Configuration error indicates that the modified configuration file contains invalid parameter values. Check the specific error message in the parameter configuration task list, correct the parameter values as prompted, save the changes, and then restart the cluster again. If the error is caused by a custom parameter, verify whether the parameter is officially supported by the corresponding component.
- **Q2: What will happen if I forget to restart the cluster after modifying parameters?**
  - If you do not restart the cluster after modifying parameters in **elasticsearch.yml** and **system.config**, the changes do not take effect and the cluster continues to run with the old configuration. **Task Status** in the cluster list shows **Configuration Not Updated**. You are advised to restart the cluster immediately after modifying parameters.
  - Modifying parameters in **kibana.yml** automatically restarts the Kibana process, so there is no need to manually restart the cluster. The new configuration takes effect immediately after the Kibana process is restarted.

2.18.4 Managing Cluster Tags

As business grows, an enterprise may operate hundreds of Elasticsearch clusters, and cluster names alone will be insufficient for categorization and identification. For example, tasks such as resource monitoring or automated operations via scripts require the ability to quickly locate clusters based on specific attributes. On

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

CSS, you can use tags (key-value pairs) to label resources, enabling efficient filtering and batch management. When used together with Tag Management Service (TMS), tags help you implement enterprise-grade resource governance.

Constraints

- Each cluster can have a maximum of 20 tags.
- If your organization has configured tag policies for CSS, add tags to clusters based on these policies. If a tag does not comply with existing tag policies, cluster creation may fail. Contact the administrator to learn more about these policies.
- To use the same tag to identify multiple cloud resources for better resource grouping, predefine tags in Tag Management Service (TMS). For details, see **Predefined Tags Overview**.

Managing Cluster Tags in Batches

Manage tags for multiple clusters in batches, including add, modify, and delete. This feature is useful in cases such as environment refactoring and large-scale migrations.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, select one or more clusters.
4. Click **Edit Tag** above the cluster list.
   - Add

     In the displayed dialog box, click **+Add**, and add new tags in the form of key-value pairs. The tags will be added for all selected clusters.
   - Modify

     In the displayed dialog box, modify existing tag values. The changes will be applied to all selected clusters, overwriting existing tags. Avoid this operation if you have custom tags that you wish to keep.
   - Delete

     In the displayed dialog box, click **Delete** next to the tag you want to delete. This deletes the tag for all selected clusters. The deletion cannot be undone.
5. Click **OK** to save the changes.
6. Verify the result: In the cluster list, hover the mouse over the icon in the **Tag** column for each cluster. Check that the tag information has been updated for all selected clusters.

Managing Tags for a Single Cluster

Add, modify, or delete tags for a single cluster.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

4. Click the **Tags** tab.
5. Click **Edit Tag**. In the displayed dialog box, modify the cluster's tags.
   - Add

     In the displayed dialog box, click **+Add**, and add new tags in the form of key-value pairs.
   - Modify

     In the displayed dialog box, modify existing tag keys and values.
   - Delete

     In the displayed dialog box, click **Delete** next to the tag you want to delete.
6. Click **OK**.
7. Verify the result: In the tag list, check that the tag keys and values have been updated.

Searching for Clusters by Tag

Locate clusters by tag from among a large number of resources.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. Click the search box above the cluster list. Select a tag key, operator, and tag value to search.
   - You can select a tag key and value from the drop-down list. The system returns a list of clusters that exactly match the tag key and value.
   - In case you use multiple tags to start a search: With the same tag key but different values, the system returns a list of clusters that match any of the tag values; If the tag keys are different, the system returns a list of clusters that match all of these keys and tags.

   The cluster list is filtered in real time to display matching clusters only.

2.18.5 Associating a Cluster with an Enterprise Project

In practice, it is common for multiple departments (such as R&D and quality management) within a large organization to share a single Huawei Cloud account. Enterprise projects let you manage Elasticsearch clusters by department or project, enabling clearer resource visibility and more fine-grained access control across regions within the shared account.

How the Feature Works

Enterprise projects are a cloud resource management feature provided by Huawei Cloud that allow you to group cloud resources by project, department, or business. For more information, see **Enterprise Project Management Service**.

A newly created cluster belongs to the **default** project by default. To move a cluster from the default project to a custom project, you simply perform a resource migration. There is no need to re-create the cluster.

Here, resource migration is the process of moving a CSS cluster from one enterprise project (for example, **default**) to another (for example, **TEST**).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- Impact: Resource migration only changes the management metadata. It does not restart the cluster or interrupt services.
- Permission control change: After the migration, only users who have been granted permissions on the new project (or all resources) can manage the cluster.

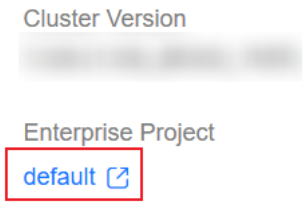
Prerequisites

- Enterprise Project Management Service (EPS) has been enabled for your account. For details, see section "Enabling Enterprise Project" in *Enterprise Management User Guide*Enabling Enterprise Project.
- You have created the destination enterprise project (for example, **TEST**). For details, see section "Creating an Enterprise Project" in *Enterprise Management User Guide*Creating an Enterprise Project.

Migrating a Cluster Between Enterprise Projects

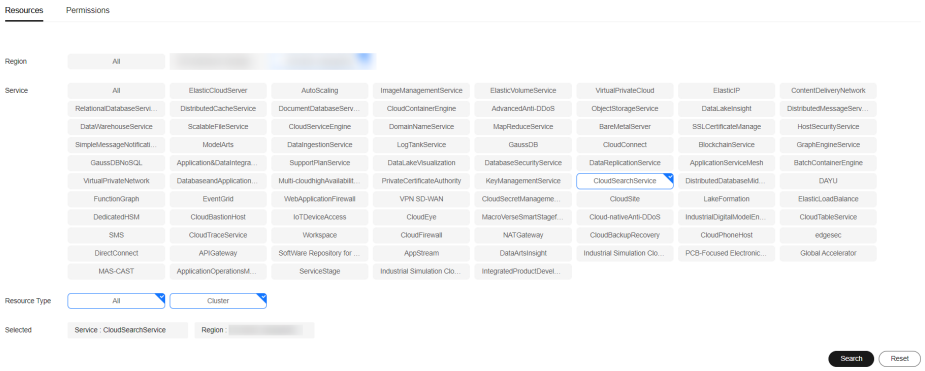
1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, find **Enterprise Project** in the **Basic Information** area, and click the enterprise project name under it. The **Project Management** console is displayed.

   Figure 2-102 Going to the Project Management console

5. On the **Resources** tab on the Project Management console, select the region of the current cluster, and select **CloudSearchService** as **Service**. Then click **Search**. The CSS clusters are displayed in the resource list below.

   Figure 2-103 Filtering CSS clusters

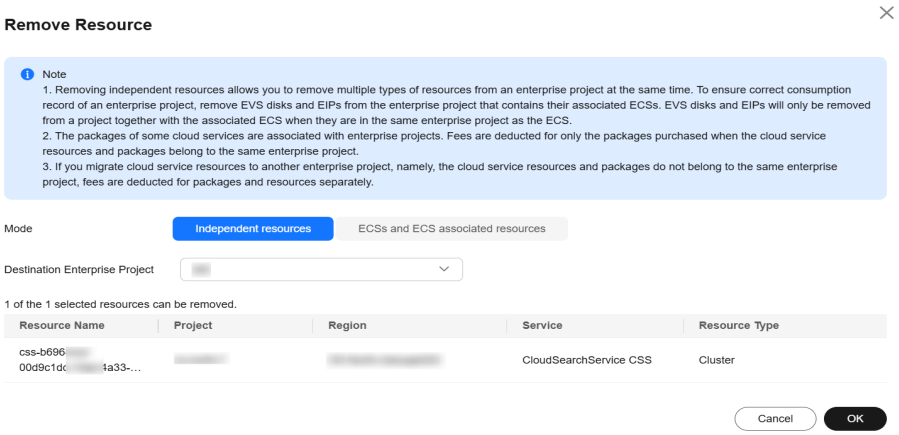
   

6. Select the cluster whose enterprise project you want to modify and click **Remove**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

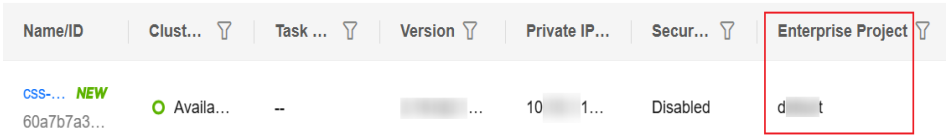
7. In the **Remove Resource** dialog box, specify **Mode** and select **Destination Enterprise Project**. Confirm the change information, and click **OK**.

Figure 2-104 Removing resources



8. After the change is complete, return to the Elasticsearch cluster list on the CSS console and check the cluster's new enterprise project to confirm the operation result.

Figure 2-105 Enterprise project



FAQ

- **Question 1: Does changing the enterprise project change a cluster's IP address?**

  Changing a cluster's enterprise project does not change its IP address. It merely changes the logical ownership of cloud resources. There is no change at the network layer.

- **Question 2: What should I do if, after a cluster's enterprise project is changed, the cluster is hidden from users under the original project?**

  After the change, the cluster management permissions are transferred to the new project. Check whether your account has the read permission on the new (destination) enterprise project.

2.18.6 Restarting a Cluster

A cluster restart may be necessary for purposes such as configuration updates or fault recovery. CSS Elasticsearch clusters support two restart modes: quick restart and rolling restart. To balance cluster availability and restart efficiency, select the mode best suited to your operational requirements.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Comparing the Two Restart Modes

Table 2-242 Introducing the two restart modes

Restart Mode	Quick Restart	Rolling Restart
Impact	High: The cluster becomes unavailable for both read and write operations for the entire duration of the upgrade.	Low: Only the node that is being restarted is unavailable. The cluster as a whole remains available all the time.
How It Works	Parallel operations: All selected nodes are stopped and restarted at once.	Serial operations: Restart node A > Wait for it to recover > Restart node B > Wait for it to recover, and so on
Data Security	There is risk of data loss if you forcibly restart nodes that are still serving write requests.	Data security is guaranteed as the system automatically handles shard allocation.
Time Required	All nodes are restarted simultaneously. The total restart time is typically shorter.	The selected nodes are restarted one at a time. After the first node, each node can restart only after the previous node has fully recovered. This process takes longer.
When to Use	Use this mode in development or testing environments, where clusters are completely deadlocked, or where service interruption is acceptable.	Use this model in production environments, where high availability takes top priority and clusters store large amounts of data.
Node Quantity	There is no limit on the number of nodes.	There are at least three nodes.

Constraints

Clusters whose **Task Status** is **Frozen** or that have ongoing tasks cannot be restarted.

Before You Start

- Check the target cluster's health status. Check the **Task Status** column to ensure there are no ongoing tasks.
- Pause high-concurrency data ingestion tasks to prevent write failures caused by primary shard changes during the restart.
- Avoid restarting a cluster during peak hours, as doing so may impact data integrity and service availability.
- Before performing a **rolling restart**, check that indexes have at least one replica configured.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Restart Considerations for Different Node Types

Before restarting nodes by node type or node name, review the following considerations based on the role of the target nodes.

Table 2-243 Restart considerations

Node Type	Restart Considerations

Data nodes and cold data nodes	• In production environments, you are advised to perform rolling restart in the following sequence: **Pause shard allocation > Restart one node > Wait for shard recovery to complete > Restart the next node**. For details, see **Related Operation: Managing Shard Allocation Policies**.
	• For clusters with a large number of shards, you are advised to increase the delayed shard allocation timeout to prevent a node's temporary unavailability from triggering unnecessary shard reallocation and placing excessive load on the cluster. The default timeout is 1 minute and can be adjusted as needed:
	`PUT _all/_settings`
	`{`
	`  "settings": {`
	`  "index.unassigned.node_left.delayed_timeout": "5m"`
	`  }`
	`}`

Master nodes	• For production clusters, do not restart all master nodes at once. Master nodes manage cluster metadata. Restarting all master nodes at once triggers metadata reallocation and prolongs cluster recovery.
	• You are advised to restart master nodes one at a time by node name, with the current master node restarted last.
	Run the following command to identify the current master node (the node with ***** in the **master** column is the current master node):
	`GET /_cat/nodes?v`
	Example response:
	`ip heap.percent ram.percent cpu load_1m node.role master name`
	`192.168.67.132 54 96 5 0.79 dimr * css-ess-esn-3-1`
	`192.168.100.80 22 94 4 0.58 dimr - css-ess-esn-1-1`

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Node Type	Restart Considerations
Client nodes	• Client nodes are responsible only for request forwarding and preprocessing. Restarting a client node does not trigger shard reallocation, but requests routed to that node will fail. Before restart a client node, check whether any applications connect directly to the node's IP address. • Client nodes can be restarted in batches and recover quickly. You can also switch the application access IP addresses in advance to avoid request failures.

Quick Restart

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and choose **More > Restart** in the **Operation** column.
4. On the **Restart Cluster** page, set parameters.

   Table 2-244 Configuring a quick restart

Parameter	Description
Restart Mode	Select **Quick Restart**.
Select By	Select **Node type** or **Node name**. • **Node type**: Select one or more node types for restart. For example, you can use this option to restart all client nodes to release connections. • **Node name**: Restart only one node at a time. Use this option to restart individual nodes, for example, to restore faulty nodes.
Operation Value	Select one or more node types or a node name, depending on the setting of **Select By**. See **Restart Considerations for Different Node Types** for more information.

5. Manually type in **YES**, and click **OK** to start the restart task.

   WARNING

   The cluster will be unavailable for the entire duration of a quick restart.

6. When the cluster status changes from **Processing** to **Available**, the cluster has restarted successfully.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Rolling Restart

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters** > **Elasticsearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Restart** in the **Operation** column.
4. On the **Restart Cluster** page, set parameters.

   Table 2-245 Configuring a rolling restart

Parameter	Description
Restart Mode	Select **Rolling Restart**. This option is unavailable when the cluster has fewer than three nodes.
Select By	The default value is **Node type**.
Value	Select node types for the restart. See **Restart Considerations for Different Node Types** for more information. • **Select all**: Select all node types. • Select specific node types: For instance, to restart cold data nodes, select **Cold Data Nodes** only. The system will skip all other nodes.

5. Manually type in **YES**, and click **OK** to start the restart task.
6. When the cluster status changes from **Processing** to **Available**, the cluster has restarted successfully.

Related Operation: Managing Shard Allocation Policies

NOTE

During a rolling restart, the system automatically handles shard allocation, so normally you do not need to manually change this setting. This operation is primarily intended for scenarios where nodes are manually restarted in batches by node type or node name during a quick restart.

Node restart can trigger shard reallocation, especially when nodes are manually restarted in batches by node type or node name during a quick restart. You are advised to disable automatic shard allocation before restarting the nodes and manually restore the automatic shard allocation policy after the restart is complete. This prevents a large number of shards from being reallocated when nodes go offline, which could otherwise cause the cluster to remain unavailable for an extended period.

1. Disable automatic shard reallocation.

```
PUT /_cluster/settings
{
  "persistent": {
    "cluster.routing.allocation.enable": "none"
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. After all target nodes have been restarted, restore the automatic shard allocation policy.

```
PUT /_cluster/settings
{
  "persistent": {
    "cluster.routing.allocation.enable": "all"
  }
}
```

FAQ

- **Question 1: What should I do if a rolling restart is stuck in Restarting for a long time?**

  During a rolling restart, after the last node is restarted, the system waits for the cluster to recover (health status turns green/yellow). If the cluster has extremely large indexes or index shards are still being restored, the task will be stuck in **Restarting** for a long time. You can run the **GET _cat/recovery?v** command to check the shard recovery progress.

- **Question 2: How do I perform a rolling-like restart on a cluster with just two nodes?**

  The rolling restart option is unavailable for clusters with just two nodes, but you can achieve a similar process manually:

  a. Check the number of index replicas. To ensure service continuity, ensure that each index has at least one replica.

  b. Select **Fast Restart**, then select **Node name** for **Select By**. Select one of the nodes to restart it.

  c. Wait for this node to fully recover and the cluster status to change to **Available**. Then restart the other node.

2.18.7 Deleting a Cluster

When an Elasticsearch cluster is no longer required (for example, after a project is decommissioned or a temporary environment is retired), you can delete the cluster to release compute and storage resources and stop billing. Cluster deletion is an irreversible high-risk operation that permanently removes all data in the cluster. Before deleting a cluster, ensure that all the required data has been backed up.

Before You Start

Cluster deletion is an irreversible high-risk operation. Before proceeding, confirm the following:

- **Data has been backed up**: Ensure that cluster data has been backed up to an OBS bucket using snapshots. Without backups, the data will be permanently lost once deleted.
- **There are no ongoing tasks**: Check cluster monitoring metrics and ensure that both the average indexing rate and average query rate are 0, indicating that the cluster has no ongoing tasks.
- **Snapshot availability has been confirmed**: If snapshots have been enabled for the cluster and snapshots are still retained in the OBS bucket, you can use these snapshots to restore data later if necessary. For details, see **Can a Deleted CSS Cluster Be Restored?**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- **VPC Endpoint permissions have been confirmed**: If the cluster has enabled the VPC Endpoint service, the account that you use must have permission to delete VPC endpoints. Otherwise, VPC endpoint resources cannot be released after cluster deletion, and you will continue to be charged for them. For details about how to configure this permission, see **Permissions**.
- Dedicated load balancer configuration has been confirmed: If the cluster is connected through a dedicated load balancer (ELB), deleting the cluster releases the load balancer's listener resources. However, you must manually delete the ELB to prevent further charges.

Deleting a Pay-per-Use Cluster

The cluster is deleted immediately, resources are released, and pay-per-use billing stops.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
3. In the cluster list, find the target cluster, and choose **More > Delete** in the **Operation** column.
4. In the displayed dialog box, read the risk warning carefully and enter **DELETE** in the text box.
5. Click **OK**.

   After the cluster is deleted, it is removed from the cluster list.

FAQ

- **Q1: Will snapshots stored in an OBS bucket be deleted together with the cluster?**

  No. Deleting a cluster only releases its compute and storage resources. Snapshots stored in OBS will be retained. You can use these snapshots to restore data to a new cluster. However, note that you will continue to be charged for OBS storage. If you do not need these snapshots, manually delete them.
- **Q2: Can data be recovered if a cluster is accidentally deleted?**

  If the cluster has snapshots stored in an OBS bucket, you can restore its data from one of these snapshots. If snapshots were never enabled or no snapshots were ever created for the cluster, then there is no way to restore cluster data. For production clusters, enable an automatic snapshot policy to create snapshots periodically and reduce the risk of data loss.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3 OpenSearch

3.1 Procedure

This topic introduces the end-to-end procedure for using CSS OpenSearch to build enterprise-grade search and analytics applications, covering key steps from environment setup to application rollout.

Figure 3-1 Procedure for using CSS OpenSearch

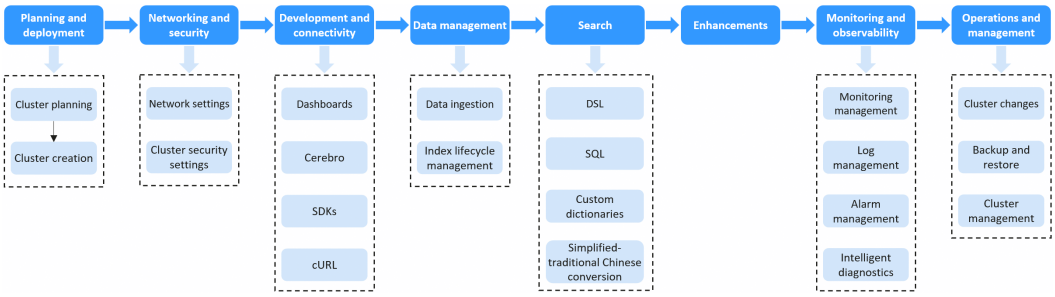


Table 3-1 Procedure for using OpenSearch

Stage	Key Step	Description	Details
Planning and deployment	Cluster planning	Based on projected data volumes and intended use cases, plan node specifications, multiple-AZ deployment, and security mode in advance to minimize the need for changes after cluster creation.	**Planning Cluster AZs and HA** **Planning the Cluster Version and Security Mode** **Planning Node Types and Resource Specifications**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Stage	Key Step	Description	Details
	Creating a cluster	Create an OpenSearch cluster based on your planning and service requirements.	**Creating a Cluster (New Version)**
Networking and security	Network settings	CSS OpenSearch clusters support four networking modes. Choose the configuration that best suits your connectivity needs.	**Overview**
	Security settings	Configure user permissions and authentication policies to ensure cluster access security.	**Creating Cluster Users and Granting Permissions** **Configuring LDAP Authentication**
Development and connectivity	Connecting to your cluster	CSS OpenSearch provides multiple connection options tailored to different roles, programming languages, and network environments. At the application layer, CSS supports open-source RESTful APIs and SDKs for popular languages. It also includes built-in visualization and management tools such as OpenSearch Dashboards and Cerebro.	**Cluster Connection Methods** **Connecting to Cluster Through OpenSearch Dashboards** **Connecting to a Cluster Through Cerebro**
Data management	Data ingestion	CSS OpenSearch clusters provide multiple data ingestion options, including Logstash, open-source OpenSearch APIs, and the Cloud Data Migration (CDM) service. Choose one that best suits your service requirements.	**Migrating Data Using CDM** **Ingesting Data Using Self-Managed Logstash** **Ingesting Data Using Open-Source OpenSearch APIs**
	Index lifecycle management	CSS OpenSearch clusters provides the Index State Management (ISM) plugin that you can use to configure and manage index lifecycle policies and automate index rollover. This helps reduce costs and improve efficiency.	**Configuring Index State Management (ISM) Policies**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Stage	Key Step	Description	Details
Search	Search and analytics	CSS OpenSearch clusters support both DSL and SQL as query languages, along with an efficient tokenization plugin. It helps you build an intelligent search and big data analytics platform for efficient data mining and accurate insights.	**Using DSL for Search** **Using SQL for Search** **Configuring Custom Dictionaries** **Configuring Simplified-Traditional Chinese Conversion Search**
Enhancements	CSS enhancements over open-source OpenSearch	CSS offers major enhancements over open-source OpenSearch, delivering better performance, stability, and more cost-effective storage.	**CSS Enhancements Over Open-Source OpenSearch**
Monitoring and observability	Monitoring management	Cloud Eye enables efficient visualization and monitoring of key cluster metrics as well as index-level monitoring. The O&M team can monitor cluster health and performance bottlenecks in real time, ensuring high service availability.	**Using Cloud Eye to Monitor Clusters** **Configuring Index-Level Monitoring** **Supported Metrics**
	Log management	The O&M team can analyze run logs, slow query logs, and operation audit records to quickly demarcate issues, analyze performance, and ensure security and compliance. Logs can be ingested into OpenSearch in real time or backed up to OBS for long-term storage, meeting diverse, enterprise-grade operational and auditing requirements.	**Querying Logs** **Ingesting Logs into a Cluster** **Backing Up Logs to OBS** **Viewing Audit Logs**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Stage	Key Step	Description	Details
	Alarm management	CSS OpenSearch clusters support custom alarm rules, flexible notification channels, and seamless integration with the native Prometheus monitoring ecosystem. Integration with SMN enables real-time push of cluster exceptions, shifting from passive response to proactive defense and ensuring service continuity.	**Configuring OpenSearch Alert Notifications via SMN** **Synchronizing Alerts and Metrics to Prometheus**
	Intelligent diagnostics	The O&M team can easily perform periodic health checks on clusters, and comprehensive results enable prompt risk detection and handling, improving cluster reliability and availability.	**Intelligent Diagnostics**
Operations and management	Managing cluster changes	You can easily manage cluster changes throughout the entire lifecycle, such as cluster scaling, version upgrade, multi-AZ configuration, and HA configuration, all while ensuring service continuity.	**Scaling Out/Up** **Scaling In/Down** **Changing Node Specifications** **Replacing a Faulty Node** **Upgrading the Cluster Version** **Switching AZs** **Changing the Security Mode**
	Backup and restore	CSS OpenSearch provides comprehensive backup and restore capabilities for clusters. Using snapshots, you can save point-in-time copies of cluster data in OBS buckets and quickly restore them in the event of data loss or other operational needs. This improves data reliability.	**Creating Snapshots to Back Up Data** **Restoring Data Using Snapshots** **Deleting a Cluster Snapshot**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Stage	Key Step	Description	Details
	Cluster management	Cluster resources can be isolated by enterprise project and tagged for refined management. You can easily restart and delete clusters and optimize them through YML parameters, ensuring efficient cluster management.	**Viewing Cluster Details** **Checking Default Cluster Plugins** **Modifying Cluster Parameters** **Managing Cluster Tags** **Associating a Cluster with an Enterprise Project** **Restarting a Cluster** **Deleting a Cluster**

3.2 Cluster Planning

3.2.1 Planning Cluster AZs and HA

This topic describes how to improve cluster availability by deploying a cluster across multiple AZs, including the rules for distributing cluster nodes among AZs, the recommended number of replicas, and policies for handling AZ failures.

An availability zone (AZ) is a physical region where resources use independent power supplies and networks. AZs in the same region can communicate with each other through internal networks but are physically isolated.

Multi-AZ deployment is a high availability feature provided by CSS. Deploying a cluster across two or three AZs located in the same region can help prevent data loss and lower the possibility of service outages.

Suggestions on Multi-AZ Deployment

If you select multi-AZ deployment when creating a cluster, CSS automatically enables cross-AZ HA to ensure that cluster nodes will be evenly distributed across the selected AZs (the difference between the number of nodes in each AZ cannot exceed 1).

You are advised to select three AZs, instead of two AZs, for multi-AZ deployment. If only two AZs are selected and one AZ becomes faulty, the cluster may not be able to elect a master node. As a result, the cluster may become unavailable.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Node Distribution Rules

Table 3-2, Table 3-3, and Table 3-4 show the distribution of data nodes, cold data nodes, master nodes, and client nodes across AZs, depending on the node and AZ counts. The node distribution rules for multi-AZ clusters are as follows:

- For a multi-AZ cluster, the system evenly distributes nodes across AZs. The difference in the number of same-type nodes across different AZs will not exceed 1.
- To ensure high service availability and prevent individual node overload in a multi-AZ deployment, the following node count requirements must be met:
  - **For a two-AZ cluster**, there must be at least four data nodes (or cold data nodes), two for each AZ; and there must be at least two client nodes (if configured), one for each AZ.
  - **For a three-AZ cluster**, each available node type must have at least one node in each AZ. This means each node type must have at least three nodes in total.
  - For node types that are not deployed (for example, no cold data nodes), the minimum node count requirement does not apply.
- If the number of data nodes or cold data nodes is not an integer multiple of the number of AZs, for example, five data nodes in a two-AZ cluster, uneven data distribution may occur, which can impact query and ingestion performance. For this reason, you should always ensure that the number of nodes is an integer multiple of the number of AZs.

Table 3-2 Number of data nodes/cold data nodes and AZ distribution

Total Nodes	Single AZ	Two AZs		Three AZs		
	AZ1	AZ1	AZ2	AZ1	AZ2	AZ3
1	1	Not supported		Not supported		
2	2	Not supported		Not supported		
3	3	Not supported		1	1	1
4	4	2	2	2	1	1
5	5	3	2	2	2	1
6	6	3	3	2	2	2
7	7	4	3	3	2	2
...	...	...	...	...	...	...

Table 3-3 Number of master nodes and AZ distribution

Total Nodes	Single AZ	Two AZs		Three AZs		
	AZ1	AZ1	AZ2	AZ1	AZ2	AZ3

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3	3	2	1	1	1	1
5	5	3	2	2	2	1
7	7	4	3	3	2	2
9	9	5	4	3	3	3

Table 3-4 Number of client nodes and AZ distribution

Total Nodes	Single 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
...	...	...	...	...	...	...

Replica Configuration Suggestions

For a multi-AZ cluster, configure the number of index replicas in a manner that can better capitalize on the high availability that comes with such as deployment.

- With a dual-AZ deployment, if one AZ becomes unavailable, the other AZ continues to provide services. In this case, configure at least one replica. If higher query performance is desired, you can increase the number of replicas.
- In the case of a three-AZ deployment, if one AZ becomes unavailable, the other AZs can continue to provide services. In this case, also configure at least one replica. To enhance the cluster's query performance, increase the number of replicas.

For Elasticsearch and OpenSearch clusters, the default number of index replicas is 1. To enable more replicas, modify relevant settings. The following are two examples.

- Adjust the number of replicas for an existing index:

  curl -XPUT http://ip:9200/{index_name}/_settings -d '{"number_of_replicas":2}'
- Set the number of replicas for new indexes using a template:

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

where, **ip** indicates the private IP address of the cluster; **index_name** indicates the index name; **templatename** indicates the template name; **template** indicates the index name matching rule (meaning the template will automatically apply to indexes that match this rule. The asterisk (*) indicates that the template will apply

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

to all new indexes); and **number_of_replicas** indicates the number of index replicas to change to. In this example, the number of index replicas is changed to 2.

Service Outage Pattern Analysis

Master nodes manage cluster-wide operations, including metadata, indexes, and shard allocation. In a cluster with master nodes, the master nodes perform these tasks. In a cluster without them, data nodes and cold data nodes share the responsibilities of the master nodes.

In the case of a single AZ failure, if fewer than half (including half) of the nodes assuming master node responsibilities are still available, service availability will be affected. In this case, you need to restore services by referring to **Table 3-5**.

For example, a cluster has three master nodes distributed in two AZs. If the AZ that contains one master node becomes unavailable while the other AZ remains available, services will still be available. However, if it is the other AZ, which contains two of the three master nodes, that fails, services will be interrupted, because fewer than half of the master nodes (or nodes that assume master node responsibilities) are available. In this case, restore services by referring to **Table 3-5**.

Table 3-5 Possible service outage patterns in the face of the failure of a single AZ

Number of AZs	Number of Master Nodes	Service Outage Patterns and Handling Suggestions
2	0 (In this case, data nodes and cold data nodes share the responsibilities of master nodes.)	• If the number of data nodes plus cold data nodes is an even number: If half of the nodes become unavailable due to the failure of a single AZ, you must add one node to the still available AZ to ensure a master node can be elected for the cluster. To add nodes to your cluster, contact technical support. • If the number of data nodes plus cold data nodes is an odd number: – If the failed AZ is the one that has one more node, you must add two nodes to the surviving AZ to ensure a master node can be elected for the cluster. To add nodes to your cluster, contact technical support. – If the failed AZ is the other one—the one that has one less node, services will not be interrupted and a master node can be elected.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Number of AZs	Number of Master Nodes	Service Outage Patterns and Handling Suggestions
2	3	There is a 50% chance of service interruption when one AZ fails. In this situation, one AZ has two master nodes, and the other has one: • If the AZ that has one master node fails while the other AZ remains available, two master nodes are still available, so a master node can be elected from among them. In this case, services remain available. • If the failed AZ is the other one, only one master node survives, in which case, no master node can be elected, and services will be interrupted. In this case, contact technical support.
3	0 (In this case, data nodes and cold data nodes share the responsibilities of master nodes.)	There is a small chance of service interruption when a single node fails. However, there is one high-risk scenario: If the cluster spans three AZs and the total number of data nodes plus cold data nodes is 4, so the node distribution across the three AZs is 2-1-1, the failure of the AZ containing two nodes will cause a service interruption. To avoid this, avoid selecting four nodes for a three-AZ cluster.
3	3	A single AZ failure will not result in service interruptions.

3.2.2 Planning the Cluster Version and Security Mode

This topic introduces different OpenSearch cluster versions, including features supported by each version and its application scenarios. It also describes the characteristics and application scenarios of different cluster security modes (including different application-layer protocols).

Introduction to Different Cluster Versions

When selecting an OpenSearch cluster version, consider factors such as service requirements, available features, performance, security updates, and long-term support, ensuring that the selected version can meet both current and future needs and provide a stable, secure environment for your data.

- If you are deploying CSS OpenSearch clusters for the first time, you are advised to use the latest version.
- If you are migrating an in-house built or third-party OpenSearch cluster to CSS without altering the cluster, keep the version of the source cluster.
- If you are migrating a self-managed or third-party OpenSearch cluster to CSS while recoding it, choose OpenSearch 2.19.0 or 3.4.0.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-6 CSS cluster version enhancements

Category	Feature	OpenSearch 1.3.6	OpenSearch 2.17.1	OpenSearch 2.19.0	OpenSearch 3.4.0	Related Documents
Vector search	Open-source vector search	√	√	√	√	**k-NN query**
	CSS vector search	√	x	√	√	**CSS Vector Database**
Cost optimization	Decoupled storage and compute	√	x	√	√	**Configuring Decoupled Storage and Compute**
	Switching between hot and cold storage tiers	√	√	√	√	**Switching Between Hot and Cold Data**
Enhanced stability	Read/write splitting	x	x	√	√	**Configuring Read/Write Splitting Between Leader and Follower**
	Flow control	x	x	√	√	**Configuring Flow Control**
	Query traffic isolation	x	x	√	√	**Configuring Query Traffic Isolation**
	Large query isolation	x	x	√	√	**Configuring Large Query Isolation**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Category	Feature	OpenSearch 1.3.6	OpenSearch 2.17.1	OpenSearch 2.19.0	OpenSearch 3.4.0	Related Documents
	OpenSearch Dashboards query response circuit breaker	x	x	√	√	**Configuring the OpenSearch Dashboards Query Response Circuit Breaker**
	Automatic retry for failed shard allocation	x	x	√	√	**Configuring Automatic Retry for Failed Shard Allocation**
Enhanced performance	Enhanced ingestion performance	√	x	√	√	**Enhancing Data Ingestion Performance**
	Enhanced aggregation performance	x	x	√	√	**Configuring Enhanced Aggregation**
Enhanced operations	Index recycle bin	x	x	√	√	**Enabling the Index Recycle Bin**
	Query resource tracker	x	x	√	√	**Tracking Query Resource Consumption**
	Index-level monitoring	x	x	√	√	**Configuring Index-Level Monitoring**

Cluster Security Modes

Configure the security mode based on the security needs of your cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

CAUTION

HTTPS can only be disabled for OpenSearch 1.3.6, 3.4.0, and 2.19.0 clusters with security mode enabled. For other versions, HTTPS access cannot be disabled.

Table 3-7 Cluster security modes

Cluster Type	Key Settings	Description	Applicable Scenario
Non-security mode cluster	**Security Mode**: disabled	Access to such a cluster requires no user authentication, and data will be transmitted in plaintext using HTTP.	Use when creating a cluster for internal testing or workloads that have a low security standard. • Advantage: easy to access the cluster. • Disadvantage: poor security, as anyone can access the cluster. When security mode is disabled, public network access and Dashboards public network access cannot be enabled. To ensure data security, deploy the cluster in a secure environment. Do not expose the cluster's network interface to the public network.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Cluster Type	Key Settings	Description	Applicable Scenario
Security-mode cluster	Security-mode cluster + HTTP: • **Security Mode**: enabled • **HTTPS Access**: disabled	Such a cluster requires user authentication. It supports access control and data encryption, and it uses HTTP to transmit data in plaintext.	Use to balance security and performance. • Advantage: User authentication improves cluster security. HTTP-based access ensures high performance of the cluster. User permissions can be configured to ensure proper isolation. • Disadvantage: Public network access is not supported. To ensure data security, deploy the cluster in a secure environment. Do not expose the cluster's network interface to the public network.
	Security-mode cluster + HTTPS: • **Security Mode**: enabled • **HTTPS Access**: enabled	Such a cluster requires user authentication. It supports access control and data encryption, and it uses HTTPS to encrypt communication and enhance data security.	Use when security takes precedence over performance and public network access is required. • Advantage: User authentication improves cluster security. HTTPS enhances cluster security by encrypting all communication over the public network. User permissions can be configured to ensure proper isolation. • Disadvantage: When HTTPS is used, data encryption and decryption introduce computational overhead and impact the cluster's read and write performance.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

To access a security-mode cluster, a username and password need to be provided. CSS supports authentication for the following two types of users:

- Administrator: The default administrator username is **admin**, and the password is the one specified during cluster creation.
- Cluster user: created by the cluster administrator on OpenSearch Dashboards. User permissions can be configured to ensure proper isolation. For details, see **Creating Cluster Users and Granting Permissions**.

3.2.3 Planning Node Types and Resource Specifications

This topic describes common application scenarios and provides configuration recommendations for each type of node in a cluster, including data nodes, master nodes, client nodes, and cold data nodes. The goal is to help you select and configure the appropriate nodes based on your specific service requirements. It also provides suggestions on selecting node specifications and configuring the node storage type, storage capacity, and node quantity, helping you properly plan the capacities of your cluster.

Key Points for Node Configuration

Table 3-8 Key points for node configuration

Configuration Item	Data Nodes	Master Nodes	Client Nodes	Cold Data Nodes
Mandatory or Not	Mandatory	Optional	Optional	Optional
Primary role	Stores data. Can also serve as a master, ingest, or analytics node.	Manages the cluster, including metadata and shard allocation.	Receives and coordinates external requests, such as search and indexing requests.	Stores infrequently accessed historical data.
Node specifications	Select node specifications based on your service requirements (such as data volume, performance requirements, and budget). For details, see **Node Specifications**.	Only EVS disks are supported. Select higher node specifications for large clusters.	Only EVS disks are supported. Select higher node specifications for heavy read/write workloads.	Select node specifications based on your service requirements (such as data volume, performance requirements, and budget). For details, see **Node Specifications**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuration Item	Data Nodes	Master Nodes	Client Nodes	Cold Data Nodes
Storage capacity	• EVS disk: Configure as needed (must be a multiple of 20 GB). For details, see **Recommended Node Storage Capacity**. • Local disk: Fixed capacity.	Fixed at 40 GB.	Fixed at 40 GB.	• EVS disk: Configure as needed (must be a multiple of 20 GB). For details, see **Recommended Node Storage Capacity**. • Local disk: Fixed capacity.
Number of nodes	• With dedicated master nodes: 1–200; without dedicated master nodes: 1–32. • A minimum of 3 is recommended. For details, see **Recommended Node Quantity**.	3, 5, 7, or 9. For details, see **Recommended Master Node Quantity**.	1–64. For details, see **Recommended Client Node Quantity**.	1–32. For details, see **Recommended Node Quantity**.

Node Types

Before creating a cluster, determine the types of nodes to use based on service requirements, query load, data growth patterns, and performance goals.

For each cluster, data nodes are mandatory, while master, client, and cold data nodes are optional. Choose whether to enable the latter three based on your service needs and performance requirements.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-9 Introduction to different node types

Node Type	When to Use
Data node (ESS)	Data nodes are primarily used to store the cluster's data. The functions of the data nodes vary depending on the node types enabled for the cluster. • If **Master node** and **Client node** are both selected, data nodes will be used for data storage only. • If **Master node** and **Client node** are both unselected, data nodes will be used for all of the following purposes: cluster management, data storage, cluster access, and data analysis. To ensure reliability, a cluster should have at least three nodes. • If **Master node** is selected but **Client node** is not, data nodes will be used for data storage, cluster access, and data analysis. • If **Master node** is unselected but **Client node** is selected, data nodes will be used for data storage and cluster management.
Master node (ess-master)	Master nodes manage cluster-wide operations, including metadata, indexes, and shard allocation. For large-scale deployments, using dedicated master nodes enhances cluster stability, service availability, and centralized control. • Large-scale cluster: For a cluster that has more than 10 nodes, you are advised to add dedicated master nodes to effectively manage the cluster status and metadata. • Large quantities of indexes and index shards: If the number of indexes or shards exceeds 10,000, master nodes will have better performance handling complex cluster management tasks, avoiding impact on the performance of data nodes. • Better management of cluster nodes: Master nodes maintain the cluster metadata, including index mapping, settings, and aliases. For a complex cluster structure, dedicated master nodes offer better management, including node joining, exiting, and fault detection. Master nodes play a critical role in cluster node management. • Improved cluster stability and reliability: Dedicated master nodes improve cluster stability and reliability by taking over cluster management responsibilities from data storage and query nodes. • Optimized performance for data nodes: By offloading cluster management tasks from data nodes to master nodes, you can allow data nodes to focus on data processing, which leads to improved performance.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Node Type	When to Use
Client node (ess-client)	Client nodes route and coordinate search and index requests, offloading processing from data nodes for enhanced query performance and cluster scalability when there are heavy loads. • High queries per second (QPS): In the face of a high QPS, a dedicated client node can evenly distribute query requests, reducing the load of data nodes and improving the overall query performance. • Complex aggregation queries: For complex, compute-intensive aggregation queries, a client node can dedicate to the handling of aggregation results, thus improving the efficiency and response speed of such queries. • Large number of shards: In a cluster with a large number of shards, a client node can effectively coordinate and manage query requests to each shard, improving efficiency in request forwarding and processing. • Reducing the load of data nodes: A client node parses search requests, determines the locations of index shards, and coordinates different nodes to execute searches. This reduces the load of data nodes by allowing them to focus on data storage and indexing. • Improved cluster scalability: The use of client nodes allows for better cluster scalability and flexibility, enabling support for large datasets and more complex query requirements.
Cold data node (ess-cold)	Cold data nodes are used to store and query latency-insensitive data in large quantities. They offer an effective way to manage large datasets while cutting storage costs. • Storage of historical data in large quantities: Cold data nodes offer a more cost-effective solution for storing large quantities of historical data that are infrequently accessed but useful for analytical purposes. • Optimizing hot data performance: By migrating cold data to cold data nodes, you reduce the storage load of hot data nodes, thereby optimizing their query and write performance. • Insensitivity to query latency: Cold data nodes are a better option for storing data that is insensitive to a high query latency. • Cost-effectiveness: Cold data nodes usually use large-capacity disks that offer inexpensive storage.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

NOTE

- If no master, client, or cold data nodes were enabled when a cluster was created, you can add them if data nodes become overloaded later at some point. For details, see **Adding New Node Types**.
- When cold data nodes are enabled, you can switch between cold and hot data storage. For details, see **Switching Between Hot and Cold Data**. If there are no cold data nodes, we recommend that you use decoupled storage and compute, which can also cut storage costs. For details, see **Configuring Decoupled Storage and Compute**.

Suggestions on Data Node Configuration

Table 3-10 Data node configuration

Parameter	Configuration Suggestions
Node Specifications	In the node flavor list, **vCPUs | Memory** indicate the number of vCPUs and memory capacity available for each flavor, and **Recommended Storage** indicates the supported storage capacity range. We recommend that you select node specifications based on service needs, such as the data volumes, performance requirements, and your spending budget. **Node Specifications** describes the application scenarios and core features of different node specifications. It can help you properly plan your cluster. For more information about different node specifications, see **ECS Types**.
Node Storage Type and Capacity	• If the selected node flavor uses EVS disks, you need to further select **Node Storage Type** and **Capacity** based on service requirements. – For more on EVS disk performance, see **Disk Types and Performance**. – The value range of node storage capacity is determined by the node flavor you select. The value must be divisible by 20. For how to calculate the required node storage capacity, see **Recommended Node Storage Capacity**. – Node storage capacity cannot be reduced once the cluster is created. Evaluate your long-term data needs and select an appropriate size.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Configuration Suggestions
Nodes	• If master nodes are configured, the number of data nodes ranges from 1 to 200. If the combined number of data nodes and cold data nodes exceeds 100, you are advised to select a flavor with at least 16 vCPUs. This helps prevent delays in cluster status synchronization across nodes and ensures optimal read and write performance. • If no master nodes are configured, the number of data nodes ranges from 1 to 32. • To ensure cluster availability, you should configure at least three data nodes. For how to calculate the number of nodes needed, see **Recommended Node Quantity**. If the number of data nodes in your cluster is not evenly divisible by the number of AZs, data distribution may become unbalanced across nodes. This will negatively impact both query and write performance.

Node Specifications

The choice of node specifications affects disk deployment. There are two types of disks: local disks and EVS disks. For example, high-performance nodes may use local disks, while standard nodes may use cost-effective EVS disks.

- EVS disks are a virtual block storage service that is independent of ECSs. They provide high reliability and fast elasticity, making them ideal for workloads that require high data reliability and highly scalable storage capacity.

Recommended Node Storage Capacity

If you select EVS disks, you can select a node storage capacity within the range supported by the selected node flavor.

When planning the storage capacity of a cluster, consider the following factors: the original data size, number of data replicas, data bloat rate, and disk usage. The following is a recommended formula for determining the needed cluster storage capacity:

Storage capacity = Original data size x (1 + Number of replicas) x (1 + Data bloat rate) x (1 + Ratio of reserved space)

where,

- Original data size: Determine the size of original data that needs to be stored.
- Number of replicas: The default value is 1.
- Data bloat rate: Extra data may be generated due to data indexing. You are advised to use a 25% data bloat rate.
- Disk usage: Considering the space occupied by the operating system and file system and the space reserved for optimized disk performance and

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

redundancy, you are advised to keep the disk usage under 70%. That is, you need to reserve 30% of the total disk capacity.

Using the recommended values above, the formula should be: Storage capacity = Original data size x 2 x 1.25 x 1.3 = Original data size x 3.25

This formula is for quick reference only. You still need to adjust it based on the actual applications and projected data growth rate. Moreover, the storage capacity must be divisible by 20.

Recommended Node Quantity

Data nodes and cold data nodes are used to store data. Their quantities largely determine the cluster performance baseline. The following formulas can be used to estimate the number of data nodes and cold data nodes needed:

Number of data nodes + Number of cold data nodes = Number of write nodes + Number of query nodes

Number of write nodes = Traffic during peak hours (MB/s) / Number of vCPUs per node / Per-vCPU write performance baseline (MB/s) x Number of index replicas

Number of query nodes = Query QPS / (Number of vCPUs per node x 3 / 2 / Average query latency (s)) x Number of index shards

where,

- The number of vCPUs per node is determined by the selected node flavor.
- The per-vCPU write performance baseline is determined by the selected disk type. For a node that uses EVS disks, the per-vCPU write performance baseline is 1 MB/s. For a node that uses local disks, this baseline is 1.5 MB/s.
- QPS indicates the number of queries that can be processed per second.
- The average query latency is used as the query performance baseline.
- Where a high write performance is desired, we recommend setting the number of index shards to equal the total number of data nodes plus cold data nodes, thus evenly distributing the write load across all nodes.
- Where a high QPS is desired, we recommend setting the number of index shards to equal the total number of data nodes plus cold data nodes **minus 1**, thus evenly distributing the query load across all nodes.

Example:

If the peak traffic is 100 MB/s; the node specifications are 16 vCPUs and 64 GB RAM, with EVS disks; the query throughput is 1,000 QPS; the average query latency is 100 ms; and each index has three shards and two replicas, then:

Number of write nodes = 100/16/1 x 2 = 12

Number of query nodes = 1000/(16 x 3/2/0.1) x 3 = 12

Number of data nodes + Number of cold data nodes = Number of write nodes + Number of query nodes = 12 + 12 = 24

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Suggestions on Master Node Configuration

Table 3-11 Master node configuration

Parameter	Configuration Suggestions
Node Specifications	In the node flavor list, **vCPUs | Memory** indicate the number of vCPUs and memory capacity available for each flavor, and **Recommended Storage** indicates the supported storage capacity range. For a large-scale cluster with a large number of index shards, use large-capacity nodes as master nodes. Master nodes support EVS disks only. For more information about different node specifications, see **ECS Types**.
Node Storage Type and Capacity	Select an appropriate storage type and capacity for master nodes. • For more on EVS disk performance, see **Disk Types and Performance**. • The node storage capacity is fixed at 40 GB.
Nodes	The number of master nodes can be 3, 5, 7, or 9. For split-brain protection, the number of master nodes must be an odd number. For the recommended master node quantity, see **Recommended Master Node Quantity**.

Recommended Master Node Quantity

Increasing the number of master nodes can improve the cluster's fault tolerance performance, but it can also lead to issues like increased communication overhead, longer election time, and low resource utilization. You are advised to select an appropriate number of master nodes based on service requirements.

The following table provides the recommended number of master nodes based on the cluster size.

Table 3-12 Recommended number of master nodes

Cluster Size	Recommended Number of Master Nodes	Performance Expectation
Small (< 10 nodes)	3	Basic high availability
Medium (50 > number of nodes ≥ 10)	3 or 5	Balanced performance and high availability
Large (≤ 50 nodes)	5 or 7	Enhanced fault tolerance and stability

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Cluster Size	Recommended Number of Master Nodes	Performance Expectation
Extra-large (≥ 100 nodes)	5 or 7	Pay attention to excessive communication overhead.

Suggestions on Client Node Configuration

Table 3-13 Client node configuration

Parameter	Configuration Suggestions
Node Specifications	In the node flavor list, **vCPUs | Memory** indicate the number of vCPUs and memory capacity available for each flavor, and **Recommended Storage** indicates the supported storage capacity range. For a cluster with heavy read and write traffic, use large-capacity nodes as client nodes. Client nodes support EVS disks only. For more information about different node specifications, see **ECS Types**.
Node Storage Type and Capacity	Select an appropriate storage type and capacity for client nodes. • For more on EVS disk performance, see **Disk Types and Performance**. • The node storage capacity is fixed at 40 GB.
Nodes	The number of client nodes ranges from 1 to 64. For the recommended client node quantity, see **Recommended Client Node Quantity**.

Recommended Client Node Quantity

Client nodes route and coordinate search and index requests. Its quantity should be determined based on the cluster traffic. For your reference, the following table provides the recommended number of client nodes based on the number of data nodes.

Table 3-14 Recommended number of client nodes

Number of Data Nodes	Recommended Number of Client Nodes
< 5	Do not use client nodes.
≥ 5 and < 10	2

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Number of Data Nodes	Recommended Number of Client Nodes
≥ 10	Number of client nodes/Number of data nodes = 1:5

Suggestions on Cold Data Node Configuration

Table 3-15 Cold data node configuration

Parameter	Configuration Suggestions
Node Specifications	In the node flavor list, **vCPUs | Memory** indicate the number of vCPUs and memory capacity available for each flavor, and **Recommended Storage** indicates the supported storage capacity range. We recommend that you select node specifications based on service needs, such as the data volumes, performance requirements, and your spending budget. **Node Specifications** describes the application scenarios and core features of different node specifications. It can help you properly plan your cluster. For more information about different node specifications, see **ECS Types**.
Node Storage Type and Capacity	• If the selected node flavor uses EVS disks, you need to further select **Node Storage Type** and **Capacity** based on service requirements. – For more on EVS disk performance, see **Disk Types and Performance**. – The value range of node storage capacity is determined by the node flavor you select. The value must be divisible by 20. For how to calculate the required node storage capacity, see **Recommended Node Storage Capacity**. – Node storage capacity cannot be reduced once the cluster is created. Evaluate your long-term data needs and select an appropriate size. • If the selected node flavor uses local disks, there is no need to select the node storage type, and the node storage capacity is a fixed value. Both of them are determined by the local disk specifications.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Configuration Suggestions
Nodes	The value range for cold data nodes is 1 to 32. For how to calculate the number of nodes needed, see **Recommended Node Quantity**. • If the combined number of data nodes and cold data nodes exceeds 100, you are advised to select a flavor with at least 16 vCPUs. This helps prevent delays in cluster status synchronization across nodes and ensures optimal read and write performance. • If the number of cold data nodes in your cluster is not evenly divisible by the number of AZs, data distribution may become unbalanced across nodes. This will negatively impact both query and write performance.

3.3 Creating a Cluster (New Version)

You can use OpenSearch clusters created in CSS for purposes like log analytics, data search, real-time data analytics, and more.

You can create an OpenSearch cluster in either of the following ways:

- Method 1: Create an OpenSearch cluster on the CSS management console. This topic provides the operation guide for the default UI (that is, the new UI). For the operation guide for the old UI, see **Creating a Cluster**.

  NOTE

  To improve efficiency in creating clusters, CSS has made several enhancements on the cluster creation page in terms of ease-of-use. Being more intuitive, the new UI streamlines cluster creation procedures.

- Method 2: Create an OpenSearch cluster using CSS APIs. For details, see **Creating a Cluster**.

Prerequisites

- You have planned the OpenSearch cluster configuration by following the guidelines in **Cluster Planning**, including the cluster size, node configuration, and storage capacity requirements.
- You have created a VPC, subnet, and security group beforehand, or you can create them when creating the cluster. For more information, see Virtual Private Cloud User Guide.
- If you are going to enable automatic snapshot creation for the cluster, make sure you have created an OBS bucket for storing snapshots. The OBS bucket must be in the same region as the target OpenSearch cluster, and the storage class must be **Standard** or **Warm**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

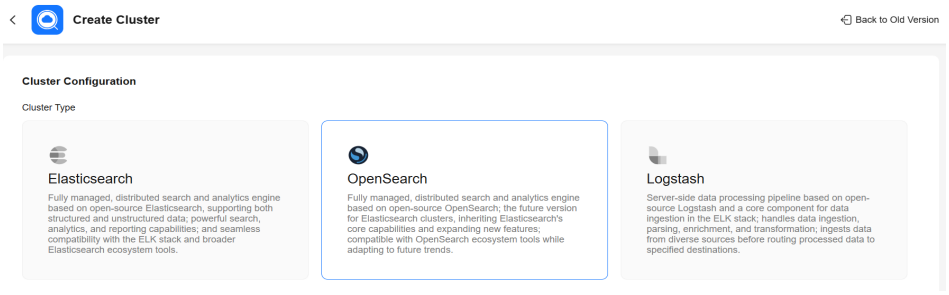
Impact on Billing

- If you enable automatic snapshot creation, automatically created snapshots will be stored in standard OBS buckets, and you will need to pay extra for the storage. For more information, see **Billing Overview**.
- VPC endpoints, if created for the cluster, will incur extra fees, depending on the resource usage. For details, see **VPC Endpoint Billing**.

Opening the Create Cluster Page

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the upper right corner, click **Create Cluster**. The new-version UI for creating a cluster is displayed by default.

Figure 3-2 Create Cluster (new version)



Cluster Configuration

Select the cluster type and version.

Table 3-16 Cluster configuration

Parameter	Description
Type	Choose **OpenSearch**.
Version	Select a cluster version from the drop-down list.

Basic Settings

Select the region and AZ of the cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-17 Basic settings

Parameter	Description
Region	Select the region where the cluster is located. A region is the location of a physical data center. Regions are defined based on their geographical location and network latency. For lower network latency and quicker resource access, select the nearest region.
AZ	Select one or more AZs associated with the cluster region. An AZ is a physical region where resources use independent power supplies and networks. AZs are physically isolated but interconnected through an internal network. A maximum of three AZs can be configured. For details about the use of multiple AZs, see **Suggestions on Multi-AZ Deployment**.

Data Nodes

Configure the cluster's data nodes.

Data nodes are primarily used to store the cluster's data. The functions of the data nodes vary depending on the node types enabled for the cluster.

- If the cluster has no client nodes, the data nodes will need to handle the additional task of query request distribution and data analysis.
- If the cluster has no master nodes, the data nodes will also need to handle cluster management.

Table 3-18 Configuring data nodes

Parameter	Description
CPU Architecture	Select the CPU architecture of the data nodes. **x86** nodes are supported. The architectures actually supported may vary depending on the regional environment.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Node Specifications	Select the specifications of the data nodes. Click **Available**. On the displayed page, select a flavor that suits your needs. The nodes may use either local or EVS disks. For more information about different node specifications, see **ECS Types**. In the node flavor list, **vCPUs | Memory** indicates the number of vCPUs and memory capacity available for each flavor, and **Recommended Storage** indicates the supported storage capacity range. The node flavors available may vary depending on the region you select. **WARNING** • Local disks are less reliable than EVS disks—if the underlying host fails, data stored on local disks may be permanently lost. For this reason, avoid using local disks to store mission-critical data. Instead, use EVS disks. • If you do use local disks, you are advised to enable automatic snapshot creation when creating the cluster, and also enable replicas when creating indexes. This helps lower the risk of data loss.
Node Storage Type and Capacity	Select the storage type and capacity of the data nodes. • If the selected node flavor uses EVS disks, you need to further select **Node Storage Type** and **Capacity** based on service requirements. – Available EVS disk types vary depending on your region. For more on EVS disk performance, see **Disk Types and Performance**. – The value range of node storage capacity is determined by the node flavor you select. The value must be divisible by 20. – Node storage capacity cannot be reduced once the cluster is created. Evaluate your long-term data needs and select an appropriate size. • If the selected node flavor uses local disks, there is no need to select the node storage type, and the node storage capacity is a fixed value. Both of them are determined by the local disk specifications.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Nodes	Set the number of data nodes for the cluster. • If master nodes are configured, the number of data nodes ranges from 1 to 200. • If no master nodes are configured, the number of data nodes ranges from 1 to 32. Configuration principles: • To ensure high service availability and prevent individual node overload, a single-AZ cluster **should** have at least three data nodes; a two-AZ cluster **must** have at least four data nodes (two in each AZ); and a three-AZ cluster **must** have at least three data nodes (one in each AZ). • For optimal performance, the number of data nodes should be an integer multiple of the number of AZs, for example, six data nodes for a three-AZ cluster. This helps prevent uneven data distribution and ensure optimal query and ingestion performance.

Master Nodes

Configure master nodes for the cluster. By default, master nodes are not enabled. To enable master nodes, select **Master Node**, and configure the node settings.

Master nodes manage cluster-wide operations, including metadata, indexes, and shard allocation. For large-scale deployments, using dedicated master nodes enhances cluster stability, service availability, and centralized control.

Table 3-19 Configuring master nodes

Parameter	Description
CPU Architecture	Select the CPU architecture of the master nodes. **x86** nodes are supported. The architectures actually supported may vary depending on the regional environment.
Node Specifications	Select the specifications of the master nodes. Click **Available**. On the displayed page, select a flavor that suits your needs. Master nodes support EVS disks only. For more information about different node specifications, see **ECS Types**. In the node flavor list, **vCPUs | Memory** indicate the number of vCPUs and memory capacity available for each flavor. The node flavors available may vary depending on the region you select.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Node Storage Type and Capacity	Select the storage type of the master nodes. • Available EVS disk types vary depending on your region. For more on EVS disk performance, see **Disk Types and Performance**. • The node storage capacity is fixed at 40 GB.
Nodes	Set the number of master nodes for the cluster. The value can be **3**, **5**, **7**, or **9**.

Client Nodes

Configure client nodes for the cluster. By default, client nodes are not enabled. To enable client nodes, select **Client Node**, and configure the node settings.

Client nodes route and coordinate search and index requests, offloading processing from data nodes for enhanced query performance and cluster scalability when there are heavy loads.

Table 3-20 Configuring client nodes

Parameter	Description
CPU Architecture	Select the CPU architecture of the client nodes. **x86** nodes are supported. The architectures actually supported may vary depending on the regional environment.
Node Specifications	Select the specifications of the client nodes. Click **Available**. On the displayed page, select a flavor that suits your needs. Client nodes support EVS disks only. For more information about different node specifications, see **ECS Types**. In the node flavor list, **vCPUs | Memory** indicate the number of vCPUs and memory capacity available for each flavor. The node flavors available may vary depending on the region you select.
Node Storage Type and Capacity	Select the storage type of the client nodes. • Available EVS disk types vary depending on your region. For more on EVS disk performance, see **Disk Types and Performance**. • The node storage capacity is fixed at 40 GB.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Nodes	Set the number of client nodes for the cluster. Value range: 1 to 64 Configuration principle: To ensure high service availability and prevent individual node overload in a multi-AZ deployment, the number of client nodes must be greater than or equal to the number of AZs. (For a cluster spanning two AZs, this means a minimum of two client nodes; for a three-AZ deployment, a minimum of three client nodes.)

Cold Data Nodes

Configure cold data nodes for the cluster. By default, it is disabled. To enable cold data nodes, select **Cold Data Node**, and configure the node settings.

Cold data nodes are used to store and query large quantities of latency-insensitive data. They offer an effective way to manage large datasets while cutting storage costs. When cold data nodes are enabled, you can switch between cold and hot data storage. For details, see **Switching Between Hot and Cold Data**. If there are no cold data nodes, we recommend that you use decoupled storage and compute, which can also cut storage costs. For details, see **Configuring Decoupled Storage and Compute**.

Table 3-21 Configuring cold data nodes

Parameter	Description
CPU Architecture	Select the CPU architecture of the cold data nodes. **x86** nodes are supported. The architectures actually supported may vary depending on the regional environment.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Node Specifications	Select the specifications of the cold data nodes. Click **Available**. On the displayed page, select a flavor that suits your needs. The nodes may use either local or EVS disks. For more information about different node specifications, see **ECS Types**. In the node flavor list, **vCPUs | Memory** indicates the number of vCPUs and memory capacity available for each flavor, and **Recommended Storage** indicates the supported storage capacity range. The node flavors available may vary depending on the region you select. **WARNING** • Local disks are less reliable than EVS disks—if the underlying host fails, data stored on local disks may be permanently lost. For this reason, avoid using local disks to store mission-critical data. Instead, use EVS disks. • If you do use local disks, you are advised to enable automatic snapshot creation when creating the cluster, and also enable replicas when creating indexes. This helps lower the risk of data loss.
Node Storage Type and Capacity	Select the storage type and capacity of the cold data nodes. • If the selected node flavor uses EVS disks, you need to further select **Node Storage Type** and **Capacity** based on service requirements. – Available EVS disk types vary depending on your region. For more on EVS disk performance, see **Disk Types and Performance**. – The value range of node storage capacity is determined by the node flavor you select. The value must be divisible by 20. – Node storage capacity cannot be reduced once the cluster is created. Evaluate your long-term data needs and select an appropriate size. • If the selected node flavor uses local disks, there is no need to select the node storage type, and the node storage capacity is a fixed value. Both of them are determined by the local disk specifications.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Nodes	Set the number of cold data nodes for the cluster. Value range: 1 to 32 Configuration principles: • To ensure high service availability and prevent individual node overload, a two-AZ cluster **must** have at least four cold data nodes (two in each AZ); and a three-AZ cluster **must** have at least three cold data nodes (one in each AZ). • To prevent uneven data distribution and ensure optimal query and ingestion performance, the number of cold data nodes should be an integer multiple of the number of AZs, for example, six cold data nodes for a three-AZ cluster.

Network Settings

Configure the VPC, subnet, IP addresses, and security group of the cluster.

Table 3-22 Network settings

Parameter	Description
VPC	Select a VPC for the cluster for proper network isolation. If none of the existing VPCs meets your requirements, click **Create VPC** to create one. (Make sure you have the permission to create VPCs.) For details, see **Creating a VPC and Subnet**.
Subnet	Select a subnet for the cluster. A subnet improves network security by providing exclusive network resources that are isolated from other networks. Select a subnet in the current VPC. If no existing subnets meet your requirements, click **Create Subnet** to create a new subnet. For details, see **Creating a VPC with a Subnet**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
IPv4 Address	Assign IPv4 addresses to cluster nodes. • **Assign automatically**: Private IPv4 addresses will be automatically assigned to cluster nodes. • **Assign manually**: Manually assign private IPv4 addresses to cluster nodes. Before assigning an IP address, click **View In-Use IP Address** to check IP addresses that are already in use. – If you manually specify fewer IP addresses than the total number of cluster nodes, the system will automatically assign random private IPv4 addresses to the remaining nodes. – Manually specified IP addresses will be assigned in the following priority order: client nodes > data nodes > cold data nodes > master nodes.
Security Group	Select a security group for the cluster. A security group serves as a virtual firewall that provides access control policies for clusters. To select a security group that meets your requirements, click **View Security Group** to go to the security group list, where you can check the details of each security group. The selected security group must allow port 9200 in the inbound direction. Otherwise, the cluster may be inaccessible to external services. Furthermore, to enable OpenSearch Dashboards private network access, port 5601 must also be allowed. If none of the existing security groups meets your requirements, click **Create Security Group** to create one. (Make sure you have the permission to create security groups.) For details, see **Creating a Security Group**.

Security Mode

Configure the cluster's security mode and access methods. The security mode is enabled by default when you create a cluster. To disable it, deselect **Security Mode**.

Configure the security mode based on the security needs of your cluster.

- **Disabling the security mode**: Use when creating a cluster for internal testing or workloads that have a low security standard. When the security mode is disabled, HTTP is used, data is transmitted in plaintext, and no user authentication is required for cluster access. Make sure the cluster is deployed in a secure environment. Do not expose the cluster's network interface to the public network.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- **Security mode + HTTP**: Use to balance security and performance. In this setting, user authentication is required for cluster access, yet data is transmitted via HTTP.
- **Security mode + HTTPS**: Use when security takes precedence over performance. In this setting, user authentication is required for cluster access, and data is transmitted via HTTPS in ciphertext.

CAUTION

HTTPS, public network access, and OpenSearch Dashboards public network access can be enabled only when the security mode is enabled.

Table 3-23 Configuring the security mode

Parameter	Description
Administrator account and password	When the security mode is enabled, cluster access requires user authentication. In this case, the administrator password must be configured. • **Administrator**: The default value is **admin** and cannot be changed. • **Administrator Password**: Set and confirm the administrator password. This password will be required when you access this cluster.
HTTPS Access	Whether to enable HTTPS access for the cluster. Enabled by default, HTTPS improves security by encrypting all communication with the cluster. To use HTTP instead, deselect **HTTPS Access**. Data security cannot be guaranteed when HTTPS is disabled. Exercise caution. HTTPS can be disabled only for OpenSearch 1.3.6, 2.19.0, and 3.4.0 clusters for which the security mode is enabled. For other versions, HTTPS access cannot be disabled. When HTTPS is used, data encryption and decryption introduce computational overhead, which may impact the cluster's read and write performance.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Public Network Access	Whether to enable public network access. By default, it is disabled. To enable it, select **Public Network Access**, and configure the necessary settings. Public network access can be enabled only if the security mode and HTTPS are both enabled for the cluster. When public network access is enabled, the cluster is automatically assigned a public IP address with dedicated dynamic BGP bandwidth. When **Public Network Access** is selected, further configure the following parameters: • **Bandwidth**: Set the bandwidth for public network access. The value ranges from 1 Mbit/s to 200 Mbit/s. • **Configure Whitelist** Control public network access to the cluster using a whitelist. – If a whitelist is configured, only IP addresses that are on this whitelist can access the cluster over the public network. Click **+Add**. In the displayed text box, enter IP addresses or CIDR blocks that are allowed to access the cluster from the public network. Separate them using commas (,). Each value must be unique. An example of valid values: **192.168.1.1,10.0.0.0/24**. Examples of invalid values: **0.0.0.0**, **xx.xx.xx.xx/0**, **172.16.0.0-172.16.255.255**, non-standard formats (e.g., **192.168.1**), and duplicate values. – If no whitelist is configured, all public IP addresses can access the cluster. However, this can be a security risk and should be avoided.

Cluster Management

Configure the cluster name, enterprise project, and more.

Table 3-24 Cluster management information

Parameter	Description
Cluster Name	User-defined cluster name. The cluster name must start with a letter and can contain 4 to 32 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Add Description	Add a description for the cluster for easy recognition. Click **+ Add Description** and then enter a description of the cluster. The value can contain 0 to 128 characters.
Enterprise Project	Associate the cluster with an enterprise project. An enterprise project groups cloud resources, so you can manage resources and members by project. The default project is **default**. If enterprise projects are enabled, you can select an enterprise project from the drop-down list. You can click **View Enterprise Projects** to go to the Enterprise Project Management Service console and check enterprise project details. For more information, see **Accessing the Enterprise Center**.
Tags	Adding tags to clusters helps you identify and manage your cluster resources. Click **+ Add**. You can select existing tag keys and values from the drop-down lists or enter new ones. If you want to use the same tag to identify multiple cloud resources for better resource grouping, we recommend that you predefine tags in Tag Management Service (TMS). For details, see **Predefined Tags**. Each cluster can have a maximum of 20 tags. If your organization has configured tag policies for CSS, add tags to clusters based on these policies. Tags that do not comply with predefined tag policies will cause a cluster creation failure. Contact the administrator to learn about the tag policies.

Automatic Snapshot Creation

Expand **More Settings** and configure automatic snapshot creation. Automatic snapshot creation is disabled by default. To enable it, select **Automatic Snapshot Creation**, and configure the necessary settings. You are advised to enable automatic snapshot creation to have data backed up periodically, thus improving cluster reliability. If you choose not to enable it, enable replicas when creating indexes, which helps prevent data loss.

CAUTION

If you enable automatic snapshot creation, automatically created snapshots will be stored in standard OBS buckets, and you will need to pay extra for the storage. For more information, see **Billing Overview**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-25 Configuring automatic snapshot creation

Parameter	Description
OBS Bucket	From the drop-down list, select an OBS bucket for storing snapshots. If no OBS buckets meet your requirements, click **Create Bucket** to go to the OBS console and create one. For details, see **Creating a Bucket**. An IAM user must be granted the following OBS permissions to view available OBS buckets: obs:bucket:getBucketStoragePolicy obs:bucket:listAllMyBuckets
Backup Path	Snapshot storage path in the OBS bucket. The backup path cannot: • Contain the following characters: **\:*?"<>|'{}** • Start with a slash (/). • Start or end with a period (.). • Contain more than two consecutive slashes (/) or periods (.). • Exceed 512 characters. **CAUTION** Only standard OBS storage can be used to store snapshots. Do not apply OBS lifecycle rules to the backup path.
Maximum Backup Rate (per Second)	The parameter sets the maximum backup rate per node. When it is exceeded, flow control is triggered to prevent excessive resource usage and ensure system stability. **Value format**: number (0–9999) + unit (KB, MB, GB, TB, PB, or B) **Default value**: 40 MB The value 0MB means there is no limit on how fast data is backed up to snapshots. An overly high backup rate may lead to excessive resource usage, which may impact cluster stability. Configure this parameter carefully to maintain optimal performance. The actual backup rate may not reach the configured value, as it depends on factors such as OBS performance and disk I/O.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description

Maximum Recovery Rate (per Second)	The parameter sets the maximum recovery rate per node. When it is exceeded, flow control is triggered to prevent excessive resource usage and ensure system stability.
	Value format: number (0–9999) + unit (KB, MB, GB, TB, PB, or B)
	The default value is 0MB, indicating no limit.
	An overly high recovery rate may lead to excessive resource usage, which may impact cluster stability. Configure this parameter carefully to maintain optimal performance.
	The actual recovery rate is also affected by the value of **indices.recovery.max_bytes_per_sec**. Whichever is smaller between the two parameters will be used. Additionally, the actual recovery rate may not reach the configured value, as it depends on factors such as OBS performance and disk I/O.
	NOTE
	• To check the value of **indices.recovery.max_bytes_per_sec**, run the following command:
	`GET _cluster/settings`
	• To modify **indices.recovery.max_bytes_per_sec**, run the following command:
	`PUT _cluster/settings`
	`{`
	`  "transient": {`
	`  "indices.recovery.max_bytes_per_sec": "100mb"`
	`  }`
	`}`

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
IAM Agency	Select an IAM agency to grant the current account the permission to access and use OBS. To store snapshots to an OBS bucket, you must have the required OBS access permissions. • If you are configuring an agency for the first time, click **Automatically Create IAM Agency** to create **css-obs-agency**. • If there is an IAM agency automatically created earlier, you can click **One-click authorization** to have the OBS Administrator permissions deleted automatically, and have the following custom policies added automatically instead to implement more refined permissions control. "obs:bucket:GetBucketLocation", "obs:object:GetObjectVersion", "obs:object:GetObject", "obs:object:DeleteObject", "obs:bucket:HeadBucket", "obs:bucket:GetBucketStoragePolicy", "obs:object:DeleteObjectVersion", "obs:bucket:ListBucketVersions", "obs:bucket:ListBucket", "obs:object:PutObject" • When OBS buckets use SSE-KMS encryption, the IAM agency must be granted KMS permissions. You can click **Automatically Create IAM Agency** and **One-click authorization** to have the following custom policies created automatically. "kms:cmk:create", "kms:dek:create", "kms:cmk:get", "kms:dek:decrypt", "kms:cmk:list" • To use **Automatically Create IAM Agency** and **One-click authorization**, the following minimum permissions are required: "iam:agencies:listAgencies", "iam:roles:listRoles", "iam:agencies:getAgency", "iam:agencies:createAgency", "iam:permissions:listRolesForAgency", "iam:permissions:grantRoleToAgency", "iam:permissions:listRolesForAgencyOnProject", "iam:permissions:revokeRoleFromAgency", "iam:roles:createRole" • To use an IAM agency, the following minimum permissions are required: "iam:agencies:listAgencies", "iam:agencies:getAgency", "iam:permissions:listRolesForAgencyOnProject", "iam:permissions:listRolesForAgency" **WARNING** The agency name can contain only letters (case-sensitive), digits, underscores (_), and hyphens (-). Otherwise, the backup will fail.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Snapshot Name Prefix	A snapshot name consists of the snapshot name prefix and timestamp, for example, *snapshot-1566921603720*. The snapshot name prefix can contain 1 to 32 characters. It must start with a lowercase letter, and it can only contain lowercase letters, digits, hyphens (-), and underscores (_). The default value is **snapshot**.
Time Zone	Select a time zone for the backup start time.
Backup Start Time	Specify the start time of auto backup. Select a value from the drop-down list. The interval can be **Daily**, **Hourly**, or weekly (by selecting a specific day of the week), and the backup time can be set to any hour from 00:00 to 23:00 (full hours only).
Retained Snapshots	Number of automatically created snapshots to be retained. The value ranges from 1 to 90. The system automatically deletes excess snapshots every half hour. (The expiration deletion policy applies only to the snapshots that were automatically created at the same frequency as the current automatic snapshot creation policy.) **NOTE** If the snapshot creation interval is short or if the data size of indexes is large, the number of automatic snapshots retained may not reach the value set using this parameter.

VPC Endpoint

Expand **More Settings** and configure a VPC endpoint for the cluster. VPC endpoint is disabled by default. To enable it, select **VPC Endpoint**, and configure the necessary settings.

VPC Endpoint enables you to access resources across VPCs using a dedicated gateway without exposing the network information of servers. The address of a VPC endpoint consists of an IPv4 address and private domain name. When you access a cluster through the IP address or private domain name of a VPC endpoint, you are not subject to the cluster's security group rules, and you can configure a whitelist for access control.

CAUTION

- VPC endpoints, if created for the cluster, will incur extra fees, depending on the resource usage. For details, see **VPC Endpoint Billing**.
- You need specific permissions to create VPC endpoints. For details, see **VPCEP Permissions**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-26 Configuring VPC Endpoints

Parameter	Description
Create Private Domain Name	Whether to create a private domain name for the VPC endpoint. • Enable: The system automatically assigns a private domain name to the VPC endpoint. After cluster creation, you can check this private domain name on the **VPC Endpoint** tab of the cluster details page. • Disable: No private domain name will be configured for the VPC endpoint. The cluster can only be accessed through an IP address assigned to the VPC endpoint.
Configure Whitelist	Configure access control for a VPC endpoint. Add accounts that are allowed to access the cluster through VPC Endpoint. Click **Add** to add accounts in **Account ID**. • To obtain your authorized account ID, point to your username in the upper right corner, and choose **My Credentials**. Copy the value of **Account ID**. • If no account is added or the account ID is set to *, all users are allowed to access the cluster.

OpenSearch Dashboards Public Network Access

Expand **More Settings** and configure OpenSearch Dashboards public network access for the cluster. OpenSearch Dashboards public network access is disabled by default. To enable it, select **OpenSearch Dashboards Public Network Access**, and configure the necessary settings.

When OpenSearch Dashboards public network access is enabled, a public IP address will be automatically assigned for OpenSearch Dashboards access. Users will be able to access this OpenSearch cluster's OpenSearch Dashboards console from the public network through this address.

CAUTION

This parameter is available only when the security mode is enabled for the cluster.

Table 3-27 Configuring public network access for OpenSearch Dashboards

Parameter	Description
Bandwidth	Bandwidth for accessing OpenSearch Dashboards from the public network. Value range: 1 Mbit/s to 200 Mbit/s

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Configure Whitelist	Control OpenSearch Dashboards public network access using a whitelist. • If a whitelist is configured, only IP addresses that are on this whitelist can access the cluster's OpenSearch Dashboards over the public network. Click **+ Add**. In the displayed text box, enter IP addresses or CIDR blocks that are allowed to access the cluster's OpenSearch Dashboards console from the public network. Separate them using commas (,). Each value must be unique. An example of valid values: **192.168.1.1,10.0.0.0/24**. Examples of invalid values: **0.0.0.0**, **xx.xx.xx.xx/0**, **172.16.0.0-172.16.255.255**, non-standard formats (e.g., **192.168.1**), and duplicate values. • If no whitelist is configured, all public IP addresses can access the cluster's OpenSearch Dashboards console. However, this can be a security risk and should be avoided.

Submit

1. In the **Configuration Summary** panel, review the cluster configuration details.

   Any mandatory fields that are missing are displayed in red, and you have to set them in the configuration area.

2. Click **Create Now**.

   Cluster creation time depends on the number of nodes. Clusters with a large number of nodes typically take longer to create.

3. Return to the cluster list and check the newly created cluster.

   If the cluster is created successfully, **Cluster Status** changes to **Available**.

3.4 Creating a Cluster

This topic describes how to create an OpenSearch cluster. You can create an OpenSearch cluster in either of the following ways:

- Method 1: Create an OpenSearch cluster on the CSS management console.
- Method 2: Create an OpenSearch cluster using CSS APIs. For details, see **Creating a Cluster**.

Scenarios

Table 3-28 lists key parameters that differentiate between different types of clusters.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-28 Parameters that differentiate between different types of clusters

Cluster Type	Security Mode	HTTPS Access	Internet Access	OpenSearch Dashboards Public Network Access
Cluster with the security mode disabled	Disabled	N/A	Cannot be enabled	Cannot be enabled
Cluster in security mode + HTTP	Enabled	Disabled	Cannot be enabled	Can be enabled
Cluster in security mode + HTTPS	Enabled	Enabled	Can be enabled	Can be enabled

Prerequisites

You have planned the OpenSearch cluster configuration by following the instructions in **Cluster Planning**.

Creating a Cluster

1. Log in to the **CSS management console**.
2. On the **Dashboard** page, click **Create Cluster** in the upper right corner. The **Create Cluster** page is displayed.

   Alternatively, choose **Clusters** > **OpenSearch** in the navigation tree on the left. Click **Create Cluster** in the upper right corner. The **Create Cluster** page is displayed.
3. On the **Basic Configuration** page, configure basic information and resources for the OpenSearch cluster.

   Table 3-29 Basic settings of the OpenSearch cluster

Parameter	Description
Region	Select the region where the cluster is located. Resources in different regions cannot communicate with each other over an internal network. For lower network latency and quicker resource access, select the nearest region.
AZ	Select one or more AZs associated with the cluster region. A maximum of three AZs can be configured. For details about the use of multiple AZs, see **Suggestions on Multi-AZ Deployment**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Type	Choose **OpenSearch**.
Version	Select a cluster version from the drop-down list.
Name	User-defined cluster name. The cluster name must start with a letter and can contain 4 to 32 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
Cluster Description	Add a description for the cluster for easy recognition. The value can contain 0 to 128 characters.
Nodes	Number of nodes in the cluster. Data nodes store the cluster's data. If there are no client nodes in the cluster, data nodes will need to additionally handle cluster access and data analytics requests; if there are no master nodes, data nodes will also need to provide cluster management. • If master nodes are configured, the number of data nodes ranges from 1 to 200. • If no master nodes are configured, the number of data nodes ranges from 1 to 32. Principles: • To ensure high service availability and prevent individual node overload, a single-AZ cluster **should** have at least three data nodes; a two-AZ cluster **must** have at least four data nodes (two in each AZ); and a three-AZ cluster **must** have at least three data nodes (one in each AZ). • For optimal performance, the number of data nodes should be an integer multiple of the number of AZs, for example, six data nodes for a three-AZ cluster. This helps prevent uneven data distribution and ensure optimal query and ingestion performance.
CPU Architecture	**x86**. The supported types depend on the actual regional environment.
Node Specifications	Data node flavor. You can select a node flavor based on your needs. Each cluster supports only one flavor. For details, see **ECS Types**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Node Storage Type	If you select EVS for node storage, you need to further select the EVS disk type for data nodes of the cluster. Supported node storage types: Common I/O, High I/O, Ultra-high I/O. **NOTE** If the type of storage in use is not supported, the storage type is not displayed.
Node Storage Capacity	Data node storage capacity. Its value range varies with node specifications. The node storage capacity must be a multiple of 20.
Master node	The master node is responsible for important cluster management tasks, such as metadata management, index creation and deletion, and shard allocation. It plays a critical role in metadata management, node management, stability guarantee, and cluster operation control for large-scale clusters. After enabling the master node, specify **Node Specifications**, **Nodes**, and **Node Storage Type**. The value of **Nodes** must be an odd number greater than or equal to 3. Up to nine nodes are supported. The value of **Node Storage Capacity** is fixed. You can select a storage type based on your needs.
Client node	Client nodes receive and coordinate external requests, such as search and write requests. They play an important role in handling high-load queries, complex aggregations, managing a large number of shards, and improving cluster scalability. After enabling the client node, specify **Node Specifications**, **Nodes** and **Node Storage Type**. The value of **Nodes** ranges from 1 to 64. The value of **Node Storage Capacity** is fixed. You can select a storage type based on your needs. To ensure high service availability and prevent individual node overload in a multi-AZ deployment, the number of client nodes should be greater than or equal to the number of AZs. (For a cluster spanning two AZs, this means a minimum of two client nodes; for a three-AZ deployment, a minimum of three client nodes.)

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Cold data node	Cold data nodes are used to store query latency-insensitive data in large quantities. They offer an effective way to manage large datasets and cut storage costs. After enabling cold data nodes, configure **Node Specifications**, **Nodes**, **Node Storage Type**, and **Node Storage Capacity**. The value of **Nodes** ranges from 1 to 32. Select **Node Storage Type** and **Node Storage Capacity** as required. • To ensure high service availability and prevent individual node overload, a two-AZ cluster **must** have at least four cold data nodes (two in each AZ); and a three-AZ cluster **must** have at least three cold data nodes (one in each AZ). • To prevent uneven data distribution and ensure optimal query and ingestion performance, the number of cold data nodes should be an integer multiple of the number of AZs, for example, six cold data nodes for a three-AZ cluster. When cold data nodes are enabled, you can switch between cold and hot data nodes. For details, see **Switching Between Hot and Cold Data**.
Enterprise Project	When creating a CSS cluster, you can bind an enterprise project to the cluster if you have enabled the enterprise project function. Select an enterprise project from the **Enterprise Project** drop-down list, or click **View Enterprise Project** to go to the **Enterprise Project Management Service** page and check existing enterprise projects.

4. Click **Next: Network**.
5. On the **Network** page, configure the network settings and security mode for the OpenSearch cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-30 Network settings for the OpenSearch cluster

Parameter	Description
VPC	Specify a VPC to isolate the cluster's network. Click **View VPC** to go to the VPC management console and check the created VPCs. If no VPC is available, contact the CSS administrator to create a new VPC. For details, see **Creating a VPC with a Subnet**. **NOTE** The 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, improving network security. Select a subnet needed by the cluster in the current VPC. If no existing subnets meet your requirements, click **Create Subnet** to create a new subnet. For details, see **Creating a VPC and Subnet**.
Security Group	A security group serves as a virtual firewall that provides access control policies for clusters. Select a security group for the cluster. Click **View Security Group** to go to the security group list, where you can view details about security groups. **NOTE**
Security Mode	Whether to enable the security mode for the cluster. • The security mode is enabled by default. In security mode, a cluster's communication is encrypted and access to the cluster requires user authentication. This is why **Administrator Username** and **Administrator Password** must be configured for the cluster. – The default administrator username is **admin**. – Set and confirm the **Administrator Password**. This password will be required when you access this cluster. • If **Security Mode** is disabled, a cluster in non-security mode will be created. With such a cluster, access to the cluster will require no user authentication, and data will be transmitted in plaintext using HTTP. Make sure the cluster is deployed in a secure environment. Do not expose the cluster's network interface to the public network.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
HTTPS Access	HTTPS access can be enabled only when security mode is enabled for the cluster. With HTTPS access enabled, communication will be encrypted when you access the cluster. HTTPS can be disabled only for OpenSearch 1.3.6, 2.19.0, and 3.4.0 clusters for which the security mode is enabled. For other versions, HTTPS access cannot be disabled. **NOTE** Compared with a non-security mode cluster that uses HTTP, a security-mode cluster that uses HTTPS has lower read performance. The performance loss is estimated at around 20% under high concurrency. If you want fast read performance as well as the isolation and permission control (such as indexes, documents, and fields) enabled by the security mode, you can disable **HTTPS Access**. After **HTTPS Access** is disabled, HTTP protocol is used for cluster communication. In this case, data security cannot be guaranteed and public network access is disabled.
Public IP Address	This parameter is available only when **Security Mode** and **HTTPS Access** are enabled. When **Public IP Address** is enabled, a public IP address is automatically assigned, which will enable access to the security cluster from the Internet. For details, see **Configuring Public Network Access**.

6. Click **Next: Advanced Settings**.
7. On the **Advanced Settings** page, configure a snapshot policy and other advanced settings for the OpenSearch cluster.
   a. Set a cluster snapshot policy.

      Cluster snapshots are disabled by default. To enable them, toggle on **Cluster Snapshots**. To store automatically created snapshots in OBS, an agency will need to be created in order to access OBS. Fees will be incurred for using OBS storage (storage class: Standard).

      Table 3-31 Basic configuration for a cluster snapshot policy

Parameter	Description
OBS Bucket	From the drop-down list, select an OBS bucket for storing snapshots. You can also click **Create Bucket** on the right to create a new OBS bucket. For details, see **Creating a Bucket**. The newly created or existing OBS bucket must meet the following requirements: • **Storage Class**: **Standard**. • **Region** must be the same as that of the created cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Backup Path	Snapshot storage path in the OBS bucket. The backup path cannot: ● Contain the following characters: \:*?"<>|'{} ● Start with a slash (/). ● Start or end with a period (.). ● Contain more than two consecutive slashes (/) or periods (.). ● Exceed 512 characters. **CAUTION** Only standard OBS storage can be used to store snapshots. Do not apply OBS lifecycle rules to the backup path.
Maximum Backup Rate (per Second)	The parameter sets the maximum backup rate per node. When it is exceeded, flow control is triggered to prevent excessive resource usage and ensure system stability. **Value format**: number (0–9999) + unit (KB, MB, GB, TB, PB, or B) **Default value**: 40 MB The value 0MB means there is no limit on how fast data is backed up to snapshots. An overly high backup rate may lead to excessive resource usage, which may impact cluster stability. Configure this parameter carefully to maintain optimal performance. The actual backup rate may not reach the configured value, as it depends on factors such as OBS performance and disk I/O.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description

Maximum Recovery Rate (per Second)	The parameter sets the maximum recovery rate per node. When it is exceeded, flow control is triggered to prevent excessive resource usage and ensure system stability.
	Value format: number (0–9999) + unit (KB, MB, GB, TB, PB, or B)
	The default value is 0MB, indicating no limit.
	An overly high recovery rate may lead to excessive resource usage, which may impact cluster stability. Configure this parameter carefully to maintain optimal performance.
	The actual recovery rate is also affected by the value of **indices.recovery.max_bytes_per_sec**. Whichever is smaller between the two parameters will be used. Additionally, the actual recovery rate may not reach the configured value, as it depends on factors such as OBS performance and disk I/O.
	NOTE
	• To check the value of **indices.recovery.max_bytes_per_sec**, run the following command:
	`GET _cluster/settings`
	• To modify **indices.recovery.max_bytes_per_sec**, run the following command:
	`PUT _cluster/settings`
	`{`
	`  "transient": {`
	`  "indices.recovery.max_bytes_per_sec": "100mb"`
	`  }`
	`}`

IAM Agency	IAM agency authorized by the current account for CSS to access or maintain data stored in OBS. If no agency is available, contact the CSS administrator to create one. For details, see **Creating an Agency**. The selected IAM agency must meet the following requirements: • **Agency Type** must be **Cloud service**. • Set **Cloud Service** to **Elasticsearch** or **CSS**. • Mandatory policies: **Tenant Administrator** or **OBS Administrator**. **WARNING** The agency name can contain only letters (case-sensitive), digits, underscores (_), and hyphens (-). Otherwise, the backup will fail.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-32 Setting automatic snapshot creation

Parameter	Description
Snapshot Name Prefix	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. A snapshot name consists of a snapshot name prefix and a timestamp, for example, **snapshot-1566921603720**.
Time Zone	Time zone for the backup time. Specify **Backup Started Time** based on the time zone.
Backup Start Time	Specify the start time of auto backup. Select a value from the drop-down list. The interval can be **Daily**, **Hourly**, or weekly (by selecting a specific day of the week), and the backup time can be set to any hour from 00:00 to 23:00 (full hours only).

b. Configure advanced settings for the cluster.

Tags

Adding tags to clusters helps you identify and manage your cluster resources. You can customize tags or use tags predefined by Tag Management Service (TMS). If you want to use the same tag to identify multiple cloud resources for better resource grouping, we recommend that you predefine tags in Tag Management Service (TMS). For details, see **Predefined Tags Overview**.

NOTE

If your organization has configured tag policies for CSS, add tags to clusters based on these policies. Tags that do not comply with predefined tag policies will cause a cluster creation failure. Contact the administrator to learn about the tag policies.

8. Click **Next: Confirm Configuration**. Check the configuration and click **Next** to create a cluster.
9. Return to the cluster list and check the newly created cluster. If the cluster is created successfully, **Cluster Status** changes to **Available**. Cluster creation time depends on the number of nodes. Typically, this process takes less than 60 minutes, though clusters with a large number of nodes may require additional time.

   If cluster creation fails, try creating the cluster again by rectifying the errors returned.

3.5 Network Settings

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3.5.1 Overview

This topic introduces the four networking modes supported by CSS OpenSearch clusters. Choose the configuration that best suits your connectivity needs, for example, whether you require intra-VPC, cross-VPC, or public network access.

Comparing different networking modes

Table 3-33 Comparing different networking modes

Dimension	Private Network Access	Public Network Access	Dedicated Load Balancer	VPC Endpoint
Intra-VPC access	Supported	Supported	Supported	Supported
Cross-VPC access	VPC peering connections and cluster routes required	Supported (automatic traversal)	VPC peering connections and cluster routes required	Supported (automatic traversal)
Internet accessibility	No	Yes	Depends on whether a public IP address is allocated to the load balancer.	No
Security	VPC isolation	Public network access control policy	ELB access control policy	VPCEP access control policy
Data transfer costs	N/A	Public network bandwidth fees	ELB resource fees	VPCEP resource fees
Advantage	Low latency, low cost, and high security	All-domain Internet accessibility	Powerful load balancing, two-way authentication via custom certificates	Private connections, network topology decoupled
When to Use	Internal service systems on the cloud	Ad hoc testing, public network services	High availability + load balancing	Cross-account/ Cross-VPC services

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuration Guide	(Optional) **Configuring Routes for a Cluster**	**Configuring Public Network Access**	• **Configuring a Dedicated Load Balancer** • (Optional) **Configuring Routes for a Cluster**	**Configuring VPC Endpoints**

Choosing a Networking Mode

1. Do clients access our service from a public network environment?
   - Yes: Use **Dedicated Load Balancer**, but the load balancers must be allocated public IP addresses.
   - No: Go to 2.
2. Since the clients are on the internal network, do they share the same VPC as our cluster?
   - Yes: This is intra-VPC access. Does our cluster need load balancers to enhance throughput and stability?
     - Yes: Use **Dedicated Load Balancer**.
     - No: Use **Private Network Access**.
   - No: This is cross-VPC access. Do the VPCs have overlapping CIDR blocks?
     - Yes: Use **VPC Endpoint**.
     - No: Use **Dedicated Load Balancer**. VPC peering connections or cluster routes are required.

3.5.2 Configuring Public Network Access

For enterprise applications, to make CSS OpenSearch clusters securely accessible from the Internet, you can enable public network access for them. When public network access is enabled for a cluster, the cluster is automatically assigned a public IP address with dedicated dynamic BGP bandwidth, making it accessible from the Internet via HTTPS. You can configure access control from the public network by IP addresses or IP address ranges.

When enabling public network access for an OpenSearch cluster, a shared load balancer is typically used for load balancing. If your workloads require quicker access, you are advised to use a dedicated load balancer to connect your cluster. For details, see **Configuring a Dedicated Load Balancer**.

Impact on Billing

When public network access is enabled for an OpenSearch cluster, the cluster is automatically assigned an EIP and public network bandwidth, which will both incur expenses.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Constraints

- Public network access can be enabled for an OpenSearch cluster only if **Security Mode** and **HTTPS Access** are both enabled.
- Public network access and VPC Endpoint share the same shared load balancer. If you configure a whitelist for public network access control, this whitelist is deployed to the shared load balancer. As such, it will control access not only from the public network but also from private IP addresses through VPCEP. In this case, you need to add IP address **198.19.128.0/17** to the public access control whitelist to allow traffic through VPCEP.

Enabling Public Network Access

To enable public network access for an existing cluster, perform the following steps:

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, go to the **Network Information** area, and check whether **Security Mode** and **HTTPS Access** are enabled.
   - If they are enabled, go to the next step to enable public network access.
   - If either one is disabled, public network access cannot be enabled for the cluster.

   Figure 3-3 Checking whether public network access can be enabled for the cluster

5. Click **Enable** next to **Public Network Access**. In the displayed dialog box, configure the necessary settings.

   Table 3-34 Enabling public network access

Parameter	Description
Bandwidth	Cluster bandwidth for public network access. Value range: 1 Mbit/s to 200 Mbit/s

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Configure Whitelist	Control public network access to the cluster using a whitelist. • If a whitelist is configured, only IP addresses that are on this whitelist can access the cluster over the public network. Click **+Add**. In the displayed text box, enter IP addresses or CIDR blocks that are allowed to access the cluster from the public network. Separate them using commas (,). Each value must be unique. An example of valid values: **192.168.1.1,10.0.0.0/24**. Examples of invalid values: **0.0.0.0**, **xx.xx.xx.xx/0**, **172.16.0.0-172.16.255.255**, non-standard formats (e.g., **192.168.1**), and duplicate values. • If no whitelist is configured, all public IP addresses can access the cluster. However, this can be a security risk and should be avoided.

6. Click **OK** to enable public network access.

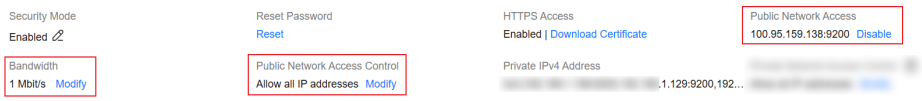
   After public network access is enabled, the **Network Information** area shows the public IP address, public network access control, and bandwidth information.

Managing Public Network Access

When public network access is enabled, you can check the public IP address, and modify the bandwidth and access control settings.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, manage public network access settings in the **Network Information** area.

   Figure 3-4 Managing public network access

   - **Checking the public IP address**

     Record the IP address and port under **Public Network Access** for later use.

   - **Modifying public network access control settings**

     Click **Modify** next to **Public Network Access Control**. In the displayed dialog box, add or remove IP addresses or CIDR blocks to or from the whitelist. Click **OK** to save the change.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- **Modifying public network bandwidth**

  Click **Modify** under **Bandwidth**. In the displayed dialog box, change the bandwidth for optimal network performance. Click **OK** to save the change.

Disabling Public Network Access

If public network access is no longer required for a cluster, disable it to release resources.

WARNING

After the public IP address is disassociated, the cluster can no longer be accessed from the Internet through this IP address. If you disable public network access for a cluster and then re-enable it, the public IP address for accessing the cluster may change. Exercise caution.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, find **Public Network Access** in the **Network Information** area, and click **Disable** under it. In the displayed dialog box, enter **CONFIRM** and click **OK**.

   After public network access is disabled, the **Network Information** area no longer shows the public IP address, public network access control, or bandwidth information.

Accessing a Cluster Through a Public IP Address

After public network access is enabled, the cluster is assigned a public IP address. External users can securely access this cluster through this IP address plus a port number.

For example, if the public IP address is **10.62.xxx.xxx** and the port number is **9200**, run the following cURL command to view indexes in the cluster.

```
curl -u username:password -k 'https://10.62.xxx.xxx:9200/_cat/indices'
```

where, **username** and **password** indicate the username and password of the HTTPS-enabled security-mode cluster.

3.5.3 Configuring a Dedicated Load Balancer

In big data processing scenarios, load balancers are essential for distributing traffic and improving the availability and performance of OpenSearch clusters. However, default shared load balancers have functional and performance limitations, making them inadequate for complex, high-concurrency data processing. Dedicated load balancers, with their richer features and superior performance, offer a more robust solution. This topic describes how to configure dedicated load balancers for more efficient and secure cluster access.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Overview

Advantages of connecting to a cluster through a dedicated load balancer:

- A non-security cluster can also utilize the capabilities of the Elastic Load Balance (ELB) service.
- You can use custom certificates for HTTPS two-way authentication.
- Seven-layer traffic monitoring and alarm configuration are supported, allowing you to keep close track of the cluster status.

There are eight different ELB service forms for clusters in different security modes to connect to a dedicated load balancer. **Table 3-35** describes the ELB capabilities for different cluster configurations. **Table 3-36** describes the configurations for different ELB service forms.

Table 3-35 ELB capabilities for different clusters

Security Mode	Service Form Provided by ELB for External Systems	ELB Load Balancing	ELB Traffic Monitoring	ELB Two-way Authentication
Non-security mode	No authentication	Supported	Supported	Not supported
	One-way authentication Two-way authentication	Supported	Supported	Supported
Security mode + HTTP	Password authentication	Supported	Supported	Not supported
	One-way authentication + Password authentication Two-way authentication + Password authentication	Supported	Supported	Supported
Security mode + HTTPS	One-way authentication + Password authentication Two-way authentication + Password authentication	Supported	Supported	Supported

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-36 Configurations for different ELB service forms depending on the cluster

Security Mode	Service Form Provided by ELB for External Systems	ELB Listener	ELB Listener	ELB Listener	Backend Server Group	Backend Server Group	Backend Server Group
		Frontend Protocol	Frontend Port	SSL Authentication	Backend Protocol	Health Check Port	Health Check Path
Non-security mode	No authentication	HTTP	9200	No authentication	HTTP	9200	/
	One-way authentication	HTTPS	9200	One-way authentication	HTTP	9200	
	Two-way authentication	HTTPS	9200	Two-way authentication	HTTP	9200	
Security mode + HTTP	Password authentication	HTTP	9200	No authentication	HTTP	9200	/_opendistro/_security/health
	One-way authentication + Password authentication	HTTPS	9200	One-way authentication	HTTP	9200	
	Two-way authentication + Password authentication	HTTPS	9200	Two-way authentication	HTTP	9200	
Security mode + HTTPS	One-way authentication + Password authentication	HTTPS	9200	One-way authentication	HTTPS	9200	
	Two-way authentication + Password authentication	HTTPS	9200	Two-way authentication	HTTPS	9200	

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Constraints

- You are not advised to connect a load balancer that has been associated with a public IP address to a non-security mode cluster. Allowing public network access through such a load balancer may cause security risks because a non-security mode cluster can be accessed using HTTP without security authentication.
- HTTPS-enabled security-mode clusters do not support HTTP-based frontend authentication. If the frontend uses HTTP, disable security mode for your cluster first. For details, see **Changing the Security Mode**. Before changing the security mode, disable load balancing first. After the security mode is changed, enable load balancing again.

Prerequisites

- A dedicated load balancer has been created. For details, see **Creating a Dedicated Load Balancer**. This load balancer must meet the following requirements:
  - Its VPC is the same as that of the CSS cluster. The network between the two is connected.
  - **IP as a Backend** is enabled. This is necessary to connect a dedicated load balancer to a CSS cluster.
  - Determine whether to configure an EIP based on service needs. A public IP address is displayed for the load balancer connecting the CSS cluster only if an EIP is configured. This will enable public network access to the cluster through this load balancer.
- If the ELB listener uses HTTPS, upload a server certificate or CA certificate to the ELB console. For details, see **Adding a Certificate**.
  - If one-way authentication is used, upload a server certificate.
  - If two-way authentication is used, upload a server certificate and a CA certificate.

Connecting a Cluster to a Load Balancer

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Cluster Access** tab, and then click the **Load Balancing** tab. On the **OpenSearch** tab, toggle on **Load Balancing**. In the displayed dialog box, set the parameters.

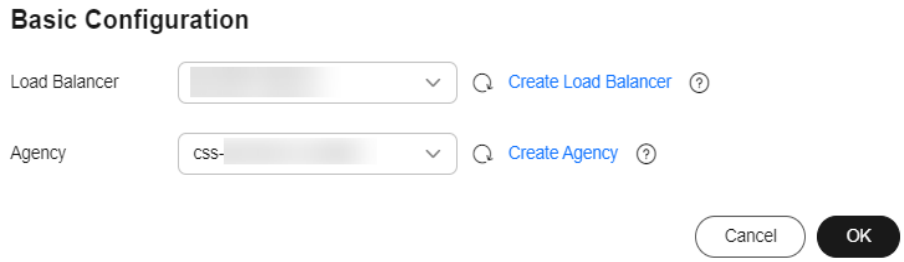
Table 3-37 Configuring load balancing

Parameter	Description
Load Balancer	Select the dedicated load balancer you have created earlier. To create a dedicated load balancer, see **Creating a Dedicated Load Balancer**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Agency	To configure a load balancer, you must have the permission to access ELB resources. By configuring an IAM agency, you can authorize CSS to access its ELB resources through an associated account. • If you are configuring an agency for the first time, click **Automatically Create IAM Agency** to create **css-elb-agency**. • If there is an IAM agency automatically created earlier, you can click **One-click authorization** to have the permissions associated with the **ELB Administrator** role or the **ELB FullAccess** system policy deleted automatically, and have the following custom policies added automatically instead to implement more refined permissions control. "elb:loadbalancers:list", "elb:loadbalancers:get", "elb:certificates:list", "elb:healthmonitors:*", "elb:members:*", "elb:pools:*", "elb:listeners:*" • To use **Automatically Create IAM Agency** and **One-click authorization**, the following minimum permissions are required: "iam:agencies:listAgencies", "iam:roles:listRoles", "iam:agencies:getAgency", "iam:agencies:createAgency", "iam:permissions:listRolesForAgency", "iam:permissions:grantRoleToAgency", "iam:permissions:listRolesForAgencyOnProject", "iam:permissions:revokeRoleFromAgency", "iam:roles:createRole" • To use an IAM agency, the following minimum permissions are required: "iam:agencies:listAgencies", "iam:agencies:getAgency", "iam:permissions:listRolesForAgencyOnProject", "iam:permissions:listRolesForAgency"

Figure 3-5 Enabling load balancing



5. Click **OK** to enable load balancing.
   Load balancer information is displayed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

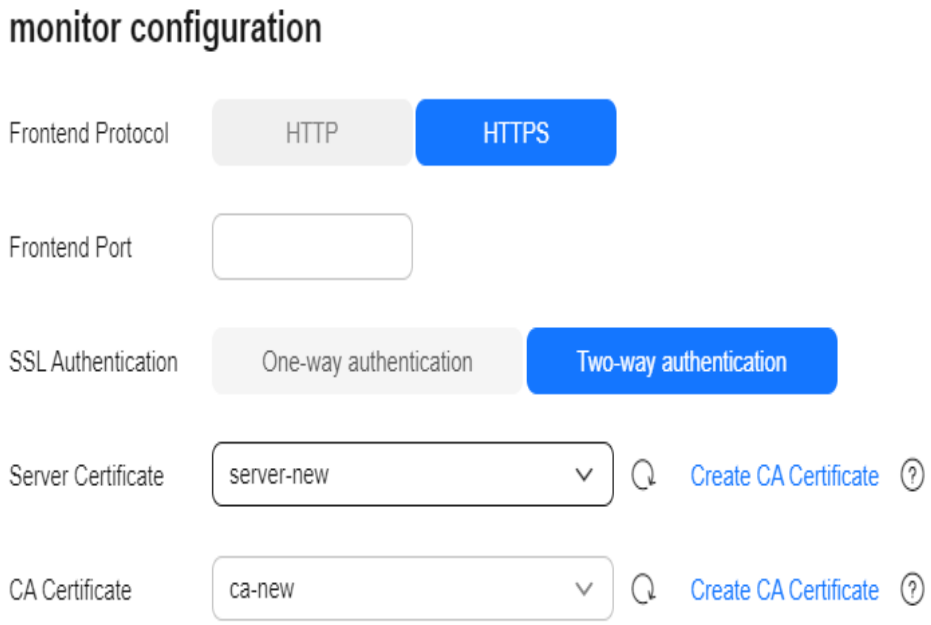
6. In the **Listener** area, click to configure a listener to ensure proper accessibility of the CSS cluster.

Table 3-38 Listener configuration

Parameter	Description
Frontend Protocol	Protocol used by the client and listener to distribute traffic. Select **HTTP** or **HTTPS**. Select this protocol based on your connectivity needs.
Frontend Port	Port used by the client and listener to distribute traffic. Set this parameter based on site requirements.
SSL Authentication	Client authentication mode. Set this parameter only when **Frontend Protocol** is set to **HTTPS**. Both one-way and two-way authentication are supported. Select an authentication mode that suits your needs.
Server Certificate	The server certificate is used for SSL handshake. The certificate content and private key must be provided. It is required only when **Frontend Protocol** is set to **HTTPS**. Select the server certificate created on ELB.
CA Certificate	Also called client CA public key certificate. It is used to verify the issuer of a client certificate. It is required only when **SSL Authentication** is set to **Two-way authentication**. Select the CA certificate created on ELB. When HTTPS two-way authentication is enabled, an HTTPS connection can be established only when the client can provide the certificate issued by a trusted CA.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-6 Configuring a listener



7. (Optional) In the **Listener** area, click **Configure** next to **Access Control** to go to the listener list of the load balancer. Click Configure in the Access Control column of a listener to configure access control for that listener. You can specify IP addresses that are allowed to access your cluster for enhanced security. For more information, see **What Is Access Control?**.

   WARNING

   Without access control policies, all IP addresses are allowed to access the CSS cluster through this load balancer, which may create security risks.

8. In the **Health Check** area, you can check the health check results for each node IP address, ensuring that each node in the cluster is functioning properly.

   Table 3-39 Health check result description

Health Check Result	Description
Normal	The node IP address is connected.
Abnormal	The node IP address is disconnected.

9. If the cluster no longer needs a dedicated load balancer, disassociate it to release resources.

   Choose **Load Balancing** > **OpenSearch**, toggle off **Load Balancing**. In the displayed dialog box, click **OK**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

> **CAUTION**
>
> After the load balancer is disassociated, any listener or backend server group configurations will be permanently deleted.

Accessing a Cluster Through a Load Balancer by Executing cURL Commands

1. In the left navigation pane on the CSS console, choose **Clusters**.
2. Log in to the **CSS management console**.
3. In the navigation pane on the left, choose **Clusters > OpenSearch**.
4. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
5. Click the **Cluster Access** tab, and then click the **Load Balancing** tab. On the **OpenSearch** tab, record the private or public IP address or IPv6 address of the load balancer, as well as the frontend protocol/port of the listener.

> **CAUTION**
>
> You are not advised to connect a load balancer that has been associated with a public IP address to a non-security mode cluster. Access from the public network using such a load balancer may cause security risks because a non-security mode cluster can be accessed using HTTP without security authentication.

6. Run the following cURL commands on an ECS to check whether the dedicated load balancer can connect to the cluster.

Table 3-40 Commands for accessing different types of clusters

Security Mode	Service Form Provided by ELB for External Systems	cURL Command for Accessing a Cluster
Non-security mode	No authentication	`curl http://IP:port`
	One-way authentication	`curl --cacert ./ca.crt https://IP:port`
	Two-way authentication	`curl --cacert ./ca.crt --cert ./client.crt --key ./client.key https://IP:port`

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Security Mode	Service Form Provided by ELB for External Systems	cURL Command for Accessing a Cluster
Security mode + HTTP	Password authentication	curl http://IP:port -u *user.pwd*
	One-way authentication + Password authentication	curl --cacert ./ca.crt https://IP:port -u *user.pwd*
	Two-way authentication + Password authentication	curl --cacert ./ca.crt --cert ./client.crt --key ./client.key https://IP:port -u *user.pwd*
Security mode + HTTPS	One-way authentication + Password authentication	curl --cacert ./ca.crt https://IP:port -u *user.pwd*
	Two-way authentication + Password authentication	curl --cacert ./ca.crt --cert ./client.crt --key ./client.key https://IP:port -u *user.pwd*

Table 3-41 Variables

Variable	Description
IP	IP address of a load balancer instance.
port	Frontend protocol and port configured for the listener.
user	Username of the cluster. This parameter is required only for a security-mode cluster.
pwd	Password of the username above. This parameter is required only for a security-mode cluster.

If cluster information is returned, the connection is successful.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

See also: **Sample Code for Accessing a Cluster Using HTTPS Two-way Authentication Through a Load Balancer (Java)**.

Sample Code for Accessing a Cluster Using HTTPS Two-way Authentication Through a Load Balancer (Java)

This section provides sample code that can be used on a Java client to access a cluster using HTTPS two-way authentication through a load balancer.

Sample Code for OpenSearchDemo

```
package org.example;

import java.io.ByteArrayInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;
import java.nio.charset.StandardCharsets;
import java.security.KeyStore;
import java.security.SecureRandom;
import java.security.cert.CertificateFactory;
import java.security.cert.X509Certificate;
import java.util.Base64;

import javax.net.ssl.KeyManagerFactory;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLHandshakeException;
import javax.net.ssl.TrustManagerFactory;

public class OpensearchDemo {
    private static final String TRUSTED_CA_PATH = "path\\ca.pem"; // CA certificate path
    private static final String PKCS12_PATH = "path\\client.p12"; // Client certificate path
    private static final String KEY_PASSWORD = ""; // Client certificate password
    private static final String HOST_NAME = ""; // ELB IP address or domain name
    private static final int PORT = 9200; //ELB port
    private static final String ELB_URL = "https://" + HOST_NAME + ":" + PORT;
    private static final String OS_USER = ""; // OpenSearch username
    private static final String OS_PASS = ""; // OpenSearch user password
    public static void main(String[] args) throws Exception {
        // 1. Create a standard SSLContext.
        SSLContext sslContext = createStrictMtlsSslContext();
        // 2. Inject the SSLContext.
        HttpClient client = HttpClient.newBuilder()
            .sslContext(sslContext)
            .connectTimeout(java.time.Duration.ofSeconds(10))
            .build();
        String authHeader = getBasicAuthHeader(OS_USER, OS_PASS);
        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create(ELB_URL + "/_cat/indices?v"))
            .GET()
            .header("Accept", "application/json")
            .header("Authorization", authHeader)
            .build();
        try {
            // Send a request.
            HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());
            System.out.println("http statusCode: " + response.statusCode());
            System.out.println("response body:\n" + response.body());
        } catch (SSLHandshakeException e) {
            System.err.println("SSLHandshakeException: " + e.getMessage());
            e.printStackTrace();
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        } catch (IOException e) {
            System.err.println("IOException: " + e.getMessage());
            e.printStackTrace();
        }
    }

    /**
     * Create an SSLContext.
     */
    private static SSLContext createStrictMtlsSslContext() throws Exception {
        // --- 1. Prepare a truststore (server authentication)---
        KeyStore trustStore = KeyStore.getInstance("PKCS12");
        trustStore.load(null, null);
        X509Certificate caCert = loadCertificate(TRUSTED_CA_PATH);
        trustStore.setCertificateEntry("ca-root", caCert);
        TrustManagerFactory tmf =
TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
        tmf.init(trustStore);
        // --- 2. Prepare a keystore (client identity)---
        KeyStore keyStore = KeyStore.getInstance("PKCS12");
        File p12File = new File(PKCS12_PATH);
        try (InputStream is = new FileInputStream(p12File)) {
            keyStore.load(is, KEY_PASSWORD.toCharArray());
        }
        KeyManagerFactory kmf =
KeyManagerFactory.getInstance(KeyManagerFactory.getDefaultAlgorithm());
        kmf.init(keyStore, KEY_PASSWORD.toCharArray());
        // --- 3. Initialize the SSLContext (standard mode)---
        SSLContext sslContext = SSLContext.getInstance("TLSv1.3");
        sslContext.init(kmf.getKeyManagers(), tmf.getTrustManagers(), new SecureRandom());
        return sslContext;
    }

    private static X509Certificate loadCertificate(String path) throws Exception {
        try (InputStream is = new FileInputStream(path)) {
            String pem = new String(is.readAllBytes(), StandardCharsets.UTF_8);
            String base64 = pem.replace("-----BEGIN CERTIFICATE-----", "")
                .replace("-----END CERTIFICATE-----", "")
                .replaceAll("\\s+", "");
            byte[] der = Base64.getDecoder().decode(base64);
            CertificateFactory cf = CertificateFactory.getInstance("X.509");
            return (X509Certificate) cf.generateCertificate(new ByteArrayInputStream(der));
        }
    }
    private static String getBasicAuthHeader(String user, String pass) {
        String auth = user + ":" + pass;
        String encoded = Base64.getEncoder().encodeToString(auth.getBytes(StandardCharsets.UTF_8));
        return "Basic " + encoded;
    }
}
```

pom.xml Sample Code

```
<?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>org.example</groupId>
    <artifactId>opensearch-demo</artifactId>
    <version>1.0</version>

    <properties>
        <maven.compiler.source>17</maven.compiler.source>
        <maven.compiler.target>17</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  <dependencies>
    <dependency>
      <groupId>org.opensearch.client</groupId>
      <artifactId>opensearch-java</artifactId>
      <version>2.19.0</version>
    </dependency>

    <dependency>
      <groupId>org.opensearch.client</groupId>
      <artifactId>opensearch-rest-client</artifactId>
      <version>2.19.0</version>
    </dependency>

    <dependency>
      <groupId>org.apache.httpcomponents.client5</groupId>
      <artifactId>httpclient5</artifactId>
      <version>5.3.1</version>
    </dependency>

    <dependency>
      <groupId>com.fasterxml.jackson.core</groupId>
      <artifactId>jackson-databind</artifactId>
      <version>2.17.0</version>
    </dependency>
  </dependencies>
</project>
```

3.5.4 Configuring VPC Endpoints

In CSS, you can configure VPC endpoints as dedicated gateways to enable cross-VPC resource access via an internal network without exposing server details. This ensures more secure and reliable internal connectivity for enterprise applications. A VPC endpoint can be accessed via an IPv4 address or private domain name.

- An IPv4 address is automatically allocated when VPC Endpoint is enabled.
- A private domain name is allocated only when you enable private domain names.

VPC Endpoint uses a shared load balancer for internal network access. If your workloads require faster access, we recommend that you use a dedicated load balancer to handle access to your cluster. For details, see **Configuring a Dedicated Load Balancer**.

Impact on Billing

VPC endpoints, if created for the cluster, will incur extra fees, depending on the resource usage. For details, see **VPC Endpoint Billing**.

Constraints

- You need specific permissions to create VPC endpoints. For details, see **VPCEP Permissions**.
- After VPCEP is enabled, access to CSS through a VPCEP IP address or private domain name from within the internal network is not controlled by any cluster security group rules. Rather, you need to configure a VPCEP whitelist to implement access control. For details, see section "Configuring Access Control for an Interface VPC Endpoint" in *VPC Endpoint User Guide*.
- Public network access and VPC Endpoint share the same shared load balancer. If you configure a whitelist for public network access control, this whitelist is deployed to the shared load balancer. As such, it will control access not only

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

from the public network but also from private IP addresses through VPCEP. In this case, you need to add IP address **198.19.128.0/17** to the public access control whitelist to allow traffic through VPCEP.

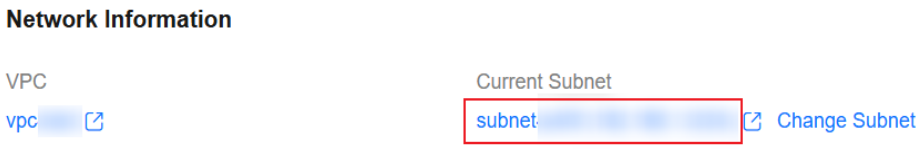
- When you access a cluster through the IP address or private domain name of a VPC endpoint, you are not subject to the cluster's security group rules, and you can configure a whitelist for access control.

Enabling VPC Endpoint

If VPC Endpoint was not enabled when the cluster was created, you can enable it as follows:

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, check the cluster's subnet information in the **Network Information** area. When you add a VPC endpoint, an IP address that belongs to the current subnet of the cluster is automatically assigned to it.

Figure 3-7 Checking the current subnet



If you want to use another subnet, switch the subnet first, and then enable VPC Endpoint. For details, see

5. Choose **Cluster Access** > **VPC Endpoint**.
6. Toggle on **VPC Endpoint**. In the displayed dialog box, select relevant options.

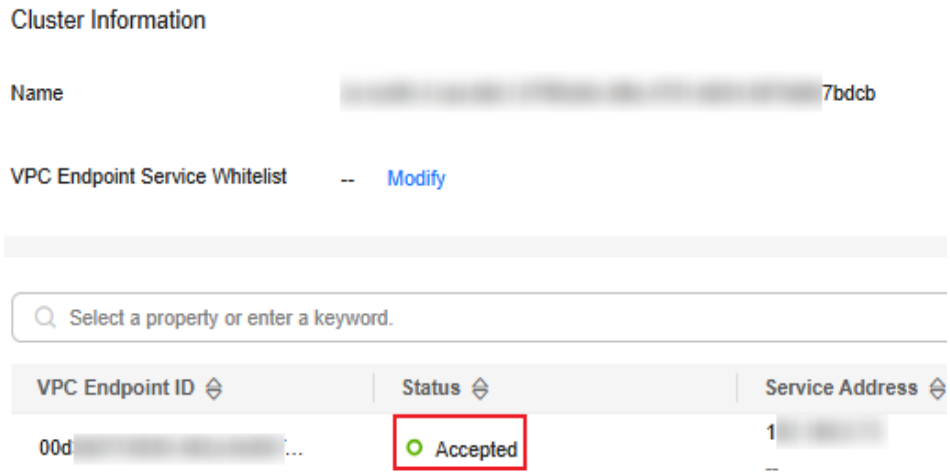
Table 3-42 Options for enabling VPC Endpoint

Option	Description
Create Private Domain Name	Whether to create a private domain name for the VPC endpoint. • Enable: The system automatically assigns a private domain name to the VPC endpoint. After cluster creation, you can check this private domain name on the **VPC Endpoint** tab of the cluster details page. • Disable: No private domain name will be configured for the VPC endpoint. The cluster can only be accessed through an IP address assigned to the VPC endpoint.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

7. Click **OK** to enable VPC Endpoint. After the VPC endpoint is enabled, its information is displayed below. When its status changes to **Accepted**, the current cluster can be accessed through the VPC endpoint.

Figure 3-8 Checking the VPC endpoint status

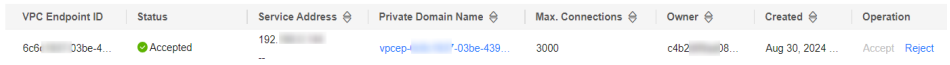


Managing VPC Endpoints

After VPC Endpoint is enabled for a cluster, you can set access control, check VPC endpoint information, and deny access from specific VPC endpoints.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Cluster Access** > **VPC Endpoint**.
5. **Configure access control for a VPC endpoint.**
   a. Click **Modify** on the right of **VPC Endpoint Whitelist**. In the displayed dialog box, add accounts that are allowed to access the cluster through the VPC endpoint. If no account is added or the account ID is set to *, all users are allowed to access the cluster.
      - Click **Add** to add accounts in **Account ID**. To obtain your authorized account ID, point to your username in the upper right corner, and choose **My Credentials**. Copy the value of **Account ID**.
      - Click **Delete** in the **Operation** column to delete an authorized account.
   b. Click **OK**.
6. **Check VPC endpoint information.**
   The VPC endpoint list shows the VPC endpoints created for the current cluster. You can obtain their **Status**, **Service Address**, and **Private Domain Name**.

Figure 3-9 Managing VPC endpoints



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

7. **Modify VPC endpoint status** to make the cluster accessible or inaccessible through specific endpoints.
   - In the VPC endpoint list, select the target endpoint and click **Reject** in the **Operation** column. If the endpoint status changes to **Rejected**, it means the cluster is no longer accessible through this endpoint.
   - Select an endpoint whose status is **Rejected**. Click **Accept** in the **Operation** column. If the endpoint status changes to **Accepted**, it means the cluster is accessible again through this endpoint.

Disabling VPC Endpoint

If the cluster no longer requires cross-VPC access via VPC endpoints, disable VPC Endpoint to release resources.

WARNING

After VPC Endpoint is disabled, the cluster is no longer accessible through a VPCEP IP address or private domain name. If you disable VPC Endpoint and then re-enable it, the VPCEP IP address or private domain name for accessing the cluster may change. When it happens, you may need to update the client connection. If you just need to temporarily disable VPC Endpoint (rather than permanently releasing its resources), do so by rejecting specific VPC endpoints. For details, see 7.

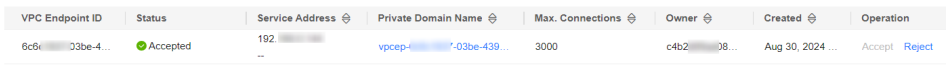
1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Cluster Access** > **VPC Endpoint**.
5. Toggle off **VPC Endpoint**. In the displayed dialog box, enter **CONFIRM** and click **OK**.

Accessing a Cluster Through a VPC Endpoint

1. Obtain the private domain name or IP address of a VPC endpoint.

   In the VPC endpoint list, check **Service Address** or **Private Domain Name**.

   Figure 3-10 Checking the service address or private domain name of a VPC endpoint

2. On the client (e.g., an ECS), run the curl command to access the cluster.

   For example, run the following command to check the cluster's index information:

   - For a cluster with the security mode disabled:
     ```
     curl "http://<host>:9200/_cat/indices"
     ```
   - For a security-mode cluster that uses HTTP:
     ```
     curl -u <user>:<password> "http://<host>:9200/_cat/indices"
     ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

  - For a security-mode cluster that uses HTTPS:
    ```
    curl -u <user>:<password> -k "https://<host>:9200/_cat/indices"
    ```

  where, **user** and **password** are the username and password used to access the cluster, and **host** indicates the VPC endpoint's private domain name or IP address.

3.5.5 Configuring Routes for a Cluster

By default, a CSS OpenSearch cluster cannot access a target service—such as a client program or a Lightweight Directory Access Protocol (LDAP) service—that is located in a different VPC. To enable it, you need to configure a route for the cluster.

1. Establish a VPC peering connection to connect two VPCs.
2. On CSS, configure a route for the OpenSearch cluster that points to the client.
3. Verify network connectivity after starting services.

Configuring routes to enable cluster communication offers the following advantages:

- Improved cross-VPC communication performance: A route table specifies the traffic forwarding paths, so that traffic does not have to pass through the public network.
- Secure communication: Private networks can communicate through VPC peering connections, which is more secure than transmitting data over the public network.
- Flexible access control policies: You can customize access permissions for specific subnets.
- Support for a hybrid cloud architecture: Services can communicate across VPCs in different environments.

Constraints

- By default, routes can be configured only for OpenSearch clusters whose image version is no earlier than x.x.x_25.3.0_x.x.x.
- Adding cluster routes will alter network connections, possibly causing disruptions in the network. Exercise caution.

Prerequisites

- The OpenSearch cluster is Available, and port 9200 is allowed in the inbound direction by the cluster's security group.
- The client is running properly, and the server's security group allows port 9200 in the outbound direction.

Establishing a VPC Peering Connection

Before configuring routes for a CSS cluster, establish a connection between the CSS cluster's VPC and that of the client, that is, a VPC peering connection.

- If the CSS cluster and the client belong to the same account, see section "Creating a VPC Peering Connection to Connect Two VPCs in the Same Account" in *Virtual Private Cloud User Guide*.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- If the CSS cluster and the client belong to different accounts, see section "Creating a VPC Peering Connection to Connect Two VPCs in Different Accounts" in *Virtual Private Cloud User Guide*.

Set the peer VPC to the VPC where the CSS cluster is located and the local VPC to the VPC where the client is located.

When adding routes for a VPC peering connection, configure the destination addresses based on service requirements.

If the CSS cluster and the client share the same security group, there is no need to configure security group rules for the instances in the VPC at either end of the VPC peering connection.

Configuring a Route for an OpenSearch Cluster

Configure a route for the OpenSearch cluster that points to the client, enabling the cluster to access the client.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, click **Add Route** under **Cluster Route** in the **Network Information** area.
5. In the displayed dialog box, configure the route information.

   Table 3-43 Adding a route

Parameter	Description
IP Address	Enter the IP address (or a CIDR block) of the client or server that the cluster needs to access, for example, **10.10.1.0**.
Subnet Mask	Enter the subnet mask used with the IP address. The subnet mask must align with the network part of the IP address. That is: • The network bits (the 1s) of the subnet mask must encompass those of the IP address. • The host bits (the 0s) of the subnet mask must accommodate the IP address's host bits. The subnet mask should not cover too wide a range, avoiding unnecessary IP address exposure or waste of the address space. 255.255.255.0 (that is, /24) is recommended, which applies to most internal networking scenarios.

6. Click **OK** to add the route.

   After the route is added, click **View Route** to check the route information. You can click **Add Route** to add another route; or click **Delete Route** to delete

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

one—In the **Delete Route** dialog box, select a route, enter **DELETE** manually, and then click **OK**.

Testing Connectivity

Connectivity between the OpenSearch cluster and client cannot be verified right away. After the route is configured, start services. If the search service can be accessed, the route is configured successfully.

3.6 Security Settings

3.6.1 Creating Cluster Users and Granting Permissions

An OpenSearch cluster with security mode enabled uses access control. Initially, only the administrator account can log in to OpenSearch Dashboards and access cluster data. In real-world business scenarios, different users (such as developers, O&M personnel, and data analysts) need to log in to the cluster using their own identities and perform only authorized operations on authorized indexes and features. To meet this requirement, CSS provides user and permission management based on the role-based access control (RBAC) model. Administrators can create users, configure roles and permissions, and assign roles to users through role mappings in the Security module of OpenSearch Dashboards. This enables fine-grained access control at the cluster, index, document, and field levels.

How the Feature Works

CSS security-mode clusters implement fine-grained access control based on the opendistro_security plugin. With the RBAC model at its core, this plugin implements authorization through the chain of users → role mappings → roles → permissions.

Figure 3-11 RBAC authorization model



Table 3-44 Core concepts

Concept	Description
User	The entity that sends requests to the cluster. A user has a username and password and can be associated with backend roles and custom attributes.
Role mapping	Associates users (or backend roles) with specific roles, granting them all permissions associated with those roles. A role can be mapped to multiple users (or backend roles). For example, if the **readall** role is mapped to user **Bob**, **Bob** will have all permissions of the **readall** role after authentication.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Concept	Description
Role	A collection of permissions or action groups that defines the operations allowed on clusters, indexes, documents, and fields.
Permission	A specific action, such as **indices:admin/create** (creating an index).
Action group	A collection of permissions. For example, the predefined **search** action group contains search-related permissions such as **search** and **msearch**.

RBAC answers the question "Who can do what in the cluster?", while Tenant addresses the question "What OpenSearch Dashboards objects can users see and edit?" RBAC and Tenant work together to provide fine-grained access control. RBAC determines the operations that users can perform on cluster resources, while Tenant controls the visibility and sharing of OpenSearch Dashboards resources, such as dashboards and visualizations. Three tenant types are available to define the visibility of Kibana resources.

Table 3-45 Tenant types

Type	Sharing Scope	Applicable Scenario
Global (global tenant)	Accessible to all users.	Shared dashboards, public monitoring views, and other common resources
Private (private tenant)	Accessible only to the resource creator. Even the administrator cannot directly view private tenant content.	Private reports used by individuals for development and debugging
Custom (custom tenant)	Administrators can precisely specify a group of users or roles through role mappings and further assign Tenant Read (read-only) or Tenant Write (write) permissions.	Team collaboration and isolation by department

Constraints

- Usernames, role names, and tenant names must be unique.
- You are advised not to use the same name for indexes and users. This helps avoid permission management confusion.

Prerequisites

- The security mode has been enabled for your cluster. Only security-mode OpenSearch clusters support user permission management.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- You have obtained the administrator account. The username is **admin** and the password is the administrator password configured when the cluster was created.

Logging In to the Security Page of OpenSearch Dashboards

Log in to the **Security** page of Kibana using the administrator account to manage cluster permissions.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards using the **admin** account.
4. After successful login, choose **Security** from the navigation pane on the left. The permission management page is displayed.

Creating a User

Create a new cluster user. The user can then be mapped to roles to be granted the corresponding permissions.

1. On the **Security** page, choose **Internal users** from the navigation pane on the left.
2. On the **Internal Users** page, click **Create internal user**.
3. On the **Create internal user** page, configure the required parameters.

Table 3-46 User configuration

Parameter	Mandatory (Yes/No)	Description
Username	Yes	Username. In this example, enter **test**.
Password/Re-enter password	Yes	Enter and confirm the login password. The two passwords must be identical.
Backend roles	No	Backend role names from an external authentication system (such as LDAP or SAML). Backend roles are used to map external users to security roles. For details, click **Learn more** on the page.
Attributes	No	Custom user attributes. They can be used as dynamic variables in document-level security (DLS) queries to implement fine-grained data access control based on user attributes. For details, click **Learn more** on the page.

4. Click **Create**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

After the user is created, it is displayed in the **Internal Users** list.

Creating a Role and Defining Its Permissions

Define a role and specify the operations that the role is allowed to perform on clusters and indexes, as well as the scope of these operations. The permissions defined in the role can then be granted to users by mapping the role to them.

NOTE

The system provides multiple predefined roles (such as **kibana_user** and **readall**). If the predefined roles meet your requirements, you are advised to use them directly instead of creating new ones.

1. On the **Security** page, choose **Roles** from the navigation pane on the left.
2. On the **Roles** page, click **Create Role**.
3. On the **Create Role** page, configure the required parameters.

Table 3-47 Role configuration

Parameter	Mandatory (Yes/No)	Description
Name	Yes	Role name. In this example, enter **role_test**.
Cluster permissions	No	Cluster-level permissions. • Select a predefined action group, such as **cluster_monitor** (read-only cluster monitoring) or cluster_manage (**cluster management**). In this example, select **cluster_monitor**. The **cluster_monitor** action group allows users to view cluster status in read-only mode, including cluster health, node information, and statistics, but does not allow modification operations. • Leaving this parameter blank means not to grant any cluster-level permissions.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory (Yes/No)	Description
Index Permissions	No	Index-level permissions. Set the following parameters: • **Index**: Mandatory. The index name or wildcard pattern to which permissions are granted, such as **my_store** or **logs-***. • **Index permissions**: Mandatory. Operations allowed on the index, such as **read** and **write**. • **Document Level Security**: Optional. Sets document-level filters so that users with this role can only access documents that meet the specified conditions. • **Field Level Security**: Optional. Sets field allowlists or blocklists to control which fields are visible to users. • **Anonymization**: Optional. Specifies fields that need to be anonymized to protect data security.
Tenant Permissions	No	Tenant permissions. • **Tenant patterns**: Specifies tenant spaces that users can access, such as global_tenant or custom tenant names. • Permissions: Tenant space permissions, which can be **Read only** or **Read and Write**.

4. Click **Create**.

   After the role is created, it is displayed in the **Roles** list.

Mapping Roles to Users

Map roles to users to grant users all permissions associated with the mapped roles.

1. On the **Security > Roles** page, click a role name, such as **role_test**. The role details page is displayed.
2. Click the **Mapped users** tab and then click **Manage mapping**.
3. On the **Map user** page, click the drop-down list box next to **Users** and select the target user, for example, **test**.

   NOTE

   A role can be mapped to multiple users. You can also use **Backend roles** to batch-map user groups from external authentication systems.
4. Click **Map** to complete the role mapping.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

After the mapping is successful, the target user **test** is displayed in the **Mapped users** list.

Verifying User Permissions

Log in to the cluster as the new user and perform operations to verify that the user's permission configuration has taken effect as expected.

1. Log out of the current **admin** account and log in to OpenSearch Dashboards again using the username **test** and the configured password.
2. On Kibana, expand the menu in the upper-left corner, and choose **Dev Tools**.
3. Run the following two commands to verify that user **test** has the permissions granted by role **role_test**, that is, only the **cluster_monitor** permission.
   - Query cluster health status.
     ```
     GET /_cluster/health?pretty
     ```
     Expected result: HTTP status code 200 is returned and cluster health information is displayed, indicating that the **cluster_monitor** permission has taken effect.
   - Create an index.
     ```
     PUT /my_test
     ```
     Expected result: HTTP status code 403 Forbidden is returned, indicating that user **test** does not have permission to create indexes and permission isolation is working.

     NOTE

     If you need to grant user **test** the permission to create indexes, add the corresponding permission (for example, **indices:admin/create**) to **Index Permissions** of **role_test** based on the permission name in the 403 error message. You do not need to create a new user.

Common Role Permission Configuration Examples

In real-world scenarios, users in different roles require different levels of permissions. This section provides role permission configurations for some common scenarios. You can refer to these examples when configuring roles.

Table 3-48 Role permission configuration examples

Scenario	Applicable Roles	Core Permissions
Read-only access to index data	Data analysts, report query users	Read permissions on specified indexes (without cluster permissions)
Read/write access to index data	Application developers, data ingestion service accounts	Read and write permissions on specified indexes (without cluster permissions)
Read-only cluster monitoring	O&M personnel, inspection accounts	Cluster monitoring permissions (without index data permissions)

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Scenario	Applicable Roles	Core Permissions
Index and document management	Application service accounts, data pipeline ingestion accounts	Full index and document management permissions (without cluster permissions)
Dashboard editing (read/write)	O&M engineers, BI application developers	OpenSearch Dashboards dashboard permissions + Tenant read/write permissions

- **Read-only access to index data**

  Scenario: Common users need to query data in specified indexes through OpenSearch Dashboards but are not allowed to perform any write, update, or delete operations. They are also not allowed to perform cluster-level operations, preventing accidental operations that may affect production data.

  User permission configuration: Map the user to the role **role_index_readonly**.

  Table 3-49 Permission configuration for read-only access to index data

Configuration Item	Sub-Configuration Item	Description
Name	-	Role name: **role_index_readonly**
Cluster permissions	-	Not configured. No cluster-level permissions are required.
Index Permissions	Index	Target index name or wildcard.
	Index permissions	Select the predefined action group **read** (document read), which includes the following permissions: indices:data/read/search (_search) indices:data/read/get (_get) indices:data/read/mget (_mget) indices:data/read/scroll (_scroll) indices:admin/mappings/fields/get* (view field mappings)

  Permission verification:

```
# Expected success (search for documents)
GET /my_store/_search
{ "query": { "match_all": {} } }

# Expected failure (no write permission)
POST /my_store/_doc
{ "name": "test" }

# Expected failure (no cluster-level permissions)
GET /_cluster/health?pretty
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- **Read/write access to index data**

  Scenario: Common users need to write, update, and query data in specified indexes through OpenSearch Dashboards but are not allowed to create or delete indexes or modify index mappings and other structural configurations. This ensures that index structures are managed by dedicated personnel.

  User permission configuration: Map the user to the role **role_index_readwrite**.

  Table 3-50 Permission configuration for read/write access to index data

Configuration Item	Sub-Configuration Item	Description
Name	-	Role name: **role_index_readwrite**
Cluster permissions	-	Not configured. No cluster-level permissions are required.
Index Permissions	Index	Target index name or wildcard.
	Index permissions	Select the predefined action groups **read** (document read) and **write** (document write). • The **read** action group includes the following permissions: indices:data/read/search (_search) indices:data/read/get (_get) indices:data/read/mget (_mget) indices:data/read/scroll (_scroll) indices:admin/mappings/fields/get* (view field mappings) • The **write** action group includes the following permissions: indices:data/write/index (create/replace documents) indices:data/write/update (update documents) indices:data/write/delete (delete documents) indices:data/write/bulk (batch write)

Permission verification:

```
# Expected success (query documents)
GET /my_store/_search
{ "query": { "match_all": {} } }

# Expected success (write documents)
POST /my_store/_doc
{ "product": "laptop", "price": 5999 }

# Expected success (delete a document)
DELETE /my_store/_doc/1

# Expected failure (no index creation permission)
PUT /my_store_v2

# Expected failure (no index deletion permission)
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
DELETE /my_store

# Expected failure (no cluster-level permissions)
GET /_cluster/health?pretty
```

- **Read-only cluster monitoring**

  Scenario: O&M personnel or automated inspection scripts need to periodically collect monitoring data such as cluster health status, node loads, and shard distribution. However, they are not allowed to modify cluster status or perform read/write operations on index data, preventing inspection accounts from being misused.

  User permission configuration: Map the user to the role **role_index_monitor**.

  Table 3-51 Permission configuration for read-only cluster monitoring

Configuration Item	Sub-Configuration Item	Description
Name	-	Role name: **role_index_monitor**
Cluster permissions	-	Select the predefined action group **cluster_monitor** (read-only cluster monitoring), which includes the following permissions: cluster:monitor/health (cluster health status) cluster:monitor/stats (cluster statistics) cluster:monitor/nodes/info (node information) cluster:monitor/nodes/stats (node statistics) cluster:monitor/task/get (task information) cluster:monitor/pending_tasks (pending tasks)
Index Permissions	Index	Set to * to view the status of all indexes.
	Index permissions	Select the predefined action group **indices_monitor** (read-only index monitoring), which includes the following permissions: indices:monitor/stats (index statistics) indices:monitor/segments (segment information) indices:monitor/recovery (recovery information) indices:monitor/shard_stores (shard storage information)

  Permission verification:

```
# Expected success (cluster health status)
GET /_cluster/health?pretty

# Expected success (index statistics)
GET /my_store/_stats

# Expected failure (no document query permissions)
GET /my_store/_search
{ "query": { "match_all": {} } }

# Expected failure (no permission to modify the cluster configuration)
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
PUT /_cluster/settings
{ "persistent": { "cluster.routing.allocation.enable": "none" } }
```

- **Index and document management**

  Scenario: Microservice applications or data pipeline service accounts need to use REST APIs to perform full index lifecycle management in OpenSearch, including index creation, index deletion, index settings updates, and mapping updates, as well as document CRUD operations. Because these accounts are used by applications rather than human users, it is recommended that you clearly define the data scope to reduce potential data security risks.

  User permission configuration: Map the user to the role **role_api_service**.

  Table 3-52 Permission configuration for index and document management

Configuration Item	Sub-Configuration Item	Description
Name	-	Role name: **role_api_service**
Cluster permissions	-	Not configured. No cluster-level permissions are required for pure data operations.
Index Permissions	Index	Use wildcards to specify the scope of indexes, for example, **log-***.
	Index permissions	Select the following predefined action groups/permissions: • **read** (document read) • **write** (document write) • **indices:admin/create** (create indexes) • **indices:admin/delete** (delete indexes) • **indices:admin/settings/update** (modify index settings) • **indices:admin/mapping/put** (add or modify index mappings)

  Permission verification:

```
# Expected success (create an index)
PUT /log-v1
{
  "settings": { "number_of_shards": 3, "number_of_replicas": 1 }
}

# Expected success (modify index settings)
PUT /log-v1/_settings
{
  "index": { "number_of_replicas": 2 }
}

# Expected success (modify index mappings)
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
PUT /log-v1/_mapping
{
  "properties": { "new_field": { "type": "keyword" } }
}

# Expected success (write documents)
POST /log-v1/_doc
{ "user_id": "u001", "action": "login" }

# Expected success (query documents)
GET /log-v1/_search
{ "query": { "match_all": {} } }

# Expected success (delete a document)
DELETE /log-v1/_doc/1

# Expected success (delete an index)
DELETE /log-v1

# Expected failure (no cluster-level permissions)
GET /_cluster/health?pretty
```

- **Dashboard editing (read/write)**

  Scenario: BI developers or O&M engineers need to create and modify dashboards, visualizations, and index patterns in OpenSearch Dashboards.

  User permission configuration: Map the user to roles **kibana_user** and **role_dashboard_editor**.

  Table 3-53 Permission configuration for dashboard editing

Configuration Item	Sub-Configuration Item	Description
Name	-	Role name: **role_dashboard_editor**
Cluster permissions	-	Not configured. No cluster-level permissions are required.
Index Permissions	Index	Target index name or wildcard, for example, **log-***.
	Index permissions	Select the predefined action groups **read** (document read) and **write** (document write).
Tenant Permissions	Tenant patterns	Leave this blank or specify a custom tenant name. By default, the role **kibana_user** has the **global_tenant** permission, so a custom role does not have to be configured here. To isolate tenant spaces, specify a custom tenant name, for example, **test_tenant**.
	Permissions	If a custom tenant is specified, select **Read and Write** to grant read/write permissions for dashboards in the tenant.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Permission verification:

Log in to OpenSearch Dashboards using an account assigned these roles and switch the tenant to **Global**.

- You can view, create, and edit dashboards and visualizations.
- You can create new index patterns.
- You cannot access the **Security** page for permission management. (The **all_access** role is required to access this page.)

FAQ: What predefined roles are available in the system?

- **kibana_user**: Allows users to create and view dashboards in OpenSearch Dashboards. This role has the **global_tenant** permissions by default.
- **readall**: Grants read-only access to all indexes. It is suitable for data analysts.
- **all_access**: Grants administrator permissions. Assign this role with caution.

On the **Roles** page, click **Learn more** to view descriptions of all predefined roles.

3.6.2 Configuring LDAP Authentication

In enterprise applications, user authentication is essential for ensuring data security and proper access control. However, traditional authentication methods often require repeated configuration for each application, which can be complex and error-prone. Using the Security plugin for Open Distro, CSS adds Active Directory as an authentication backend for clusters, connecting them seamlessly to the LDAP service. The Lightweight Directory Access Protocol (LDAP) is a lightweight version of the directory access protocol based on the X.500 standard, providing user authentication and authorization capabilities. This topic will describe how to enable LDAP authentication for a CSS OpenSearch cluster to grant access to LDAP users of specific roles, and how to connect to a cluster using LDAP.

Preparations

- A security-mode OpenSearch cluster has been created in CSS and its status is available.
- The LDAP service that is in the same VPC as the OpenSearch cluster and the necessary user data have been prepared. For details, see **the OpenLDAP document: A Quick-Start Guide**.

Configuring LDAP Authentication

Step 1 Install an LDAP service on an ECS. If the LDAP service and user data have already been prepared, skip this step.

1. Create an ECS. The ECS must run a Windows OS and must be in the same VPC and security group as the security-mode Elasticsearch cluster of CSS. The Windows Server running on the ECS provides the built-in Active Directory service that supports the LDAP protocol.
2. Log in to the ECS, and enable the Active Directory service. Create a domain, administrator, users, and user groups.

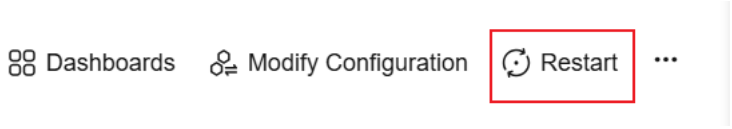
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Step 2 Modify the parameter settings of the security-mode OpenSearch cluster on CSS. Configure a static parameter in **opensearch.yml** to connect the cluster to the LDAP service.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Cluster Settings** > **Parameter Settings** > **OpenSearch**, and click **Edit**. Expand **Custom**, and add the following parameter and value.
   - **Parameter**: plugins.security.unsupported.restapi.allow_securityconfig_modification
   - **Value**: true
5. Click **Submit** above. In the displayed **Submit Configuration** dialog box, select the box that says "I understand that the modification will take effect after the cluster is restarted." and click **Yes**.

   If **Job Status** is **Succeeded** in the parameter settings task list below, the change has been saved.
6. After the changes are saved, click **Restart** in the upper right corner to restart the cluster, thus applying the changes.

Figure 3-12 Restarting a cluster



- You need to restart the cluster after the change, or **Configuration not updated** will be displayed in the **Task Status** column in the cluster list.
- If you restart the cluster after the change, and **Task Status** displays **Configuration error** in the cluster list, the parameter configuration file has failed to be modified.

Step 3 Configure a route for an OpenSearch cluster on the CSS console to connect the cluster to the LDAP service.

1. On the cluster information page, click the **Overview** tab.
2. In the **Network Information** area, click **Add Route** under **Cluster Route**.
3. In the displayed dialog box, configure the route information.

Table 3-54 Adding a route

Parameter	Description
IP Address	Enter the IP address of the LDAP server. If the LDAP service on the ECS is used, enter the IP address of the ECS.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Subnet Mask	Enter the subnet mask of the LDAP server. If the LDAP service on the ECS is used, enter the subnet mask of the ECS.

4. Click OK.

Step 4 Configure LDAP authentication for a security-mode OpenSearch cluster.

1. On the cluster information page, click **Dashboards** in the upper-right corner to log in to OpenSearch Dashboards.
2. On OpenSearch Dashboards, expand the menu in the upper-left corner, and choose **Dev Tools**.
3. Run the following commands to configure LDAP authentication.

   NOTE

   - Concepts used in an X.500 directory access protocol (including LDAP):
     - CN = Common Name
     - OU = Organizational Unit
     - DC = Domain Component
     - DN = Distinguished Name

     The CN, OU, and DC must be provided in the correct order. Otherwise, authentication will fail.
   - The configuration file consists of two parts: **authc** and **authz**.
     - **authc** (authentication): verifies whether a user is truly who they claim they are (password verification).
     - **authz** (authorization): verifies what the current user has access to.

```
PUT _plugins/_security/api/securityconfig/config
{
   "dynamic": {
      "authc": {
         "basic_internal_auth_domain": {
            "description": "Authenticate via HTTP Basic against internal users database",
            "http_enabled": true,
            "transport_enabled": true,
            "order": 1,
            "http_authenticator": {
               "type": "basic",
               "challenge": true
            },
            "authentication_backend": {
               "type": "intern"
            }
         },
         "ldap": {
            "description": "Authenticate via LDAP or Active Directory",
            "http_enabled": true,
            "transport_enabled": true,
            "order": 2,
            "http_authenticator": {
               "type": "basic",
               "challenge": false
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            },
            "authentication_backend": {
               "type": "ldap",
               "config": {
                  "enable_ssl": false,
                  "enable_start_tls": false,
                  "enable_ssl_client_auth": false,
                  "verify_hostnames": true,
                  "hosts": ["10.0.XXX.XXX:389"],
                  "bind_dn": "CN=adminAD,DC=test,DC=ldap,DC=com",
                  "password": "<password>",
                  "userbase": "OU=ITDepartment,DC=test,DC=ldap,DC=com",
                  "usersearch": "(sAMAccountName={0})",
                  "username_attribute": "uid"
               }
            }
         }
      },
      "authz": {
         "roles_from_myldap": {
            "description": "Authorize via LDAP or Active Directory",
            "http_enabled": true,
            "transport_enabled": true,
            "authorization_backend": {
               "type": "ldap",
               "config": {
                  "enable_ssl": false,
                  "enable_start_tls": false,
                  "enable_ssl_client_auth": false,
                  "verify_hostnames": true,
                  "hosts": ["10.0.XXX.XXX:389"],
                  "bind_dn": "CN=adminAD,DC=test,DC=ldap,DC=com",
                  "password": "<password>",
                  "rolebase": "OU=groups,DC=test,DC=ldap,DC=com",
                  "rolesearch": "(member={0})",
                  "userroleattribute": null,
                  "userrolename": "disabled",
                  "rolename": "CN",
                  "resolve_nested_roles": true,
                  "userbase": "OU=ITDepartment,DC=test,DC=ldap,DC=com",
                  "usersearch": "(uid={0})"
               }
            }
         }
      }
   }
}
```

The parameters in **Table 3-55** need to be modified based on the actual environment.

Table 3-55 Parameter description

Parameter	Description
hosts	Address of the LDAP service. The port number is 389. If the LDAP service on the ECS is used, enter the IP address of the ECS.
bind_dn	It is similar to the LDAP user name (CN - OU - DC) and is used to access the LDAP server. Select a user name from the user data of the LDAP service.
password	Password of the LDAP user configured using **bind_dn**.

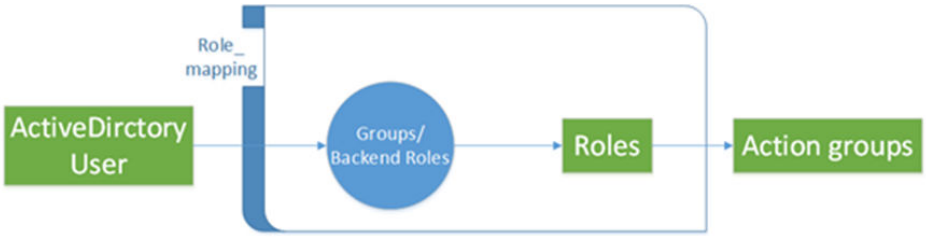
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
userbase	After the LDAP service is connected, the DN that the user belongs to is obtained. In this example, all user information in the **ITDepartment** directory is synchronized.
rolebase	The collection of permissions that can be configured for the **userbase** user group of the LDAP service.

Step 5 Configure the mapping between LDAP user permissions and OpenSearch permissions in the OpenSearch security-mode cluster to enable fine-grained access control.

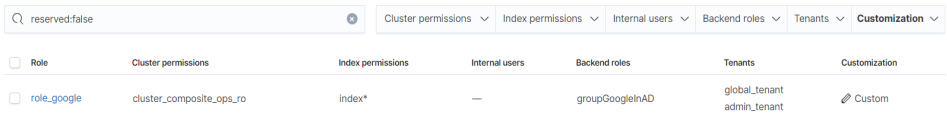
The rolebase permissions group of the LDAP server must be mapped to the roles in the OpenSearch cluster. **Figure 3-13** illustrates the mapping. For details about the configuration, see **Creating Cluster Users and Granting Permissions**.

Figure 3-13 Permissions mapping



1. On OpenSearch Dashboards, expand the menu in the upper-left corner, and choose **Security**. The **Security** page is displayed.
2. Click **Roles** to go to the Open Distro Security Roles page. Click **Create Role**, set **Name**, **Cluster Permissions**, **Index Permissions**, and **Tenant Permissions**. Then click **Save Role Definition** to save the role settings. The parameters are as follows:
   - Name (name of the role)
   - Cluster Permissions
   - Index permissions
   - Tenant permissions
3. Click the newly created role, select **Mapped users**, enter a permissions group of the LDAP service in **Backend roles**, and click **Map**.
4. Check the configuration result.

Figure 3-14 Permissions mapping



5. Repeat **Step 5.2** to **Step 5.4** to map other permissions groups.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Step 6 Log in to OpenSearch Dashboards using the LDAP user to verify the configuration.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column. Use the LDAP user to log in to OpenSearch Dashboards.

   If the login is successful, the configuration is successful, and users can access the OpenSearch cluster through LDAP. The specific permissions granted are controlled by role permissions configured in OpenSearch.

----End

3.7 Connecting to Clusters

3.7.1 Cluster Connection Methods

To ensure secure and efficient access to deployed OpenSearch clusters, CSS provides multiple connection options tailored to different roles, programming languages, and network environments. At the application layer, CSS supports open-source RESTful APIs and SDKs for popular languages. It also includes built-in visualization and management tools such as OpenSearch Dashboards and Cerebro. By following the instructions in this topic, you can quickly establish connections to your OpenSearch clusters to access search and analytics capabilities both securely and efficiently.

Selecting the Appropriate Tools and Clients

Select an appropriate connection method based on your role and use scenario.

Table 3-56 Comparing different cluster connection methods

Purpose	Recommended Tool	When to Use	Advantage	Details
Visualization and management	OpenSearch Dashboards	Data exploration, dashboards, cluster monitoring, development and debugging	Official visualization tool, comprehensive features, no coding required	**Connecting to Cluster Through OpenSearch Dashboards**
Cluster O&M	Cerebro	Cluster monitoring, index management, shard migration	Official visualization tool for cluster monitoring and operations	**Connecting to a Cluster Through Cerebro**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Purpose	Recommended Tool	When to Use	Advantage	Details
Application development	Java SDK	Connecting from a Java-based system	High performance; connection pool and high-availability retry supported	**Connecting to a Cluster Using the High Level REST Client** **Connecting to a Cluster Using the Low Level REST Client** **Connecting to a Cluster Through the Elasticsearch Java API Client**
	Python SDK	Connecting from a Python-based system		**Connecting to a Cluster Using Python**
Lightweight debugging	cURL	Automated scripts, ad-hoc connectivity testing	Cross-platform access, no client installation	**Connecting to a Cluster Using cURL Commands**

Obtaining the cluster access address

After the network is configured for a cluster (), to connect to it via a client, you must first obtain its access address. **Table 3-57** describes how to obtain the cluster address under different networking modes.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-57 Obtaining the cluster access address

Networking Mode	Operation Guide	Configuration Guide
Private network access	Connect to your cluster through the private network address of a directly connected node. This allows for low-cost, high-performance private network access. To obtain the private network address of one or more nodes in your cluster, perform the following steps: 1. Log in to the **CSS management console**. 2. In the navigation pane on the left, choose **Clusters > OpenSearch**. 3. In the cluster list, obtain the target cluster's internal network address from the **Internal Network Address** column. Typical address format: *<host>:<port>* or *<host>:<port>,<host>:<port>*. Example: **10.62.179.32:9200,10.62.179.33:9200**. If the cluster has client nodes, only the IP addresses and ports of all the client nodes are displayed. Otherwise, the IP addresses and ports of all data nodes and cold data nodes are displayed.	Default
Public network access	Enable public network access for a security-mode cluster that uses HTTPS, and configure a whitelist for access control. To obtain the cluster's public network address, perform the following steps: 1. Log in to the **CSS management console**. 2. In the navigation pane on the left, choose **Clusters > OpenSearch**. 3. In the cluster list, click the name of the target cluster. The cluster information page is displayed. 4. Click the **Overview** tab. In the **Network Information** area, find and record the IP address used for public network access.	**Configuring Public Network Access**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Networking Mode	Operation Guide	Configuration Guide
Dedicated load balancer	Bind a dedicated load balancer to your cluster for reliable load balancing. This mode supports two-way authentication via a custom certificate. To obtain the load balancer's private or public IP address, perform the following steps: 1. Log in to the **CSS management console**. 2. In the navigation pane on the left, choose **Clusters > OpenSearch**. 3. In the cluster list, click the name of the target cluster. The cluster information page is displayed. 4. Choose **Cluster Access > Load Balancing**. On the **OpenSearch** tab, find and record the private or public IP address of the target load balancer.	**Configuring a Dedicated Load Balancer**

3.7.2 Connecting to Cluster Through OpenSearch Dashboards

OpenSearch Dashboards is the data visualization and exploration platform for OpenSearch. It enables in-depth data analysis and interactive visualization. In CSS, OpenSearch Dashboards is pre-built for each OpenSearch cluster. You can start OpenSearch Dashboards with one click, without installing anything. OpenSearch Dashboards offers a comprehensive suite of dashboard features and visualization tools. It seamlessly integrates OpenSearch's analytical capabilities, supporting the full analytics process from data exploration to actionable business insights.

OpenSearch Dashboards supports multiple access methods. Steps needed to log in to an OpenSearch cluster vary depending on the access method you choose. See **Table 3-58**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-58 Methods for logging in to an OpenSearch cluster through OpenSearch Dashboards

OpenSearch Dashboards Access Method	Constraints	Details
One-click access to OpenSearch Dashboards from the service console	N/A	**Logging In to an OpenSearch Cluster by Accessing OpenSearch Dashboards Through the Console**
Accessing OpenSearch Dashboards using a private IP address	The client that accesses OpenSearch Dashboards must be able to reach the private IP address of OpenSearch Dashboards.	**Logging In to an OpenSearch Cluster by Accessing OpenSearch Dashboards Using a Private Network Address**

Constraints Concerning the Use of OpenSearch Dashboards

You can customize the username, role name, and tenant name in OpenSearch Dashboards.

Logging In to an OpenSearch Cluster by Accessing OpenSearch Dashboards Through the Console

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
   - Non-security mode cluster: The OpenSearch Dashboards console is displayed.
   - Security-mode cluster: Enter the username and password on the login page and click **Log In** to go to the OpenSearch Dashboards console. The default username is **admin** and the password is the one specified during cluster creation.
4. After logging in, you can use OpenSearch Dashboards to manage the cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Logging In to an OpenSearch Cluster by Accessing OpenSearch Dashboards Using a Private Network Address

> **CAUTION**
>
> The client that accesses OpenSearch Dashboards must be able to reach the private IP address of OpenSearch Dashboards.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Overview** tab. In the **Network Information** area, record **Private IPv4 Address**.
5. Obtain the private IP address of OpenSearch Dashboards.

   By changing the port number in the cluster's private IP address from **9200** to **5601**, you get the private IP address of OpenSearch Dashboards. For example, if the cluster's private IPv4 address is **xx.xx.xx.xx:9200**, the private IP address of OpenSearch Dashboards is **xx.xx.xx.xx:5601**.
6. (Optional) Configure a return route. To enable a client to access the cluster's OpenSearch Dashboards console across different VPCs, configure a route for the OpenSearch cluster.
   a. Connect the client and OpenSearch Dashboards through a Direct Connect or VPC peering connection.
   b. Configure the route connecting the OpenSearch cluster and the client.

      On the **Overview** tab, find **Cluster Route** in the **Network Information** area, and click **Add Route** under it. In the displayed dialog box, set **IP Address** and **Subnet Mask**.
7. Access OpenSearch Dashboards.
   - If the security mode is enabled for an OpenSearch cluster, the OpenSearch Dashboards access address is **https://{Dashboards private network address}**, for example, **https://xx.xx.xx.xx:5601**.
   - If the security mode is disabled, the OpenSearch Dashboards access address is **http://{Dashboards private network address}**, for example, **http://xx.xx.xx.xx:5601**.

Related Documents

- For how to use in-house developed OpenSearch Dashboards to access OpenSearch clusters in CSS, see **How Do I Connect In-house Developed OpenSearch Dashboards to an OpenSearch Cluster in CSS?**.
- For routine O&M tasks (such as shard adjustment, index management, and performance monitoring), you are advised to use Cerebro for cluster login. For details, see **Connecting to a Cluster Through Cerebro**.

3.7.3 Connecting to a Cluster Through Cerebro

Cerebro is an open-source OpenSearch web admin tool, designed for real-time monitoring and efficient cluster operations. Cerebro works for OpenSearch just as

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

well. In CSS, Cerebro is pre-built for each OpenSearch cluster. You can start Cerebro with one click, without installing anything. Cerebro allows you to quickly check a cluster's health status, node distribution, and index details. For routine O&M tasks, including shard adjustment, index management, and performance monitoring, we recommend using Cerebro. Its intuitive cluster topology views and comprehensive management features streamline O&M workflows, reducing complexity while improving efficiency. Cerebro is suitable for different roles, including developers, O&M teams, and data analysts.

Accessing a Cluster Through Cerebro on the CSS Console

The OpenSearch clusters created in CSS provide a built-in Cerebro component, using which you can quickly access these clusters.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. Find the target cluster, and choose **More** > **Cerebro** in the **Operation** column. The Cerebro login page is displayed.
   - Non-security cluster: Click the cluster name on the Cerebro login page to go to the Cerebro console.
   - Security cluster: Click the cluster name on the Cerebro login page, enter the username and password, and click **Authenticate** to go to the Cerebro console. The default username is **admin** and the password is the one specified during cluster creation.
4. After the login is successful, you can access clusters through Cerebro.

Accessing a Cluster Using an In-House Built Cerebro

CAUTION

When using in-house built Cerebro to access a cluster, make sure the network between them is connected.

1. Obtain the cluster access address. For details about how to obtain the cluster access address for different network configurations, see **Obtaining the cluster access address**.
2. Start Cerebro, and enter the cluster access address.
   - For a cluster in security mode, enter https://*access address*:9200.

     To log in to a cluster in security mode, you need to provide a username and password.
   - For a cluster in non-security mode, enter http://*access address*:9200.

3.7.4 Connecting to a Cluster Using Java

3.7.4.1 Connecting to a Cluster Using the High Level REST Client

Working directly with OpenSearch's diverse APIs can be complex and inefficient. CSS simplifies this by providing a High Level REST Client, offering a more efficient

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

way for querying and managing your data. The High Level REST Client encapsulates OpenSearch APIs. You only need to construct the required structures to access an OpenSearch cluster. This simplifies the process of working with OpenSearch clusters, as well as the development process. For details about how to use the REST Client, see **Java high-level REST client**.

Prerequisites

- The target OpenSearch cluster is available.
- The server that runs the Java code can communicate with the OpenSearch cluster.
- Depending on the network configuration method used, obtain the cluster access address. For details, see **Obtaining the cluster access address**.
- Java has been installed on the server and the JDK version is 11 or later. Download JDK 11 from **Java Archive Downloads**.

Introducing Dependencies

Introduce the required Java dependencies on the server where you run Java code.

CSS allows you to use the Elasticsearch 7.10.2 Java client to connect to OpenSearch clusters. To ensure better compatibility, however, you are advised to use an OpenSearch Java client that has the same version as the target OpenSearch cluster.

If your High Level REST Client version is later than the OpenSearch cluster version and there are incompatibility issues with some requests, you can use **RestHighLevelClient.getLowLevelClient()** to obtain the Low Level REST Client and customize OpenSearch requests. For details, see **Connecting to a Cluster Using the Low Level REST Client**.

Select a reference example based on the Java client you use.

Scenario 1 (recommended): Use the OpenSearch Java client to access an OpenSearch cluster and introduce dependencies.

- Maven:

  Replace 1.3.6 with the actual Java client version.

  ```
  <dependency>
     <groupId>org.opensearch.client</groupId>
     <artifactId>opensearch-rest-high-level-client</artifactId>
     <version>1.3.6</version>
  </dependency>
  ```

- Gradle:

  Replace 1.3.6 with the actual Java client version.

  ```
  compile group: 'org.opensearch.client', name: 'opensearch-rest-high-level-client', version: '1.3.6'
  ```

Scenario 2: Use the Elasticsearch 7.10.2 Java client to access an OpenSearch cluster and introduce dependencies.

- Maven:

  ```
  <dependency>
     <groupId>org.elasticsearch.client</groupId>
     <artifactId>elasticsearch-rest-high-level-client</artifactId>
     <version>7.10.2</version>
  </dependency>
  ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
<dependency>
   <groupId>org.elasticsearch</groupId>
   <artifactId>elasticsearch</artifactId>
   <version>7.10.2</version>
</dependency>
```

- Gradle:

```
compile group: 'org.elasticsearch.client', name: 'elasticsearch-rest-high-level-client', version: '7.10.2'
```

Accessing a Cluster

The sample code varies depending on the Java client and the security mode settings of the target OpenSearch cluster. Select the right reference document based on your service scenario.

Table 3-59 Cluster access scenarios

Java Client	OpenSearch Cluster Security-Mode Settings	Security Certificate	Details
OpenSearch	Non-security mode	-	**Connecting to a Non-Security Mode Cluster Using the OpenSearch High Level REST Client**
OpenSearch	Security mode + HTTP Security mode + HTTPS	No	**Connecting to a Security-Mode Cluster Using the OpenSearch High Level REST Client (Without a Certificate)**
OpenSearch	Security mode + HTTPS	Yes	**Connecting to a Security-Mode Cluster Using the OpenSearch High Level REST Client (With a Certificate)**
Elasticsearch 7.10.2	Non-security mode	-	**Connecting to a Non-Security Mode Cluster Using the Elasticsearch High Level REST Client**
Elasticsearch 7.10.2	Security mode + HTTP Security mode + HTTPS	No	**Connecting to a Security-Mode Cluster Using the Elasticsearch High Level REST Client (Without a Certificate)**
Elasticsearch 7.10.2	Security mode + HTTPS	Yes	**Connecting to a Security-Mode Cluster Using the Elasticsearch High Level REST Client (With a Certificate)**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Connecting to a Non-Security Mode Cluster Using the OpenSearch High Level REST Client

Use the OpenSearch High Level REST Client to connect to an OpenSearch cluster for which the security mode is disabled, and query whether the **test** index exists. The sample code is as follows:

```
import org.apache.http.HttpHost;
import org.opensearch.client.RequestOptions;
import org.opensearch.client.RestClient;
import org.opensearch.client.RestClientBuilder;
import org.opensearch.client.RestHighLevelClient;
import org.opensearch.client.indices.GetIndexRequest;
import java.io.IOException;
import java.util.Arrays;
import java.util.List;
public class RestHighLevelClientExample {
    public static void main(String[] args) throws IOException {
    List<String> host = Arrays.asList("{Cluster access address}");
        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, 9200, "http"));
        final RestHighLevelClient client = new RestHighLevelClient(builder);
        GetIndexRequest indexRequest = new GetIndexRequest("test");
        boolean exists = client.indices().exists(indexRequest, RequestOptions.DEFAULT);
        System.out.println(exists);
        client.close();
    }
    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }
}
```

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the OpenSearch High Level REST Client (Without a Certificate)

Use the OpenSearch High Level REST Client to connect to a security-mode OpenSearch cluster (HTTP or HTTPS) without loading a security certificate, and query whether the **test** index exists. The sample code is as follows:

```
import org.apache.http.HttpHost;
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.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.opensearch.action.admin.cluster.health.ClusterHealthRequest;
import org.opensearch.action.admin.cluster.health.ClusterHealthResponse;
import org.opensearch.client.RequestOptions;
import org.opensearch.client.RestClient;
import org.opensearch.client.RestClientBuilder;
import org.opensearch.client.RestHighLevelClient;
import org.opensearch.client.indices.GetIndexRequest;
import org.opensearch.common.Nullable;
import java.io.IOException;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
/**
 * Use the High Level REST Client to connect to a security-mode cluster (without a certificate).
 */
public class RestHighLevelClientExample {
    /**
     * Create a class for the client. Define the create function.
     */
    public static RestHighLevelClient create(List<String> host, int port, String protocol, int connectTimeout,
int connectionRequestTimeout, int socketTimeout, String username, String password) throws IOException{
        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 | NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new NullHostNameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
            credentialsProvider);
        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientConfigCallback);
        final RestHighLevelClient client = new RestHighLevelClient(builder);
        logger.info("opensearch rest client build success {} ", client);

        ClusterHealthRequest request = new ClusterHealthRequest();
        ClusterHealthResponse response = client.cluster().health(request, RequestOptions.DEFAULT);
        logger.info("opensearch rest client health response {} ", response);
        return client;
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }

    /**
     * Configure trustAllCerts to ignore the certificate configuration.
     */
    public 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 {
            }
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            @Override
            public X509Certificate[] getAcceptedIssuers() {
                return null;
            }
        }
    };

    private static final Logger logger = LogManager.getLogger(RestHighLevelClientExample.class);

    static class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
        @Nullable
        private final CredentialsProvider credentialsProvider;
        /**
         * The {@link SSLIOSessionStrategy} for all requests to enable SSL / TLS encryption.
         */
        private final SSLIOSessionStrategy sslStrategy;
        /**
         * Create a new {@link SecuredHttpClientConfigCallback}.
         *
         * @param credentialsProvider The credential provider, if a username/password have been supplied
         * @param sslStrategy         The SSL strategy, if SSL / TLS have been supplied
         * @throws NullPointerException if {@code sslStrategy} is {@code null}
         */
        SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
            @Nullable final CredentialsProvider credentialsProvider) {
            this.sslStrategy = Objects.requireNonNull(sslStrategy);
            this.credentialsProvider = credentialsProvider;
        }
        /**
         * Get the {@link CredentialsProvider} that will be added to the HTTP client.
         *
         * @return Can be {@code null}.
         */
        @Nullable
        CredentialsProvider getCredentialsProvider() {
            return credentialsProvider;
        }
        /**
         * Get the {@link SSLIOSessionStrategy} that will be added to the HTTP client.
         *
         * @return Never {@code null}.
         */
        SSLIOSessionStrategy getSSLStrategy() {
            return sslStrategy;
        }
        /**
         * Sets the {@linkplain HttpAsyncClientBuilder#setDefaultCredentialsProvider(CredentialsProvider)
credential provider},
         *
         * @param httpClientBuilder The client to configure.
         * @return Always {@code httpClientBuilder}.
         */
        @Override
        public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder httpClientBuilder) {
            // enable SSL / TLS
            httpClientBuilder.setSSLStrategy(sslStrategy);
            // enable user authentication
            if (credentialsProvider != null) {
                httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
            }
            return httpClientBuilder;
        }
    }

    public static class NullHostNameVerifier implements HostnameVerifier {
        @Override
        public boolean verify(String arg0, SSLSession arg1) {
            return true;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        }
    }

    /**
     * The following is an example of the main function. Call the create function to create a client and check whether the test index exists.
     */
    public static void main(String[] args) throws IOException {
        RestHighLevelClient client = create(Arrays.asList("{host}"), 9200, "https", 30000, 30000, 30000, "username", "password");
        GetIndexRequest indexRequest = new GetIndexRequest("test");
        boolean exists = client.indices().exists(indexRequest, RequestOptions.DEFAULT);
        System.out.println(exists);
        client.close();
    }
}
```

Table 3-60 Variables

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol, which can be **http** or **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the OpenSearch High Level REST Client (With a Certificate)

Use the OpenSearch High Level REST Client to connect to a security-mode OpenSearch cluster that uses HTTPS with a security certificate loaded, and query whether the **test** index exists. The sample code is as follows:

CAUTION

For how to obtain and upload a security certificate, see **Obtaining and Uploading a Security Certificate**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.opensearch.action.admin.cluster.health.ClusterHealthRequest;
import org.opensearch.action.admin.cluster.health.ClusterHealthResponse;
import org.opensearch.client.RequestOptions;
import org.opensearch.client.RestClient;
import org.opensearch.client.RestClientBuilder;
import org.opensearch.client.RestHighLevelClient;
import org.opensearch.client.indices.GetIndexRequest;
import org.opensearch.common.Nullable;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;
/**
 * Use the Hive Level REST Client to connect to a security-mode cluster (with an HTTPS certificate).
 */
public class RestHighLevelClientExample {
    public static RestHighLevelClient create(List<String> host, int port, String protocol, int connectTimeout,
int connectionRequestTimeout, int socketTimeout, String username, String password, String certFilePath,
        String certPassword) throws IOException {

        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials(username,
password));
        SSLContext sc = null;
        try {
            TrustManager[] tm = {new MyX509TrustManager(certFilePath, certPassword)};
            sc = SSLContext.getInstance("SSL", "SunJSSE");
            //You can also use SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
            sc.init(null, tm, new SecureRandom());
        } catch (Exception e) {
            e.printStackTrace();
        }

        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new NoopHostnameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
            credentialsProvider);

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientConfigCallback);
        final RestHighLevelClient client = new RestHighLevelClient(builder);
        logger.info("opensearch rest client build success {} ", client);

        ClusterHealthRequest request = new ClusterHealthRequest();
        ClusterHealthResponse response = client.cluster().health(request, RequestOptions.DEFAULT);
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
    logger.info("opensearch rest client health response {} ", response);
    return client;
}

/**
 * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
 */
public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
    return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
}

/**
 * SecuredHttpClientConfigCallback class definition
 */
static class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
    @Nullable
    private final CredentialsProvider credentialsProvider;

    private final SSLIOSessionStrategy sslStrategy;

    SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
        @Nullable final CredentialsProvider credentialsProvider) {
        this.sslStrategy = Objects.requireNonNull(sslStrategy);
        this.credentialsProvider = credentialsProvider;
    }

    @Nullable
    CredentialsProvider getCredentialsProvider() {
        return credentialsProvider;
    }

    SSLIOSessionStrategy getSSLStrategy() {
        return sslStrategy;
    }

    @Override
    public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder httpClientBuilder) {
        httpClientBuilder.setSSLStrategy(sslStrategy);
        if (credentialsProvider != null) {
            httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
        }
        return httpClientBuilder;
    }
}

private static final Logger logger = LogManager.getLogger(RestHighLevelClientExample.class);

public static class MyX509TrustManager implements X509TrustManager {
    X509TrustManager sunJSSEX509TrustManager;

    MyX509TrustManager(String certFilePath, String certPassword) throws Exception {
        File file = new File(certFilePath);
        if (!file.isFile()) {
            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, certPassword.toCharArray());
        TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
        tmf.init(ks);
        TrustManager[] tms = tmf.getTrustManagers();
        for (TrustManager tm : tms) {
            if (tm instanceof X509TrustManager) {
                sunJSSEX509TrustManager = (X509TrustManager) tm;
                return;
            }
        }
        throw new Exception("Couldn't initialize");
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        }

        @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 new X509Certificate[0];
        }
    }

    /**
     * The following is an example of the main function. Call the create function to create a client and check whether the test index exists.
     */
    public static void main(String[] args) throws IOException {
        RestHighLevelClient client = create(Arrays.asList("{host}"), 9200, "https", 30000, 30000, 30000, "username", "password", "certFilePath", "certPassword");
        GetIndexRequest indexRequest = new GetIndexRequest("test");
        boolean exists = client.indices().exists(indexRequest, RequestOptions.DEFAULT);
        System.out.println(exists);
        client.close();
    }
}
```

Table 3-61 Function parameters

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol. Set this parameter to **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.
certFilePath	Path for storing the security certificate.
certPassword	Password of the security certificate.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Non-Security Mode Cluster Using the Elasticsearch High Level REST Client

Use the Elasticsearch 7.10.2 High Level REST Client to connect to an OpenSearch cluster for which the security mode is disabled, and query whether the **test** index exists. The sample code is as follows:

```
import org.apache.http.HttpHost;
import org.elasticsearch.client.RequestOptions;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.client.indices.GetIndexRequest;
import java.io.IOException;
import java.util.Arrays;
import java.util.List;
public class RestHighLevelClientExample {
    public static void main(String[] args) throws IOException {
    List<String> host = Arrays.asList("{Cluster access address}");
        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, 9200, "http"));
        final RestHighLevelClient client = new RestHighLevelClient(builder);
        GetIndexRequest indexRequest = new GetIndexRequest("test");
        boolean exists = client.indices().exists(indexRequest, RequestOptions.DEFAULT);
        System.out.println(exists);
        client.close();
    }
    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }
}
```

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the Elasticsearch High Level REST Client (Without a Certificate)

Use the Elasticsearch 7.10.2 High Level REST Client to connect to a security-mode OpenSearch cluster (HTTP or HTTPS) without loading a security certificate, and query whether the **test** index exists. The sample code is as follows:

```
import org.apache.http.HttpHost;
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.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthRequest;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthResponse;
import org.elasticsearch.client.RequestOptions;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.client.indices.GetIndexRequest;
import org.elasticsearch.common.Nullable;

import java.io.IOException;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
/**
 * Use the High Level REST Client to connect to a security-mode cluster (without a certificate).
 */
public class RestHighLevelClientExample {
    /**
     * Create a class for the client. Define the create function.
     */
    public static RestHighLevelClient create(List<String> host, int port, String protocol, int connectTimeout,
int connectionRequestTimeout, int socketTimeout, String username, String password) throws IOException{
        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 | NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new NullHostNameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
            credentialsProvider);
        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientConfigCallback);
        final RestHighLevelClient client = new RestHighLevelClient(builder);
        logger.info("es rest client build success {} ", client);

        ClusterHealthRequest request = new ClusterHealthRequest();
        ClusterHealthResponse response = client.cluster().health(request, RequestOptions.DEFAULT);
        logger.info("es rest client health response {} ", response);
        return client;
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }

    /**
     * Configure trustAllCerts to ignore the certificate configuration.
     */
    public static TrustManager[] trustAllCerts = new TrustManager[] {
        new X509TrustManager() {
            @Override
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            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;
            }
        }
    };

    private static final Logger logger = LogManager.getLogger(RestHighLevelClientExample.class);

    static class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
        @Nullable
        private final CredentialsProvider credentialsProvider;
        /**
         * The {@link SSLIOSessionStrategy} for all requests to enable SSL / TLS encryption.
         */
        private final SSLIOSessionStrategy sslStrategy;
        /**
         * Create a new {@link SecuredHttpClientConfigCallback}.
         *
         * @param credentialsProvider The credential provider, if a username/password have been supplied
         * @param sslStrategy         The SSL strategy, if SSL / TLS have been supplied
         * @throws NullPointerException if {@code sslStrategy} is {@code null}
         */
        SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
            @Nullable final CredentialsProvider credentialsProvider) {
            this.sslStrategy = Objects.requireNonNull(sslStrategy);
            this.credentialsProvider = credentialsProvider;
        }
        /**
         * Get the {@link CredentialsProvider} that will be added to the HTTP client.
         *
         * @return Can be {@code null}.
         */
        @Nullable
        CredentialsProvider getCredentialsProvider() {
            return credentialsProvider;
        }
        /**
         * Get the {@link SSLIOSessionStrategy} that will be added to the HTTP client.
         *
         * @return Never {@code null}.
         */
        SSLIOSessionStrategy getSSLStrategy() {
            return sslStrategy;
        }
        /**
         * Sets the {@linkplain HttpAsyncClientBuilder#setDefaultCredentialsProvider(CredentialsProvider)
credential provider},
         *
         * @param httpClientBuilder The client to configure.
         * @return Always {@code httpClientBuilder}.
         */
        @Override
        public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder httpClientBuilder) {
            // enable SSL / TLS
            httpClientBuilder.setSSLStrategy(sslStrategy);
            // enable user authentication
            if (credentialsProvider != null) {
                httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
            }
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            return httpClientBuilder;
        }
    }

    public static class NullHostNameVerifier implements HostnameVerifier {
        @Override
        public boolean verify(String arg0, SSLSession arg1) {
            return true;
        }
    }

    /**
     * The following is an example of the main function. Call the create function to create a client and check whether the test index exists.
     */
    public static void main(String[] args) throws IOException {
        RestHighLevelClient client = create(Arrays.asList("{host}"), 9200, "https", 30000, 30000, 30000, "username", "password");
        GetIndexRequest indexRequest = new GetIndexRequest("test");
        boolean exists = client.indices().exists(indexRequest, RequestOptions.DEFAULT);
        System.out.println(exists);
        client.close();
    }
}
```

Table 3-62 Variables

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol, which can be **http** or **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Connecting to a Security-Mode Cluster Using the Elasticsearch High Level REST Client (With a Certificate)

Use the Elasticsearch 7.10.2 High Level REST Client to connect to a security-mode OpenSearch cluster that uses HTTPS with a security certificate loaded, and query whether the **test** index exists. The sample code is as follows:

CAUTION

For how to obtain and upload a security certificate, see **Obtaining and Uploading a Security Certificate**.

```
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthRequest;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthResponse;
import org.elasticsearch.client.RequestOptions;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.client.indices.GetIndexRequest;
import org.elasticsearch.common.Nullable;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;
/**
 * Use the Hive Level REST Client to connect to a security-mode cluster (with an HTTPS certificate).
 */
public class RestHighLevelClientExample {
    public static RestHighLevelClient create(List<String> host, int port, String protocol, int connectTimeout,
int connectionRequestTimeout, int socketTimeout, String username, String password, String certFilePath,
        String certPassword) throws IOException {

        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials(username,
password));
        SSLContext sc = null;
        try {
            TrustManager[] tm = {new MyX509TrustManager(certFilePath, certPassword)};
            sc = SSLContext.getInstance("SSL", "SunJSSE");
            //You can also use SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
            sc.init(null, tm, new SecureRandom());
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        } catch (Exception e) {
            e.printStackTrace();
        }

        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new NoopHostnameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
            credentialsProvider);

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientConfigCallback);
        final RestHighLevelClient client = new RestHighLevelClient(builder);
        logger.info("es rest client build success {} ", client);

        ClusterHealthRequest request = new ClusterHealthRequest();
        ClusterHealthResponse response = client.cluster().health(request, RequestOptions.DEFAULT);
        logger.info("es rest client health response {} ", response);
        return client;
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }

    /**
     * SecuredHttpClientConfigCallback class definition
     */
    static class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
        @Nullable
        private final CredentialsProvider credentialsProvider;

        private final SSLIOSessionStrategy sslStrategy;

        SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
            @Nullable final CredentialsProvider credentialsProvider) {
            this.sslStrategy = Objects.requireNonNull(sslStrategy);
            this.credentialsProvider = credentialsProvider;
        }

        @Nullable
        CredentialsProvider getCredentialsProvider() {
            return credentialsProvider;
        }

        SSLIOSessionStrategy getSSLStrategy() {
            return sslStrategy;
        }

        @Override
        public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder httpClientBuilder) {
            httpClientBuilder.setSSLStrategy(sslStrategy);
            if (credentialsProvider != null) {
                httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
            }
            return httpClientBuilder;
        }
    }

    private static final Logger logger = LogManager.getLogger(RestHighLevelClientExample.class);

    public static class MyX509TrustManager implements X509TrustManager {
        X509TrustManager sunJSSEX509TrustManager;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        MyX509TrustManager(String certFilePath, String certPassword) throws Exception {
            File file = new File(certFilePath);
            if (!file.isFile()) {
                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, certPassword.toCharArray());
            TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
            tmf.init(ks);
            TrustManager[] tms = tmf.getTrustManagers();
            for (TrustManager tm : tms) {
                if (tm instanceof X509TrustManager) {
                    sunJSSEX509TrustManager = (X509TrustManager) tm;
                    return;
                }
            }
            throw new Exception("Couldn't initialize");
        }

        @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 new X509Certificate[0];
        }
    }

    /**
     * The following is an example of the main function. Call the create function to create a client and check whether the test index exists.
     */
    public static void main(String[] args) throws IOException {
        RestHighLevelClient client = create(Arrays.asList("{host}"), 9200, "https", 30000, 30000, 30000, "username", "password", "certFilePath", "certPassword");
        GetIndexRequest indexRequest = new GetIndexRequest("test");
        boolean exists = client.indices().exists(indexRequest, RequestOptions.DEFAULT);
        System.out.println(exists);
        client.close();
    }
}
```

Table 3-63 Function parameters

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol. Set this parameter to **https**.
connectTimeout	Connection timeout (in ms).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.
certFilePath	Path for storing the security certificate.
certPassword	Password of the security certificate.

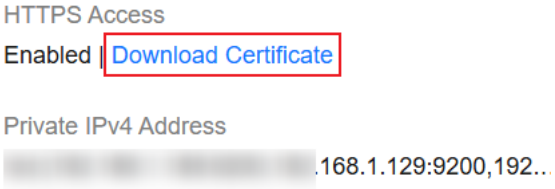
This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

Obtaining and Uploading a Security Certificate

To access a security-mode OpenSearch cluster that uses HTTPS, a security certificate must be loaded. Perform the following steps to obtain the security certificate and upload it to the client:

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

      Figure 3-15 Downloading a security certificate

2. Convert the security certificate **CloudSearchService.cer**. Upload the downloaded security certificate to the client and use keytool to convert the **.cer** certificate into a **.jks** certificate that can be read by Java.
   - In Linux, run the following command to convert the certificate:
     ```
     keytool -import -alias newname -keystore ./truststore.jks -file ./CloudSearchService.cer
     ```
   - In Windows, run the following command to convert the certificate:
     ```
     keytool -import -alias newname -keystore .\truststore.jks -file .\CloudSearchService.cer
     ```

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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

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

3.7.4.2 Connecting to a Cluster Using the Low Level REST Client

When querying and managing data in OpenSearch, sometimes you may find the High Level REST Client inadequate. For instance, due to inherent limitations, the High Level REST Client may not be able to execute complex custom requests. In this case, you may consider using the Low Level REST Client, which encapsulates OpenSearch APIs. You only need to construct the required request structures to access an OpenSearch cluster. This simplifies the process of working with OpenSearch clusters. The Low Level REST Client allows you to customize the request structure, which is more flexible and supports all the request formats of OpenSearch, such as GET, POST, DELETE, and HEAD.

You can use the Low Level REST Client to access an OpenSearch cluster in either of the following ways:

- Create the Low Level Rest Client directly.
- Create the High Level Rest Client and then call getLowLevelClient() to obtain the Low Level Rest Client. That is, use **RestHighLevelClient.getLowLevelClient()** to obtain the Low Level Rest Client.

How do I determine when to use which method? If you need to execute highly customized requests, create the Low Level Rest Client directly. If you are already using the High Level Rest Client, you can call the getLowLevelClient() method to obtain the Low Level Rest Client. This simplifies code complexity.

Prerequisites

- The target OpenSearch cluster is available.
- The server that runs the Java code can communicate with the OpenSearch cluster.
- Depending on the network configuration method used, obtain the cluster access address. For details, see **Obtaining the cluster access address**.
- Java has been installed on the server and the JDK version is 11 or later. Download JDK 11 from **Java Archive Downloads**.

Introducing Dependencies

Introduce the required Java dependencies on the server where you run Java code.

CSS allows you to use the Elasticsearch 7.10.2 Java client to connect to OpenSearch clusters. To ensure better compatibility, however, you are advised to use an OpenSearch Java client that has the same version as the target OpenSearch cluster.

Select a reference example based on the Java client you use.

- Scenario 1 (recommended): Use the OpenSearch Java client to access the OpenSearch cluster and introduce the Apache version using Maven.

  Replace 1.3.6 with the actual Java client version.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
<dependency>
   <groupId>org.opensearch.client</groupId>
   <artifactId>opensearch-rest-high-level-client</artifactId>
   <version>1.3.6</version>
</dependency>
```

- Scenario 2: Use the Elasticsearch 7.10.2 Java client to access the OpenSearch cluster and introduce the Apache version using Maven.

```
<dependency>
   <groupId>org.elasticsearch.client</groupId>
   <artifactId>elasticsearch-rest-high-level-client</artifactId>
   <version>7.10.2</version>
</dependency>
<dependency>
   <groupId>org.elasticsearch</groupId>
   <artifactId>elasticsearch</artifactId>
   <version>7.10.2</version>
</dependency>
```

Accessing a Cluster

The sample code varies depending on the Java client and the security mode settings of the target OpenSearch cluster. Select the right reference document based on your service scenario.

Table 3-64 Cluster access scenarios

Java Client	How Low Level REST Client Is Created	OpenSearch Cluster Security-Mode Settings	Security Certificate	Details
OpenSearch	Directly create the Low Level REST Client	Non-security mode	-	**Connecting to a Non-Security Mode Cluster Using the OpenSearch Low Level REST Client**
		Security mode + HTTP Security mode + HTTPS	No	**Connecting to a Security-Mode Cluster Using the OpenSearch Low Level REST Client (Without a Certificate)**
		Security mode + HTTPS	Yes	**Connecting to a Security-Mode Cluster Using the OpenSearch Low Level REST Client (With a Certificate)**
	Create the High Level REST Client first and then call getLowLevelClient() to obtain the Low Level REST Client	Non-security mode	-	**Connecting to a Non-Security Mode Cluster Using the OpenSearch High Level REST Client**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Java Client	How Low Level REST Client Is Created	OpenSearch Cluster Security-Mode Settings	Security Certificate	Details
		Security mode + HTTP Security mode + HTTPS	No	**Connecting to a Security-Mode Cluster Using the OpenSearch High Level REST Client (Without a Certificate)**
		Security mode + HTTPS	Yes	**Connecting to a Security-Mode Cluster Using the OpenSearch High Level REST Client (With a Certificate)**
Elasticsearch 7.10.2	Directly create the Low Level REST Client	Non-security mode	-	**Connecting to a Non-Security Mode Cluster Using the Elasticsearch Low Level REST Client**
		Security mode + HTTP Security mode + HTTPS	No	**Connecting to a Security-Mode Cluster Using the Elasticsearch Low Level REST Client (Without a Certificate)**
		Security mode + HTTPS	Yes	**Connecting to a Security-Mode Cluster Using the Elasticsearch Low Level REST Client (With a Certificate)**
	Create the High Level REST Client first and then call getLowLevelClient() to obtain the Low Level REST Client	Non-security mode	-	**Connecting to a Non-Security Mode Cluster Using the Elasticsearch High Level REST Client**
		Security mode + HTTP Security mode + HTTPS	No	**Connecting to a Security-Mode Cluster Using the Elasticsearch High Level REST Client (Without a Certificate)**
		Security mode + HTTPS	Yes	**Connecting to a Security-Mode Cluster Using the Elasticsearch High Level REST Client (With a Certificate)**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Connecting to a Non-Security Mode Cluster Using the OpenSearch Low Level REST Client

Use the OpenSearch Low Level REST Client to connect to an OpenSearch cluster for which the security mode is disabled, and query whether the **test** index exists. The sample code is as follows:

```
import org.apache.http.HttpHost;
import org.opensearch.client.Request;
import org.opensearch.client.Response;
import org.opensearch.client.RestClient;
import org.opensearch.client.RestClientBuilder;

import java.io.IOException;
import java.util.Arrays;
import java.util.List;
public class RestLowLevelClientExample {
    public static void main(String[] args) throws IOException {
    List<String> host = Arrays.asList("{Cluster access address}");
        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, 9200, "http"));
        /**
         *Create the Low Level Rest Client.
         */
        RestClient lowLevelClient = builder.build();
        /**
         * Check whether the test index exists. If the index exists, 200 is returned. If the index does not exist, 
404 is returned.
         */
        Request request = new Request("HEAD", "/test");
        Response response = lowLevelClient.performRequest(request);
        System.out.println(response.getStatusLine().getStatusCode());
        lowLevelClient.close();
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }
}
```

This piece of code checks whether the **test** index exists in the cluster. If **200** (the index exists) or **404** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the OpenSearch Low Level REST Client (Without a Certificate)

Use the OpenSearch Low Level REST Client to connect to a security-mode OpenSearch cluster (HTTP or HTTPS) without loading a security certificate, and query whether the **test** index exists. The sample code is as follows:

```
import org.apache.http.HttpHost;
import org.apache.http.HttpResponse;
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.impl.client.DefaultConnectionKeepAliveStrategy;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import org.apache.http.protocol.HttpContext;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.opensearch.client.Request;
import org.opensearch.client.Response;
import org.opensearch.client.RestClient;
import org.opensearch.client.RestClientBuilder;
import org.opensearch.common.Nullable;
import java.io.IOException;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;
import java.util.concurrent.TimeUnit;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;import javax.net.ssl.X509TrustManager;

public class RestLowLevelClientExample {
    /**
     * Create a class for the client. Define the create function.
     */
    public static RestClient create(List<String> host, int port, String protocol, int connectTimeout, int
connectionRequestTimeout, int socketTimeout,  String username, String password) throws IOException {
        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 | NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new NullHostNameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
            credentialsProvider);

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientConfigCallback);
        final RestClient client = builder.build();
        logger.info("opensearch rest client build success {} ", client);
        return client;
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }

    /**
     * Configure trustAllCerts to ignore the certificate configuration.
     */
    public static TrustManager[] trustAllCerts = new TrustManager[] {
        new X509TrustManager() {
            @Override
            public void checkClientTrusted(X509Certificate[] chain, String authType) throws CertificateException
{
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            }

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

            @Override
            public X509Certificate[] getAcceptedIssuers() {
                return null;
            }
        }
    };

    /**
     * The CustomConnectionKeepAliveStrategy function is used to set the connection keepalive when there
are a large number of short connections or when there are not many data requests.
     */
    public static class CustomConnectionKeepAliveStrategy extends DefaultConnectionKeepAliveStrategy {
        public static final CustomConnectionKeepAliveStrategy INSTANCE = new
CustomConnectionKeepAliveStrategy();

        private CustomConnectionKeepAliveStrategy() {
            super();
        }

        /**
         * Maximum keepalive time (in minutes)
         * The default value is 10 minutes. You can set it based on the number of TCP connections in
TIME_WAIT state. If there are too many TCP connections, you can increase this value.
         */
        private final long MAX_KEEP_ALIVE_MINUTES = 10;

        @Override
        public long getKeepAliveDuration(HttpResponse response, HttpContext context) {
            long keepAliveDuration = super.getKeepAliveDuration(response, context);
            // <0 indicates that the keepalive period is unlimited.
            // Change the period from unlimited to a default period.
            if (keepAliveDuration < 0) {
                return TimeUnit.MINUTES.toMillis(MAX_KEEP_ALIVE_MINUTES);
            }
            return keepAliveDuration;
        }
    }

    private static final Logger logger = LogManager.getLogger(RestLowLevelClientExample.class);

    static class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
        @Nullable
        private final CredentialsProvider credentialsProvider;
        /**
         * The {@link SSLIOSessionStrategy} for all requests to enable SSL / TLS encryption.
         */
        private final SSLIOSessionStrategy sslStrategy;
        /**
         * Create a new {@link SecuredHttpClientConfigCallback}.
         *
         * @param credentialsProvider The credential provider, if a username/password have been supplied
         * @param sslStrategy         The SSL strategy, if SSL / TLS have been supplied
         * @throws NullPointerException if {@code sslStrategy} is {@code null}
         */
        SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
            @Nullable final CredentialsProvider credentialsProvider) {
            this.sslStrategy = Objects.requireNonNull(sslStrategy);
            this.credentialsProvider = credentialsProvider;
        }
        /**
         * Get the {@link CredentialsProvider} that will be added to the HTTP client.
         *
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
         * @return Can be {@code null}.
         */
        @Nullable
        CredentialsProvider getCredentialsProvider() {
            return credentialsProvider;
        }
        /**
         * Get the {@link SSLIOSessionStrategy} that will be added to the HTTP client.
         *
         * @return Never {@code null}.
         */
        SSLIOSessionStrategy getSSLStrategy() {
            return sslStrategy;
        }
        /**
         * Sets the {@linkplain HttpAsyncClientBuilder#setDefaultCredentialsProvider(CredentialsProvider)
credential provider},
         *
         * @param httpClientBuilder The client to configure.
         * @return Always {@code httpClientBuilder}.
         */
        @Override
        public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder httpClientBuilder) {
            // enable SSL / TLS
            httpClientBuilder.setSSLStrategy(sslStrategy);
            // enable user authentication
            if (credentialsProvider != null) {
                httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
            }
            return httpClientBuilder;
        }
    }

    public static class NullHostNameVerifier implements HostnameVerifier {
        @Override
        public boolean verify(String arg0, SSLSession arg1) {
            return true;
        }
    }

    /**
     * The following is an example of the main function. Call the create function to create a Low Level REST
Client and check whether the test index exists.
     */
    public static void main(String[] args) throws IOException {
        RestClient lowLevelClient = create(Arrays.asList("xxx.xxx.xxx.xxx", "xxx.xxx.xxx.xxx"), 9200, "http",
30000, 30000, 30000, "username", "password");
        Request request = new Request("HEAD", "/test");
        Response response = lowLevelClient.performRequest(request);
        System.out.println(response.getStatusLine().getStatusCode());
        lowLevelClient.close();
    }
}
```

Table 3-65 Variables

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
protocol	Connection protocol, which can be **http** or **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.

This piece of code checks whether the **test** index exists in the cluster. If **200** (the index exists) or **404** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the OpenSearch Low Level REST Client (With a Certificate)

Use the OpenSearch Low Level REST Client to connect to a security-mode OpenSearch cluster that uses HTTPS with a security certificate loaded, and query whether the **test** index exists. The sample code is as follows:

CAUTION

For how to obtain and upload a security certificate, see **Obtaining and Uploading a Security Certificate**.

```
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.opensearch.client.Request;
import org.opensearch.client.Response;
import org.opensearch.client.RestClient;
import org.opensearch.client.RestClientBuilder;
import org.opensearch.common.Nullable;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import java.util.List;
import java.util.Objects;

import javax.net.ssl.SSLContext;import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;

public class RestLowLevelClientExample {

   private static final Logger logger = LogManager.getLogger(RestLowLevelClientExample.class);

   /**
    * Create a class for the client. Define the create function.
    */
   public static RestClient create(List<String> host, int port, String protocol, int connectTimeout, int
connectionRequestTimeout, int socketTimeout, String username, String password, String certFilePath, String
certPassword) throws IOException {
       final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
       credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials(username,
password));
       SSLContext sc = null;
       try {
           TrustManager[] tm = {new MyX509TrustManager(certFilePath, certPassword)};
           sc = SSLContext.getInstance("SSL", "SunJSSE");
           //You can also use SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
           sc.init(null, tm, new SecureRandom());
       } catch (Exception e) {
           e.printStackTrace();
       }

       SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new NoopHostnameVerifier());
       SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
           credentialsProvider);

       RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
           .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
               .setConnectionRequestTimeout(connectionRequestTimeout)
               .setSocketTimeout(socketTimeout))
           .setHttpClientConfigCallback(httpClientConfigCallback);
       final RestClient client = builder.build();
       logger.info("opensearch rest client build success {} ", client);
       return client;
   }

   /**
    * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
    */
   public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
       return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);}

   static class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
       @Nullable
       private final CredentialsProvider credentialsProvider;

       private final SSLIOSessionStrategy sslStrategy;

       SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
           @Nullable final CredentialsProvider credentialsProvider) {
           this.sslStrategy = Objects.requireNonNull(sslStrategy);
           this.credentialsProvider = credentialsProvider;
       }

       @Nullable
       CredentialsProvider getCredentialsProvider() {
           return credentialsProvider;
       }

       SSLIOSessionStrategy getSSLStrategy() {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            return sslStrategy;
        }

        @Override
        public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder httpClientBuilder) {
            httpClientBuilder.setSSLStrategy(sslStrategy);
            if (credentialsProvider != null) {
                httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
            }
            return httpClientBuilder;
        }}

    public static class MyX509TrustManager implements X509TrustManager {
        X509TrustManager sunJSSEX509TrustManager;

        MyX509TrustManager(String certFilePath, String certPassword) throws Exception {
            File file = new File(certFilePath);
            if (!file.isFile()) {
                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, certPassword.toCharArray());
            TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
            tmf.init(ks);
            TrustManager[] tms = tmf.getTrustManagers();
            for (TrustManager tm : tms) {
                if (tm instanceof X509TrustManager) {
                    sunJSSEX509TrustManager = (X509TrustManager) tm;
                    return;
                }
            }
            throw new Exception("Couldn't initialize");
        }

        @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 new X509Certificate[0];
        }
    }

    /**
     * The following is an example of the main function. Call the create function to create a Low Level REST
Client and check whether the test index exists.
     */
    public static void main(String[] args) throws IOException {
        RestClient lowLevelClient = create(Arrays.asList("xxx.xxx.xxx.xxx", "xxx.xxx.xxx.xxx"), 9200, "https",
30000, 30000, 30000, "username", "password", "certFilePath", "certPassword");
        Request request = new Request("HEAD", "test");
        Response response = lowLevelClient.performRequest(request);
        System.out.println(response.getStatusLine().getStatusCode());
        lowLevelClient.close();
    }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-66 Function parameters

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol. Set this parameter to **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.
certFilePath	Path for storing the security certificate.
certPassword	Password of the security certificate.

This piece of code checks whether the **test** index exists in the cluster. If **200** (the index exists) or **404** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Non-Security Mode Cluster Using the OpenSearch High Level REST Client

Use the OpenSearch High Level REST Client to obtain the Low Level REST Client by calling **getLowLevelClient()**, use the low-level client to connect to an OpenSearch cluster for which the security mode is disabled, and query whether the **test** index exists. The sample code is as follows:

```
import org.apache.http.HttpHost;
import org.opensearch.client.Request;
import org.opensearch.client.Response;
import org.opensearch.client.RestClient;
import org.opensearch.client.RestClientBuilder;
import org.opensearch.client.RestHighLevelClient;

import java.io.IOException;
import java.util.Arrays;
import java.util.List;
public class RestLowLevelClientExample {

    public static void main(String[] args) throws IOException {
    List<String> host = Arrays.asList("{Cluster access address}");
        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, 9200, "http"));
        final RestHighLevelClient restHighLevelClient = new RestHighLevelClient(builder);
        /**
         * Create a high-level client and then call getLowLevelClient() to obtain a low-level client. The code
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
differs from the client creation code only in the following line:
         */
        final RestClient lowLevelClient = restHighLevelClient.getLowLevelClient();
        /**
         * Check whether the test index exists. If the index exists, 200 is returned. If the index does not exist,
404 is returned.
         */
        Request request = new Request("HEAD", "/test");
        Response response = lowLevelClient.performRequest(request);
        System.out.println(response.getStatusLine().getStatusCode());
        lowLevelClient.close();
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }
}
```

This piece of code checks whether the **test** index exists in the cluster. If **200** (the index exists) or **404** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the OpenSearch High Level REST Client (Without a Certificate)

Use the OpenSearch High Level REST Client to obtain the Low Level REST Client by calling **getLowLevelClient()**, use the low-level client to connect to a security-mode OpenSearch cluster that uses HTTP or HTTPS without loading a security certificate, and query whether the **test** index exists. The sample code is as follows:

```
import org.apache.http.HttpHost;
import org.apache.http.HttpResponse;
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.impl.client.DefaultConnectionKeepAliveStrategy;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.http.protocol.HttpContext;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.opensearch.client.Request;
import org.opensearch.client.Response;
import org.opensearch.client.RestClient;
import org.opensearch.client.RestClientBuilder;
import org.opensearch.client.RestHighLevelClient;
import org.opensearch.common.Nullable;

import java.io.IOException;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;
import java.util.concurrent.TimeUnit;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import javax.net.ssl.TrustManager;import javax.net.ssl.X509TrustManager;

public class RestLowLevelClientExample {

    /**
     * Create a class for the client. Define the create function.
     */
    public static RestHighLevelClient create(List<String> host, int port, String protocol, int connectTimeout,
int connectionRequestTimeout, int socketTimeout,  String username, String password) throws IOException {

        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 | NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new NullHostNameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
            credentialsProvider);

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientConfigCallback);
        final RestHighLevelClient client = new RestHighLevelClient(builder);
        logger.info("opensearch rest client build success {} ", client);
        return client;
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }

    /**
     * Configure trustAllCerts to ignore the certificate configuration.
     */
    public 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;
            }
        }
    };

    /**
     * The CustomConnectionKeepAliveStrategy function is used to set the connection keepalive when there
are a large number of short connections or when there are not many data requests.
     */
    public static class CustomConnectionKeepAliveStrategy extends DefaultConnectionKeepAliveStrategy {
```

```
        public static final CustomConnectionKeepAliveStrategy INSTANCE = new
CustomConnectionKeepAliveStrategy();

        private CustomConnectionKeepAliveStrategy() {
            super();
        }

        /**
         * Maximum keepalive time (in minutes)
         * The default value is 10 minutes. You can set it based on the number of TCP connections in
TIME_WAIT state. If there are too many TCP connections, you can increase this value.
         */
        private final long MAX_KEEP_ALIVE_MINUTES = 10;

        @Override
        public long getKeepAliveDuration(HttpResponse response, HttpContext context) {
            long keepAliveDuration = super.getKeepAliveDuration(response, context);
            // <0 indicates an unlimited keepalive period.
            // Change the period from unlimited to a default period.
            if (keepAliveDuration < 0) {
                return TimeUnit.MINUTES.toMillis(MAX_KEEP_ALIVE_MINUTES);
            }
            return keepAliveDuration;
        }
    }

    private static final Logger logger = LogManager.getLogger(RestLowLevelClientExample.class);

    static class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
        @Nullable
        private final CredentialsProvider credentialsProvider;
        /**
         * The {@link SSLIOSessionStrategy} for all requests to enable SSL / TLS encryption.
         */
        private final SSLIOSessionStrategy sslStrategy;
        /**
         * Create a new {@link SecuredHttpClientConfigCallback}.
         *
         * @param credentialsProvider The credential provider, if a username/password have been supplied
         * @param sslStrategy         The SSL strategy, if SSL / TLS have been supplied
         * @throws NullPointerException if {@code sslStrategy} is {@code null}
         */
        SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
            @Nullable final CredentialsProvider credentialsProvider) {
            this.sslStrategy = Objects.requireNonNull(sslStrategy);
            this.credentialsProvider = credentialsProvider;
        }
        /**
         * Get the {@link CredentialsProvider} that will be added to the HTTP client.
         *
         * @return Can be {@code null}.
         */
        @Nullable
        CredentialsProvider getCredentialsProvider() {
            return credentialsProvider;
        }
        /**
         * Get the {@link SSLIOSessionStrategy} that will be added to the HTTP client.
         *
         * @return Never {@code null}.
         */
        SSLIOSessionStrategy getSSLStrategy() {
            return sslStrategy;
        }
        /**
         * Sets the {@linkplain HttpAsyncClientBuilder#setDefaultCredentialsProvider(CredentialsProvider)
credential provider},
         *
         * @param httpClientBuilder The client to configure.
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
         * @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;
        }
    }

    public static class NullHostNameVerifier implements HostnameVerifier {
        @Override
        public boolean verify(String arg0, SSLSession arg1) {
            return true;
        }
    }

    /**
* The following is an example of the main function. Call the create function to create the high-level client, then call the getLowLevelClient() function to obtain the low-level client, and query whether the test index exists.
     */
    public static void main(String[] args) throws IOException {
        RestHighLevelClient client = create(Arrays.asList("xxx.xxx.xxx.xxx", "xxx.xxx.xxx.xxx"), 9200, "http",
30000, 30000, 30000, "username", "password");
        RestClient lowLevelClient = client.getLowLevelClient();
        Request request = new Request("HEAD", "test");
        Response response = lowLevelClient.performRequest(request);
        System.out.println(response.getStatusLine().getStatusCode());
        lowLevelClient.close();
    }
}
```

Table 3-67 Variables

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol, which can be **http** or **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

This piece of code checks whether the **test** index exists in the cluster. If **200** (the index exists) or **404** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the OpenSearch High Level REST Client (With a Certificate)

Use the OpenSearch High Level REST Client to connect to a security-mode OpenSearch cluster that uses HTTPS with a security certificate loaded, and query whether the **test** index exists. The sample code is as follows:

CAUTION

For how to obtain and upload a security certificate, see **Obtaining and Uploading a Security Certificate**.

```
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.opensearch.action.admin.cluster.health.ClusterHealthRequest;
import org.opensearch.action.admin.cluster.health.ClusterHealthResponse;
import org.opensearch.client.Request;
import org.opensearch.client.RequestOptions;
import org.opensearch.client.Response;
import org.opensearch.client.RestClient;
import org.opensearch.client.RestClientBuilder;
import org.opensearch.client.RestHighLevelClient;
import org.opensearch.common.Nullable;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;

public class RestLowLevelClientExample {
    private static final Logger logger = LogManager.getLogger(RestLowLevelClientExample.class);

    /**
     * Create a class for the client. Define the create function.
     */
    public static RestHighLevelClient create(List<String> host, int port, String protocol, int connectTimeout,
int connectionRequestTimeout, int socketTimeout, String username, String password, String certFilePath,
String certPassword) throws IOException {
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials(username,
password));
```

```
        SSLContext sc = null;
        try {
            TrustManager[] tm = {new MyX509TrustManager(certFilePath, certPassword)};
            sc = SSLContext.getInstance("SSL", "SunJSSE");
            //You can also use SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
            sc.init(null, tm, new SecureRandom());
        } catch (Exception e) {
            e.printStackTrace();
        }

        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new NoopHostnameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
            credentialsProvider);

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientConfigCallback);
        final RestHighLevelClient client = new RestHighLevelClient(builder);
        logger.info("opensearch rest client build success {} ", client);

        ClusterHealthRequest request = new ClusterHealthRequest();
        ClusterHealthResponse response = client.cluster().health(request, RequestOptions.DEFAULT);
        logger.info("opensearch rest client health response {} ", response);
        return client;
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);}

    static class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
        @Nullable
        private final CredentialsProvider credentialsProvider;

        private final SSLIOSessionStrategy sslStrategy;

        SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
            @Nullable final CredentialsProvider credentialsProvider) {
            this.sslStrategy = Objects.requireNonNull(sslStrategy);
            this.credentialsProvider = credentialsProvider;
        }

        @Nullable
        CredentialsProvider getCredentialsProvider() {
            return credentialsProvider;
        }

        SSLIOSessionStrategy getSSLStrategy() {
            return sslStrategy;
        }

        @Override
        public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder httpClientBuilder) {
            httpClientBuilder.setSSLStrategy(sslStrategy);
            if (credentialsProvider != null) {
                httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
            }
            return httpClientBuilder;
        }}

    public static class MyX509TrustManager implements X509TrustManager {
        X509TrustManager sunJSSEX509TrustManager;

        MyX509TrustManager(String certFilePath, String certPassword) throws Exception {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            File file = new File(certFilePath);
            if (!file.isFile()) {
                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, certPassword.toCharArray());
            TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
            tmf.init(ks);
            TrustManager[] tms = tmf.getTrustManagers();
            for (TrustManager tm : tms) {
                if (tm instanceof X509TrustManager) {
                    sunJSSEX509TrustManager = (X509TrustManager) tm;
                    return;
                }
            }
            throw new Exception("Couldn't initialize");
        }

        @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 new X509Certificate[0];
        }
    }

    /**
* The following is an example of the main function. Call the create function to create the high-level client, then call the getLowLevelClient() function to obtain the low-level client, and query whether the test index exists.
     */
    public static void main(String[] args) throws IOException {
        RestHighLevelClient client = create(Arrays.asList("xxx.xxx.xxx.xxx", "xxx.xxx.xxx.xxx"), 9200, "https", 30000, 30000, 30000, "username", "password", "certFilePath", "certPassword");
        RestClient lowLevelClient = client.getLowLevelClient();
        Request request = new Request("HEAD", "test");
        Response response = lowLevelClient.performRequest(request);
        System.out.println(response.getStatusLine().getStatusCode());
        lowLevelClient.close();
    }
}
```

Table 3-68 Function parameters

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol. Set this parameter to **https**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.
certFilePath	Path for storing the security certificate.
certPassword	Password of the security certificate.

This piece of code checks whether the **test** index exists in the cluster. If **200** (the index exists) or **404** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Non-Security Mode Cluster Using the Elasticsearch Low Level REST Client

Use the Elasticsearch 7.10.2 Low Level REST Client to connect to an OpenSearch cluster for which the security mode is disabled, and query whether the **test** index exists. The sample code is as follows:

```
import org.apache.http.HttpHost;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;

import java.io.IOException;
import java.util.Arrays;
import java.util.List;
public class RestLowLevelClientExample {
    public static void main(String[] args) throws IOException {
    List<String> host = Arrays.asList("{Cluster access address}");
        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, 9200, "http"));
        /**
         *Create the Low Level Rest Client.
         */
        RestClient lowLevelClient = builder.build();
        /**
         * Check whether the test index exists. If the index exists, 200 is returned. If the index does not exist, 404 is returned.
         */
        Request request = new Request("HEAD", "/test");
        Response response = lowLevelClient.performRequest(request);
        System.out.println(response.getStatusLine().getStatusCode());
        lowLevelClient.close();
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
    }
}
```

This piece of code checks whether the **test** index exists in the cluster. If **200** (the index exists) or **404** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the Elasticsearch Low Level REST Client (Without a Certificate)

Use the Elasticsearch 7.10.2 Low Level REST Client to connect to a security-mode OpenSearch cluster (HTTP or HTTPS) without loading a security certificate, and query whether the **test** index exists. The sample code is as follows:

```
import org.apache.http.HttpHost;
import org.apache.http.HttpResponse;
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.impl.client.DefaultConnectionKeepAliveStrategy;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.http.protocol.HttpContext;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.common.Nullable;

import java.io.IOException;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;
import java.util.concurrent.TimeUnit;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;import javax.net.ssl.X509TrustManager;

public class RestLowLevelClientExample {
    /**
     * Create a class for the client. Define the create function.
     */
    public static RestClient create(List<String> host, int port, String protocol, int connectTimeout, int
connectionRequestTimeout, int socketTimeout,  String username, String password) throws IOException {
        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 | NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new NullHostNameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
            credentialsProvider);
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientConfigCallback);
        final RestClient client = builder.build();
        logger.info("es rest client build success {} ", client);
        return client;
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }

    /**
     * Configure trustAllCerts to ignore the certificate configuration.
     */
    public 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;
            }
        }
    };

    /**
     * The CustomConnectionKeepAliveStrategy function is used to set the connection keepalive when there
are a large number of short connections or when there are not many data requests.
     */
    public static class CustomConnectionKeepAliveStrategy extends DefaultConnectionKeepAliveStrategy {
        public static final CustomConnectionKeepAliveStrategy INSTANCE = new
CustomConnectionKeepAliveStrategy();

        private CustomConnectionKeepAliveStrategy() {
            super();
        }

        /**
         * Maximum keepalive time (in minutes)
         * The default value is 10 minutes. You can set it based on the number of TCP connections in
TIME_WAIT state. If there are too many TCP connections, you can increase this value.
         */
        private final long MAX_KEEP_ALIVE_MINUTES = 10;

        @Override
        public long getKeepAliveDuration(HttpResponse response, HttpContext context) {
            long keepAliveDuration = super.getKeepAliveDuration(response, context);
            // <0 indicates an unlimited keepalive period.
            // Change the period from unlimited to a default period.
            if (keepAliveDuration < 0) {
                return TimeUnit.MINUTES.toMillis(MAX_KEEP_ALIVE_MINUTES);
            }
            return keepAliveDuration;
        }
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
    }

    private static final Logger logger = LogManager.getLogger(RestLowLevelClientExample.class);

    static class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
        @Nullable
        private final CredentialsProvider credentialsProvider;
        /**
         * The {@link SSLIOSessionStrategy} for all requests to enable SSL / TLS encryption.
         */
        private final SSLIOSessionStrategy sslStrategy;
        /**
         * Create a new {@link SecuredHttpClientConfigCallback}.
         *
         * @param credentialsProvider The credential provider, if a username/password have been supplied
         * @param sslStrategy         The SSL strategy, if SSL / TLS have been supplied
         * @throws NullPointerException if {@code sslStrategy} is {@code null}
         */
        SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
            @Nullable final CredentialsProvider credentialsProvider) {
            this.sslStrategy = Objects.requireNonNull(sslStrategy);
            this.credentialsProvider = credentialsProvider;
        }
        /**
         * Get the {@link CredentialsProvider} that will be added to the HTTP client.
         *
         * @return Can be {@code null}.
         */
        @Nullable
        CredentialsProvider getCredentialsProvider() {
            return credentialsProvider;
        }
        /**
         * Get the {@link SSLIOSessionStrategy} that will be added to the HTTP client.
         *
         * @return Never {@code null}.
         */
        SSLIOSessionStrategy getSSLStrategy() {
            return sslStrategy;
        }
        /**
         * Sets the {@linkplain HttpAsyncClientBuilder#setDefaultCredentialsProvider(CredentialsProvider)
credential provider},
         *
         * @param httpClientBuilder The client to configure.
         * @return Always {@code httpClientBuilder}.
         */
        @Override
        public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder httpClientBuilder) {
            // enable SSL / TLS
            httpClientBuilder.setSSLStrategy(sslStrategy);
            // enable user authentication
            if (credentialsProvider != null) {
                httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
            }
            return httpClientBuilder;
        }
    }

    public static class NullHostNameVerifier implements HostnameVerifier {
        @Override
        public boolean verify(String arg0, SSLSession arg1) {
            return true;
        }
    }

    /**
     * The following is an example of the main function. Call the create function to create a Low Level REST
Client and check whether the test index exists.
```

```
    */
    public static void main(String[] args) throws IOException {
        RestClient lowLevelClient = create(Arrays.asList("xxx.xxx.xxx.xxx", "xxx.xxx.xxx.xxx"), 9200, "http",
30000, 30000, 30000, "username", "password");
        Request request = new Request("HEAD", "/test");
        Response response = lowLevelClient.performRequest(request);
        System.out.println(response.getStatusLine().getStatusCode());
        lowLevelClient.close();
    }
}
```

Table 3-69 Variables

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol, which can be **http** or **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.

This piece of code checks whether the **test** index exists in the cluster. If **200** (the index exists) or **404** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the Elasticsearch Low Level REST Client (With a Certificate)

Use the Elasticsearch 7.10.2 Low Level REST Client to connect to a security-mode OpenSearch cluster that uses HTTPS with a security certificate loaded, and query whether the **test** index exists. The sample code is as follows:

CAUTION

For how to obtain and upload a security certificate, see **Obtaining and Uploading a Security Certificate**.

```
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.common.Nullable;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;

import javax.net.ssl.SSLContext;import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;

public class RestLowLevelClientExample {

    private static final Logger logger = LogManager.getLogger(RestLowLevelClientExample.class);

    /**
     * Create a class for the client. Define the create function.
     */
    public static RestClient create(List<String> host, int port, String protocol, int connectTimeout, int
connectionRequestTimeout, int socketTimeout, String username, String password, String certFilePath, String
certPassword) throws IOException {
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials(username,
password));
        SSLContext sc = null;
        try {
            TrustManager[] tm = {new MyX509TrustManager(certFilePath, certPassword)};
            sc = SSLContext.getInstance("SSL", "SunJSSE");
            //You can also use SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
            sc.init(null, tm, new SecureRandom());
        } catch (Exception e) {
            e.printStackTrace();
        }

        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new NoopHostnameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
            credentialsProvider);

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientConfigCallback);
        final RestClient client = builder.build();
        logger.info("es rest client build success {} ", client);
        return client;
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
```

```
 */
public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
    return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);}

static class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
    @Nullable
    private final CredentialsProvider credentialsProvider;

    private final SSLIOSessionStrategy sslStrategy;

    SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
        @Nullable final CredentialsProvider credentialsProvider) {
        this.sslStrategy = Objects.requireNonNull(sslStrategy);
        this.credentialsProvider = credentialsProvider;
    }

    @Nullable
    CredentialsProvider getCredentialsProvider() {
        return credentialsProvider;
    }

    SSLIOSessionStrategy getSSLStrategy() {
        return sslStrategy;
    }

    @Override
    public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder httpClientBuilder) {
        httpClientBuilder.setSSLStrategy(sslStrategy);
        if (credentialsProvider != null) {
            httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
        }
        return httpClientBuilder;
    }}

public static class MyX509TrustManager implements X509TrustManager {
    X509TrustManager sunJSSEX509TrustManager;

    MyX509TrustManager(String certFilePath, String certPassword) throws Exception {
        File file = new File(certFilePath);
        if (!file.isFile()) {
            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, certPassword.toCharArray());
        TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
        tmf.init(ks);
        TrustManager[] tms = tmf.getTrustManagers();
        for (TrustManager tm : tms) {
            if (tm instanceof X509TrustManager) {
                sunJSSEX509TrustManager = (X509TrustManager) tm;
                return;
            }
        }
        throw new Exception("Couldn't initialize");
    }

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

    }

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

    }

    @Override
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        public X509Certificate[] getAcceptedIssuers() {
            return new X509Certificate[0];
        }
    }

    /**
     * The following is an example of the main function. Call the create function to create a Low Level REST Client and check whether the test index exists.
     */
    public static void main(String[] args) throws IOException {
        RestClient lowLevelClient = create(Arrays.asList("xxx.xxx.xxx.xxx", "xxx.xxx.xxx.xxx"), 9200, "https", 30000, 30000, 30000, "username", "password", "certFilePath", "certPassword");
        Request request = new Request("HEAD", "test");
        Response response = lowLevelClient.performRequest(request);
        System.out.println(response.getStatusLine().getStatusCode());
        lowLevelClient.close();
    }
}
```

Table 3-70 Function parameters

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol. Set this parameter to **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.
certFilePath	Path for storing the security certificate.
certPassword	Password of the security certificate.

This piece of code checks whether the **test** index exists in the cluster. If **200** (the index exists) or **404** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Non-Security Mode Cluster Using the Elasticsearch High Level REST Client

Use the Elasticsearch 7.10.2 High Level REST Client obtain the Low Level REST Client by calling **getLowLevelClient()**, use the low-level client to connect to an OpenSearch cluster for which the security mode is disabled, and query whether the **test** index exists. The sample code is as follows:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import org.apache.http.HttpHost;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;

import java.io.IOException;
import java.util.Arrays;
import java.util.List;
public class RestLowLevelClientExample {

    public static void main(String[] args) throws IOException {
    List<String> host = Arrays.asList("{Cluster access address}");
        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, 9200, "http"));
        final RestHighLevelClient restHighLevelClient = new RestHighLevelClient(builder);
        /**
         * Create a high-level client and then call getLowLevelClient() to obtain a low-level client. The code
differs from the client creation code only in the following line:
         */
        final RestClient lowLevelClient = restHighLevelClient.getLowLevelClient();
        /**
         * Check whether the test index exists. If the index exists, 200 is returned. If the index does not exist,
404 is returned.
         */
        Request request = new Request("HEAD", "/test");
        Response response = lowLevelClient.performRequest(request);
        System.out.println(response.getStatusLine().getStatusCode());
        lowLevelClient.close();
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }
}
```

This piece of code checks whether the **test** index exists in the cluster. If **200** (the index exists) or **404** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the Elasticsearch High Level REST Client (Without a Certificate)

Use the Elasticsearch 7.10.2 High Level REST Client to obtain the Low Level REST Client by calling **getLowLevelClient()**, use the low-level client to connect to a security-mode OpenSearch cluster that uses HTTP or HTTPS without loading a security certificate, and query whether the **test** index exists. The sample code is as follows:

```
import org.apache.http.HttpHost;
import org.apache.http.HttpResponse;
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.impl.client.DefaultConnectionKeepAliveStrategy;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.http.protocol.HttpContext;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.Response;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.common.Nullable;

import java.io.IOException;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;
import java.util.concurrent.TimeUnit;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;import javax.net.ssl.X509TrustManager;

public class RestLowLevelClientExample {

    /**
     * Create a class for the client. Define the create function.
     */
    public static RestHighLevelClient create(List<String> host, int port, String protocol, int connectTimeout,
int connectionRequestTimeout, int socketTimeout,  String username, String password) throws IOException {

        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 | NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new NullHostNameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
            credentialsProvider);

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientConfigCallback);
        final RestHighLevelClient client = new RestHighLevelClient(builder);
        logger.info("es rest client build success {} ", client);
        return client;
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }

    /**
     * Configure trustAllCerts to ignore the certificate configuration.
     */
    public static TrustManager[] trustAllCerts = new TrustManager[] {
        new X509TrustManager() {
            @Override
            public void checkClientTrusted(X509Certificate[] chain, String authType) throws CertificateException
{
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            }

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

            @Override
            public X509Certificate[] getAcceptedIssuers() {
                return null;
            }
        }
    };

    /**
     * The CustomConnectionKeepAliveStrategy function is used to set the connection keepalive when there
are a large number of short connections or when there are not many data requests.
     */
    public static class CustomConnectionKeepAliveStrategy extends DefaultConnectionKeepAliveStrategy {
        public static final CustomConnectionKeepAliveStrategy INSTANCE = new
CustomConnectionKeepAliveStrategy();

        private CustomConnectionKeepAliveStrategy() {
            super();
        }

        /**
         * Maximum keepalive time (in minutes)
         * The default value is 10 minutes. You can set it based on the number of TCP connections in
TIME_WAIT state. If there are too many TCP connections, you can increase this value.
         */
        private final long MAX_KEEP_ALIVE_MINUTES = 10;

        @Override
        public long getKeepAliveDuration(HttpResponse response, HttpContext context) {
            long keepAliveDuration = super.getKeepAliveDuration(response, context);
            // <0 indicates an unlimited keepalive period.
            // Change the period from unlimited to a default period.
            if (keepAliveDuration < 0) {
                return TimeUnit.MINUTES.toMillis(MAX_KEEP_ALIVE_MINUTES);
            }
            return keepAliveDuration;
        }
    }

    private static final Logger logger = LogManager.getLogger(RestLowLevelClientExample.class);

    static class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
        @Nullable
        private final CredentialsProvider credentialsProvider;
        /**
         * The {@link SSLIOSessionStrategy} for all requests to enable SSL / TLS encryption.
         */
        private final SSLIOSessionStrategy sslStrategy;
        /**
         * Create a new {@link SecuredHttpClientConfigCallback}.
         *
         * @param credentialsProvider The credential provider, if a username/password have been supplied
         * @param sslStrategy         The SSL strategy, if SSL / TLS have been supplied
         * @throws NullPointerException if {@code sslStrategy} is {@code null}
         */
        SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
            @Nullable final CredentialsProvider credentialsProvider) {
            this.sslStrategy = Objects.requireNonNull(sslStrategy);
            this.credentialsProvider = credentialsProvider;
        }
        /**
         * Get the {@link CredentialsProvider} that will be added to the HTTP client.
         *
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        * @return Can be {@code null}.
        */
       @Nullable
       CredentialsProvider getCredentialsProvider() {
           return credentialsProvider;
       }
       /**
        * Get the {@link SSLIOSessionStrategy} that will be added to the HTTP client.
        *
        * @return Never {@code null}.
        */
       SSLIOSessionStrategy getSSLStrategy() {
           return sslStrategy;
       }
       /**
        * Sets the {@linkplain HttpAsyncClientBuilder#setDefaultCredentialsProvider(CredentialsProvider)
credential provider},
        *
        * @param httpClientBuilder The client to configure.
        * @return Always {@code httpClientBuilder}.
        */
       @Override
       public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder httpClientBuilder) {
           // enable SSL / TLS
           httpClientBuilder.setSSLStrategy(sslStrategy);
           // enable user authentication
           if (credentialsProvider != null) {
               httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
           }
           return httpClientBuilder;
       }
   }

   public static class NullHostNameVerifier implements HostnameVerifier {
       @Override
       public boolean verify(String arg0, SSLSession arg1) {
           return true;
       }
   }

   /**
* The following is an example of the main function. Call the create function to create the high-level client,
then call the getLowLevelClient() function to obtain the low-level client, and query whether the test index
exists.
    */
   public static void main(String[] args) throws IOException {
       RestHighLevelClient client = create(Arrays.asList("xxx.xxx.xxx.xxx", "xxx.xxx.xxx.xxx"), 9200, "http",
30000, 30000, 30000, "username", "password");
       RestClient lowLevelClient = client.getLowLevelClient();
       Request request = new Request("HEAD", "test");
       Response response = lowLevelClient.performRequest(request);
       System.out.println(response.getStatusLine().getStatusCode());
       lowLevelClient.close();
   }
}
```

Table 3-71 Variables

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
protocol	Connection protocol, which can be **http** or **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.

This piece of code checks whether the **test** index exists in the cluster. If **200** (the index exists) or **404** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the Elasticsearch High Level REST Client (With a Certificate)

Use the Elasticsearch 7.10.2 High Level REST Client to obtain the Low Level REST Client by calling **getLowLevelClient()**, use the low-level client to connect to a security-mode OpenSearch cluster that uses HTTPS with a security certificate loaded, and query whether the **test** index exists. The sample code is as follows:

CAUTION

For how to obtain and upload a security certificate, see **Obtaining and Uploading a Security Certificate**.

```
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthRequest;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthResponse;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.RequestOptions;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.common.Nullable;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import java.security.KeyStore;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;

import javax.net.ssl.SSLContext;import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;

public class RestLowLevelClientExample {
    private static final Logger logger = LogManager.getLogger(RestLowLevelClientExample.class);

    /**
     * Create a class for the client. Define the create function.
     */
    public static RestHighLevelClient create(List<String> host, int port, String protocol, int connectTimeout,
int connectionRequestTimeout, int socketTimeout, String username, String password, String certFilePath,
String certPassword) throws IOException {
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials(username,
password));
        SSLContext sc = null;
        try {
            TrustManager[] tm = {new MyX509TrustManager(certFilePath, certPassword)};
            sc = SSLContext.getInstance("SSL", "SunJSSE");
            //You can also use SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
            sc.init(null, tm, new SecureRandom());
        } catch (Exception e) {
            e.printStackTrace();
        }

        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new NoopHostnameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
            credentialsProvider);

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientConfigCallback);
        final RestHighLevelClient client = new RestHighLevelClient(builder);
        logger.info("es rest client build success {} ", client);

        ClusterHealthRequest request = new ClusterHealthRequest();
        ClusterHealthResponse response = client.cluster().health(request, RequestOptions.DEFAULT);
        logger.info("es rest client health response {} ", response);
        return client;
    }

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);}

    static class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
        @Nullable
        private final CredentialsProvider credentialsProvider;

        private final SSLIOSessionStrategy sslStrategy;

        SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
            @Nullable final CredentialsProvider credentialsProvider) {
            this.sslStrategy = Objects.requireNonNull(sslStrategy);
            this.credentialsProvider = credentialsProvider;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
    }

    @Nullable
    CredentialsProvider getCredentialsProvider() {
        return credentialsProvider;
    }

    SSLIOSessionStrategy getSSLStrategy() {
        return sslStrategy;
    }

    @Override
    public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder httpClientBuilder) {
        httpClientBuilder.setSSLStrategy(sslStrategy);
        if (credentialsProvider != null) {
            httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
        }
        return httpClientBuilder;
    }}

public static class MyX509TrustManager implements X509TrustManager {
    X509TrustManager sunJSSEX509TrustManager;

    MyX509TrustManager(String certFilePath, String certPassword) throws Exception {
        File file = new File(certFilePath);
        if (!file.isFile()) {
            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, certPassword.toCharArray());
        TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
        tmf.init(ks);
        TrustManager[] tms = tmf.getTrustManagers();
        for (TrustManager tm : tms) {
            if (tm instanceof X509TrustManager) {
                sunJSSEX509TrustManager = (X509TrustManager) tm;
                return;
            }
        }
        throw new Exception("Couldn't initialize");
    }

    @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 new X509Certificate[0];
    }
}

 /**
* The following is an example of the main function. Call the create function to create the high-level client, then call the getLowLevelClient() function to obtain the low-level client, and query whether the test index exists.
 */
public static void main(String[] args) throws IOException {
    RestHighLevelClient client = create(Arrays.asList("xxx.xxx.xxx.xxx", "xxx.xxx.xxx.xxx"), 9200, "https", 30000, 30000, 30000, "username", "password", "certFilePath", "certPassword");
    RestClient lowLevelClient = client.getLowLevelClient();
```

```
        Request request = new Request("HEAD", "test");
        Response response = lowLevelClient.performRequest(request);
        System.out.println(response.getStatusLine().getStatusCode());
        lowLevelClient.close();
    }
}
```

Table 3-72 Function parameters

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol. Set this parameter to **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.
certFilePath	Path for storing the security certificate.
certPassword	Password of the security certificate.

This piece of code checks whether the **test** index exists in the cluster. If **200** (the index exists) or **404** (the index does not exist) is returned, it indicates that the cluster is connected.

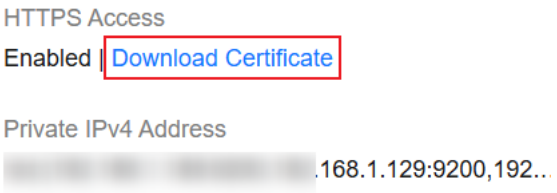
Obtaining and Uploading a Security Certificate

To access a security-mode OpenSearch cluster that uses HTTPS, a security certificate must be loaded. Perform the following steps to obtain the security certificate and upload it to the client:

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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-16 Downloading a security certificate



2. Convert the security certificate **CloudSearchService.cer**. Upload the downloaded security certificate to the client and use keytool to convert the **.cer** certificate into a **.jks** certificate that can be read by Java.
   - In Linux, run the following command to convert the certificate:
     ```
     keytool -import -alias newname -keystore ./truststore.jks -file ./CloudSearchService.cer
     ```
   - In Windows, run the following command to convert the certificate:
     ```
     keytool -import -alias newname -keystore .\truststore.jks -file .\CloudSearchService.cer
     ```

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

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

3.7.4.3 Connecting to a Cluster Through the Elasticsearch Java API Client

You can query and manage data in a CSS OpenSearch cluster using the Elasticsearch Java API Client. As the official Java client supported by Elasticsearch 8.x, the Elasticsearch Java API Client encapsulates native Elasticsearch APIs. You only need to construct the required request structure to access an OpenSearch cluster. For details about how to use the Elasticsearch Java API Client, see **Java API Client**.

Although CSS's OpenSearch clusters can be connected using the Elasticsearch Java API Client, some features of the client may still be missing if later client versions are used. To ensure compatibility, confirm that the target API has been implemented and verified in CSS.

Prerequisites

- The target OpenSearch cluster is available, the cluster version is 2.19.0 or 3.4.0, and the image version is 2.19.0_25.6.1_xxx or later.
- The server that runs the Java code can communicate with the target OpenSearch cluster.
- Depending on the network configuration method used, obtain the cluster access address. For details, see **Obtaining the cluster access address**.
- Java has been installed on the server and the JDK version is 11 or later. Download JDK 11 from **Java Archive Downloads**.

Introducing Dependencies

Import Java dependencies on the server that runs the Java code using either of the following ways:

- Maven:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Replace 8.19.0 with the actual Java client version.

```
<dependencies>
    <dependency>
        <groupId>co.elastic.clients</groupId>
        <artifactId>elasticsearch-java</artifactId>
        <version>8.19.0</version>
    </dependency>
    <dependency>
        <groupId>com.fasterxml.jackson.core</groupId>
        <artifactId>jackson-databind</artifactId>
        <version>2.17.0</version>
    </dependency>
</dependencies>
```

- Gradle:

  Replace 8.19.0 with the actual Java client version.

```
dependencies {
    implementation 'co.elastic.clients:elasticsearch-java:8.19.0'
    implementation 'com.fasterxml.jackson.core:jackson-databind:2.17.0'
}
```

Accessing a Cluster

The sample code varies depending on the security mode settings of the target OpenSearch cluster. Select the right reference document based on your service scenario.

Table 3-73 Cluster access scenarios

OpenSearch Cluster Security-Mode Settings	Whether to Load a Security Certificate	Details
Non-security mode	-	**Connecting to a Non-Security Mode Cluster Using the Elasticsearch Java API Client**
Security mode + HTTP Security mode + HTTPS	No	**Connecting to a Security-Mode Cluster Using the Elasticsearch Java API Client (Without a Certificate)**
Security mode + HTTPS	Yes	**Connecting to a Security-Mode Cluster Using the Elasticsearch Java API Client (With a Certificate)**

Connecting to a Non-Security Mode Cluster Using the Elasticsearch Java API Client

Use the Elasticsearch Java API Client to connect to an OpenSearch cluster for which the security mode is disabled, and query whether the **test** index exists. The sample code is as follows:

```
import co.elastic.clients.elasticsearch.ElasticsearchClient;
import co.elastic.clients.elasticsearch.indices.CreateIndexResponse;
import co.elastic.clients.elasticsearch.indices.DeleteIndexResponse;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import co.elastic.clients.elasticsearch.indices.GetIndexResponse;
import co.elastic.clients.json.jackson.JacksonJsonpMapper;
import co.elastic.clients.transport.ElasticsearchTransport;
import co.elastic.clients.transport.rest_client.RestClientTransport;

import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.elasticsearch.client.RestClient;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.KeyManagementException;
import java.security.KeyStore;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.concurrent.TimeUnit;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;

public class Main {

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }

    private static ElasticsearchClient create(List<String> host, int port, String protocol, int connectTimeout,
        int connectionRequestTimeout, int socketTimeout, String username, String password, String
certFilePath,
        String certPassword) throws IOException {
        // Create the low-level client
        RestClient restClient = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientBuilder -> {
                // enable user authentication
                if (username != null && password != null) {
                    final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
                    credentialsProvider.setCredentials(AuthScope.ANY,
                        new UsernamePasswordCredentials(username, password));
                    httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
                }
                // set keepalive
                httpClientBuilder.setKeepAliveStrategy(((httpResponse, httpContext) ->
TimeUnit.MINUTES.toMinutes(10)));
                // enable SSL / TLS
                SSLContext sc = (certFilePath != null && certPassword != null) ?
                    createContextFromCaCert(certFilePath, certPassword) : createTrustAllCertsContext();

                SSLIOSessionStrategy sslStrategy = new SSLIOSessionStrategy(sc, new NoopHostnameVerifier());
                httpClientBuilder.setSSLStrategy(sslStrategy);
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            httpClientBuilder.setMaxConnTotal(500);
            httpClientBuilder.setMaxConnPerRoute(300);
            return httpClientBuilder;
        })
        .build();

    // Create the transport with a Jackson mapper
    ElasticsearchTransport transport = new RestClientTransport(restClient, new JacksonJsonpMapper());

    // And create the API client
    return new ElasticsearchClient(transport);
}

private static SSLContext createTrustAllCertsContext() {
    try {
        SSLContext sslContext = SSLContext.getInstance("SSL");
        sslContext.init(null, trustAllCerts, new SecureRandom());
        return sslContext;
    } catch (NoSuchAlgorithmException | KeyManagementException e) {
        throw new RuntimeException("Can not create the SSLContext", e);
    }
}

private static SSLContext createContextFromCaCert(String certFilePath, String certPassword) {
    try {
        // enable SSL / TLS
        TrustManager[] tm = {new MyX509TrustManager(certFilePath, certPassword)};
        SSLContext sc = SSLContext.getInstance("SSL", "SunJSSE");
        //You can also use SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
        sc.init(null, tm, new SecureRandom());
        return sc;
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
}

/**
 * Configure trustAllCerts to ignore the certificate configuration.
 */
private 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;
        }
    }
};

/**
 * Configure certificate authentication
 */
private static class MyX509TrustManager implements X509TrustManager {
    X509TrustManager sunJSSEX509TrustManager;

    MyX509TrustManager(String certFilePath, String certPassword) throws Exception {
        File file = new File(certFilePath);
        if (!file.isFile()) {
            throw new Exception("Wrong Certification Path");
        }
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            System.out.println("Loading KeyStore " + file + "...");
            InputStream in = new FileInputStream(file);
            KeyStore ks = KeyStore.getInstance("JKS");
            ks.load(in, certPassword.toCharArray());
            TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
            tmf.init(ks);
            TrustManager[] tms = tmf.getTrustManagers();
            for (TrustManager tm : tms) {
                if (tm instanceof X509TrustManager) {
                    sunJSSEX509TrustManager = (X509TrustManager) tm;
                    return;
                }
            }
            throw new Exception("Couldn't initialize");
        }

        @Override
        public void checkClientTrusted(X509Certificate[] chain, String authType) {

        }

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

        }

        @Override
        public X509Certificate[] getAcceptedIssuers() {
            return new X509Certificate[0];
        }
    }

    /**
* The following is an example of the main function. Call the create function to create a client and check whether the test index exists.
     */
    public static void main(String[] args) throws IOException {
        ElasticsearchClient esClient = create(Arrays.asList("{IP address of the target cluster}"), 9200, "http", 1000, 1000, 1000, null, null, null, null);

        CreateIndexResponse indexRequest = esClient.indices()
            .create(createIndexBuilder -> createIndexBuilder.index("test"));
        boolean acknowledged = indexRequest.acknowledged();
        System.out.println("Create index successfully! " + acknowledged);

        GetIndexResponse getIndexResponse = esClient.indices().get(getIndexRequest -> getIndexRequest.index("test"));
        System.out.println("Query index successfully! \n" + getIndexResponse.toString());

        DeleteIndexResponse deleteResponse = esClient.indices()
            .delete(createIndexBuilder -> createIndexBuilder.index("test"));
        System.out.println("Delete index successfully! \n" + deleteResponse.toString());
    }
}
```

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the Elasticsearch Java API Client (Without a Certificate)

Use the Elasticsearch Java API Client to connect to a security-mode OpenSearch cluster (HTTP or HTTPS) without loading a security certificate, and query whether the **test** index exists. The sample code is as follows:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import co.elastic.clients.elasticsearch.ElasticsearchClient;
import co.elastic.clients.elasticsearch.indices.CreateIndexResponse;
import co.elastic.clients.elasticsearch.indices.DeleteIndexResponse;
import co.elastic.clients.elasticsearch.indices.GetIndexResponse;
import co.elastic.clients.json.jackson.JacksonJsonpMapper;
import co.elastic.clients.transport.ElasticsearchTransport;
import co.elastic.clients.transport.rest_client.RestClientTransport;

import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.elasticsearch.client.RestClient;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.KeyManagementException;
import java.security.KeyStore;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.concurrent.TimeUnit;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;

public class Main {

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }

    private static ElasticsearchClient create(List<String> host, int port, String protocol, int connectTimeout,
        int connectionRequestTimeout, int socketTimeout, String username, String password, String
certFilePath,
        String certPassword) throws IOException {
        // Create the low-level client
        RestClient restClient = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientBuilder -> {
                // enable user authentication
                if (username != null && password != null) {
                    final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
                    credentialsProvider.setCredentials(AuthScope.ANY,
                        new UsernamePasswordCredentials(username, password));
                    httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
                }
                // set keepalive
                httpClientBuilder.setKeepAliveStrategy(((httpResponse, httpContext) ->
TimeUnit.MINUTES.toMinutes(10)));
                // enable SSL / TLS
                SSLContext sc = (certFilePath != null && certPassword != null) ?
                    createContextFromCaCert(certFilePath, certPassword) : createTrustAllCertsContext();
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            SSLIOSessionStrategy sslStrategy = new SSLIOSessionStrategy(sc, new NoopHostnameVerifier());
            httpClientBuilder.setSSLStrategy(sslStrategy);

            httpClientBuilder.setMaxConnTotal(500);
            httpClientBuilder.setMaxConnPerRoute(300);
            return httpClientBuilder;
        })
        .build();

    // Create the transport with a Jackson mapper
    ElasticsearchTransport transport = new RestClientTransport(restClient, new JacksonJsonpMapper());

    // And create the API client
    return new ElasticsearchClient(transport);
}

private static SSLContext createTrustAllCertsContext() {
    try {
        SSLContext sslContext = SSLContext.getInstance("SSL");
        sslContext.init(null, trustAllCerts, new SecureRandom());
        return sslContext;
    } catch (NoSuchAlgorithmException | KeyManagementException e) {
        throw new RuntimeException("Can not create the SSLContext", e);
    }
}

private static SSLContext createContextFromCaCert(String certFilePath, String certPassword) {
    try {
        // enable SSL / TLS
        TrustManager[] tm = {new MyX509TrustManager(certFilePath, certPassword)};
        SSLContext sc = SSLContext.getInstance("SSL", "SunJSSE");
        //You can also use SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
        sc.init(null, tm, new SecureRandom());
        return sc;
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
}

/**
 * Configure trustAllCerts to ignore the certificate configuration.
 */
private 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;
        }
    }
};

/**
 * Configure certificate authentication
 */
private static class MyX509TrustManager implements X509TrustManager {
    X509TrustManager sunJSSEX509TrustManager;

    MyX509TrustManager(String certFilePath, String certPassword) throws Exception {
        File file = new File(certFilePath);
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            if (!file.isFile()) {
                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, certPassword.toCharArray());
            TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
            tmf.init(ks);
            TrustManager[] tms = tmf.getTrustManagers();
            for (TrustManager tm : tms) {
                if (tm instanceof X509TrustManager) {
                    sunJSSEX509TrustManager = (X509TrustManager) tm;
                    return;
                }
            }
            throw new Exception("Couldn't initialize");
        }

        @Override
        public void checkClientTrusted(X509Certificate[] chain, String authType) {

        }

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

        }

        @Override
        public X509Certificate[] getAcceptedIssuers() {
            return new X509Certificate[0];
        }
    }

    /**
* The following is an example of the main function. Call the create function to create a client and check whether the test index exists.
     */
    public static void main(String[] args) throws IOException {
        ElasticsearchClient esClient = create(Arrays.asList("{IP address of the target cluster}"), 9200, "https", 1000, 1000, 1000, "{Username}", "{Password}", null, null);

        CreateIndexResponse indexRequest = esClient.indices()
            .create(createIndexBuilder -> createIndexBuilder.index("test"));
        boolean acknowledged = indexRequest.acknowledged();
        System.out.println("Create index successfully! " + acknowledged);

        GetIndexResponse getIndexResponse = esClient.indices().get(getIndexRequest -> getIndexRequest.index("test"));
        System.out.println("Query index successfully! \n" + getIndexResponse.toString());

        DeleteIndexResponse deleteResponse = esClient.indices()
            .delete(createIndexBuilder -> createIndexBuilder.index("test"));
        System.out.println("Delete index successfully! \n" + deleteResponse.toString());
    }
}
```

Table 3-74 Variables

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol, which can be **http** or **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode Cluster Using the Elasticsearch Java API Client (With a Certificate)

Use the Elasticsearch Java API Client to connect to a security-mode OpenSearch cluster that uses HTTPS with a security certificate loaded, and query whether the **test** index exists. The sample code is as follows:

CAUTION

For how to obtain and upload a security certificate, see **Obtaining and Uploading a Security Certificate**.

```
import co.elastic.clients.elasticsearch.ElasticsearchClient;
import co.elastic.clients.elasticsearch.indices.CreateIndexResponse;
import co.elastic.clients.elasticsearch.indices.DeleteIndexResponse;
import co.elastic.clients.elasticsearch.indices.GetIndexResponse;
import co.elastic.clients.json.jackson.JacksonJsonpMapper;
import co.elastic.clients.transport.ElasticsearchTransport;
import co.elastic.clients.transport.rest_client.RestClientTransport;

import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.elasticsearch.client.RestClient;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import java.security.KeyManagementException;
import java.security.KeyStore;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.concurrent.TimeUnit;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;

public class Main {

    /**
     * Use the constructHttpHosts function to convert the node IP address list of the host cluster.
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }

    private static ElasticsearchClient create(List<String> host, int port, String protocol, int connectTimeout,
        int connectionRequestTimeout, int socketTimeout, String username, String password, String
certFilePath,
        String certPassword) throws IOException {
        // Create the low-level client
        RestClient restClient = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientBuilder -> {
                // enable user authentication
                if (username != null && password != null) {
                    final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
                    credentialsProvider.setCredentials(AuthScope.ANY,
                        new UsernamePasswordCredentials(username, password));
                    httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
                }
                // set keepalive
                httpClientBuilder.setKeepAliveStrategy(((httpResponse, httpContext) ->
TimeUnit.MINUTES.toMinutes(10)));
                // enable SSL / TLS
                SSLContext sc = (certFilePath != null && certPassword != null) ?
                    createContextFromCaCert(certFilePath, certPassword) : createTrustAllCertsContext();

                SSLIOSessionStrategy sslStrategy = new SSLIOSessionStrategy(sc, new NoopHostnameVerifier());
                httpClientBuilder.setSSLStrategy(sslStrategy);

                httpClientBuilder.setMaxConnTotal(500);
                httpClientBuilder.setMaxConnPerRoute(300);
                return httpClientBuilder;
            })
            .build();

        // Create the transport with a Jackson mapper
        ElasticsearchTransport transport = new RestClientTransport(restClient, new JacksonJsonpMapper());

        // And create the API client
        return new ElasticsearchClient(transport);
    }

    private static SSLContext createTrustAllCertsContext() {
        try {
            SSLContext sslContext = SSLContext.getInstance("SSL");
            sslContext.init(null, trustAllCerts, new SecureRandom());
            return sslContext;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        } catch (NoSuchAlgorithmException | KeyManagementException e) {
            throw new RuntimeException("Can not create the SSLContext", e);
        }
    }

    private static SSLContext createContextFromCaCert(String certFilePath, String certPassword) {
        try {
            // enable SSL / TLS
            TrustManager[] tm = {new MyX509TrustManager(certFilePath, certPassword)};
            SSLContext sc = SSLContext.getInstance("SSL", "SunJSSE");
            //You can also use SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
            sc.init(null, tm, new SecureRandom());
            return sc;
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }

    /**
     * Configure trustAllCerts to ignore the certificate configuration.
     */
    private 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;
            }
        }
    };

    /**
     * Configure certificate authentication
     */
    private static class MyX509TrustManager implements X509TrustManager {
        X509TrustManager sunJSSEX509TrustManager;

        MyX509TrustManager(String certFilePath, String certPassword) throws Exception {
            File file = new File(certFilePath);
            if (!file.isFile()) {
                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, certPassword.toCharArray());
            TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
            tmf.init(ks);
            TrustManager[] tms = tmf.getTrustManagers();
            for (TrustManager tm : tms) {
                if (tm instanceof X509TrustManager) {
                    sunJSSEX509TrustManager = (X509TrustManager) tm;
                    return;
                }
            }
            throw new Exception("Couldn't initialize");
        }

        @Override
        public void checkClientTrusted(X509Certificate[] chain, String authType) {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        }

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

        }

        @Override
        public X509Certificate[] getAcceptedIssuers() {
            return new X509Certificate[0];
        }
    }

    /**
* The following is an example of the main function. Call the create function to create a client and check whether the test index exists.
     */
    public static void main(String[] args) throws IOException {
        ElasticsearchClient esClient = create(Arrays.asList("{IP address of the target cluster}"), 9200, "https", 1000, 1000, 1000, "{Username}", "{Password}", "{Certificate path}", "{Certificate password}")

        CreateIndexResponse indexRequest = esClient.indices()
            .create(createIndexBuilder -> createIndexBuilder.index("test"));
        boolean acknowledged = indexRequest.acknowledged();
        System.out.println("Create index successfully! " + acknowledged);

        GetIndexResponse getIndexResponse = esClient.indices().get(getIndexRequest -> getIndexRequest.index("test"));
        System.out.println("Query index successfully! \n" + getIndexResponse.toString());

        DeleteIndexResponse deleteResponse = esClient.indices()
            .delete(createIndexBuilder -> createIndexBuilder.index("test"));
        System.out.println("Delete index successfully! \n" + deleteResponse.toString());
    }
}
```

Table 3-75 Function parameters

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them using a comma (,).
port	Access port of the cluster. The default value is **9200**.
protocol	Connection protocol. Set this parameter to **https**.
connectTimeout	Connection timeout (in ms).
connectionRequestTimeout	Socket connection request timeout (in ms).
socketTimeout	Socket request timeout (in ms).
username	Username for accessing the cluster.
password	Password of the user.
certFilePath	Path for storing the security certificate.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
certPassword	Password of the security certificate.

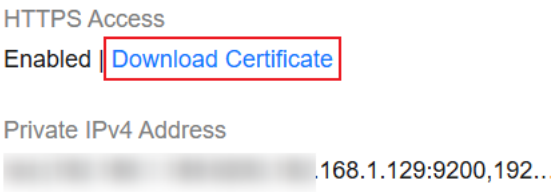
This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

Obtaining and Uploading a Security Certificate

To access a security-mode OpenSearch cluster that uses HTTPS, a security certificate must be loaded. Perform the following steps to obtain the security certificate and upload it to the client:

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

      Figure 3-17 Downloading a security certificate

2. Convert the security certificate **CloudSearchService.cer**. Upload the downloaded security certificate to the client and use keytool to convert the **.cer** certificate into a **.jks** certificate that can be read by Java.
   - In Linux, run the following command to convert the certificate:
     ```
     keytool -import -alias newname -keystore ./truststore.jks -file ./CloudSearchService.cer
     ```
   - In Windows, run the following command to convert the certificate:
     ```
     keytool -import -alias newname -keystore .\truststore.jks -file .\CloudSearchService.cer
     ```

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

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

FAQ: How Do I Increase the Number of Client Connections to Handle High Concurrency?

When using the Elasticsearch Java API Client, the default maximum number of client connections may become insufficient under high concurrency, leading to request delays or failures.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

To address this, you can increase the connection limit by customizing HttpClientConfigCallback during client initialization. This involves injecting an HttpClientBuilder to increase the maximum number of client connections supported, thereby improving the cluster's ability to handle concurrent requests.

Configuration example:

```
// Set the maximum number of total connections to 500. (Set this parameter based on the typical concurrency the cluster handles.)
httpClientBuilder.setMaxConnTotal(500);
//Set the maximum number of connections per route to 300. You are advised to set this value to approximately 60% of the total client connection limit.
httpClientBuilder.setMaxConnPerRoute(300);
```

3.7.5 Connecting to a Cluster Using Python

You can access and manage data in a CSS OpenSearch cluster using the official OpenSearch Python client (opensearch-py) or Elasticsearch Python client (elasticsearch-py). Both client libraries are compatible with OpenSearch REST APIs, though version compatibility needs to be verified for elasticsearch-py. Developers can use Python APIs to manage indexes; create, read, update, and delete (CRUD) documents; and search data.

You are advised to use the OpenSearch Python client (opensearch-py) to access complete OpenSearch features. For further information, see **OpenSearch Python Client Documentation**.

Prerequisites

- The target OpenSearch cluster is available.
- The server that runs the Python code can communicate with the OpenSearch cluster.
- Depending on the network configuration method used, obtain the cluster access address. For details, see **Obtaining the cluster access address**.
- Python 3 has been installed on the server.

Installing the Python Client Library

Install the Python client library on the server that runs the Python code.

CSS allows you to use the Elasticsearch 7.10 Python client to connect to OpenSearch clusters. To ensure better compatibility, however, you are advised to use a Python client library that has the same version as the target OpenSearch cluster.

Select a reference example based on the Python client you use.

- Scenario 1 (recommended): Use the OpenSearch Python client to access an OpenSearch cluster and install the OpenSearch Python client library.
  ```
  pip install opensearch-py
  ```
- Scenario 2: Use the Elasticsearch 7.10 Python client to access an OpenSearch cluster and install the Elasticsearch 7.10 Python client library.
  ```
  pip install Elasticsearch==7.10
  ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Cluster Access Scenarios

The sample code varies depending on the Python client and the security mode settings of the target OpenSearch cluster. Select the right reference document based on your service scenario.

Table 3-76 Cluster access scenarios

Python Client Type	OpenSearch Cluster Security-Mode Settings	Reference Document
OpenSearch	Non-security mode	**Connecting to a Non-Security Mode Cluster with the OpenSearch Python Client**
	Security mode + HTTP	**Connecting to a Security-Mode +HTTP Cluster with the OpenSearch Python Client**
	Security mode + HTTPS	**Connecting to a Security-Mode +HTTPS Cluster with the OpenSearch Python Client**
Elasticsearch 7.10.2	Non-security mode	**Connecting to a Non-Security Mode Cluster with the Elasticsearch Python Client**
	Security mode + HTTP	**Connecting to a Security-Mode +HTTP Cluster with the Elasticsearch Python Client**
	Security mode + HTTPS	**Connecting to a Security-Mode +HTTPS Cluster with the Elasticsearch Python Client**

Connecting to a Non-Security Mode Cluster with the OpenSearch Python Client

Use the OpenSearch Python client to connect to an OpenSearch cluster for which the security mode is disabled, and query whether the **test** index exists.

The following is an example of the code for creating a client instance using the cluster connection information:

```
from opensearchpy import OpenSearch

class OpenSearchFactory(object):

    def __init__(self, host: list, port: str, username: str, password: str):
        self.port = port
        self.host = host
        self.username = username
        self.password = password

    def create(self) -> OpenSearch:
        addrs = []
        for host in self.host:
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            addr = {'host': host, 'port': self.port}
            addrs.append(addr)

        if self.username and self.password:
            opensearch = OpenSearch(addrs, http_auth=(self.username, self.password))
        else:
            opensearch = OpenSearch(addrs)
        return opensearch
os = OpenSearchFactory(["{cluster address}"], "9200", None, None).create()
print(os.indices.exists(index='test'))
```

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode+HTTP Cluster with the OpenSearch Python Client

Use the OpenSearch Python client to connect to a security-mode OpenSearch cluster that uses HTTP, and query whether the **test** index exists.

The following is an example of the code for creating a client instance using the cluster connection information:

```
from opensearchpy import OpenSearch

class OpenSearchFactory(object):

    def __init__(self, host: list, port: str, username: str, password: str):
        self.port = port
        self.host = host
        self.username = username
        self.password = password

    def create(self) -> OpenSearch:
        addrs = []
        for host in self.host:
            addr = {'host': host, 'port': self.port}
            addrs.append(addr)

        if self.username and self.password:
            opensearch = OpenSearch(addrs, http_auth=(self.username, self.password))
        else:
            opensearch = OpenSearch(addrs)
        return opensearch
os = OpenSearchFactory(["xxx.xxx.xxx.xxx"], "9200", "username", "password").create()
print(os.indices.exists(index='test'))
```

Table 3-77 Variables

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them with commas (,).
port	Access port of the cluster. Enter **9200**.
username	Username for accessing the cluster.
password	Password of the user.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode+HTTPS Cluster with the OpenSearch Python Client

Use the OpenSearch Python client to connect to a security-mode OpenSearch cluster that uses HTTPS, and query whether the **test** index exists.

The following is an example of the code for creating a client instance using the cluster connection information:

```
from opensearchpy import OpenSearch
import ssl

class OpenSearchFactory(object):

    def __init__(self, host: list, port: str, username: str, password: str):
        self.port = port
        self.host = host
        self.username = username
        self.password = password

    def create(self) -> OpenSearch:
        context = ssl._create_unverified_context()

        addrs = []
        for host in self.host:
            addr = {'host': host, 'port': self.port}
            addrs.append(addr)

        if self.username and self.password:
            opensearch = OpenSearch(addrs, http_auth=(self.username, self.password), scheme="https",
ssl_context=context)
        else:
            opensearch = OpenSearch(addrs)
        return opensearch
os = OpenSearchFactory(["xxx.xxx.xxx.xxx"], "9200", "username", "password").create()
print(os.indices.exists(index='test'))
```

Table 3-78 Variables

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them with commas (,).
port	Access port of the cluster. Enter **9200**.
username	Username for accessing the cluster.
password	Password of the user.

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Connecting to a Non-Security Mode Cluster with the Elasticsearch Python Client

Use the Elasticsearch 7.10.2 Python client to connect to an OpenSearch cluster for which the security mode is disabled, and query whether the **test** index exists.

The following is an example of the code for creating a client instance using the cluster connection information:

```
from elasticsearch import Elasticsearch

class ElasticFactory(object):

    def __init__(self, host: list, port: str, username: str, password: str):
        self.port = port
        self.host = host
        self.username = username
        self.password = password

    def create(self) -> Elasticsearch:
        addrs = []
        for host in self.host:
            addr = {'host': host, 'port': self.port}
            addrs.append(addr)

        if self.username and self.password:
            elasticsearch = Elasticsearch(addrs, http_auth=(self.username, self.password))
        else:
            elasticsearch = Elasticsearch(addrs)
        return elasticsearch

es = ElasticFactory(["{cluster address}"], "9200", None, None).create()
print(es.indices.exists(index='test'))
```

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode+HTTP Cluster with the Elasticsearch Python Client

Use the Elasticsearch 7.10.2 Python client to connect to a security-mode OpenSearch cluster that uses HTTP, and query whether the **test** index exists.

The following is an example of the code for creating a client instance using the cluster connection information:

```
from elasticsearch import Elasticsearch

class ElasticFactory(object):

    def __init__(self, host: list, port: str, username: str, password: str):
        self.port = port
        self.host = host
        self.username = username
        self.password = password

    def create(self) -> Elasticsearch:
        addrs = []
        for host in self.host:
            addr = {'host': host, 'port': self.port}
            addrs.append(addr)

        if self.username and self.password:
            elasticsearch = Elasticsearch(addrs, http_auth=(self.username, self.password))
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        else:
            elasticsearch = Elasticsearch(addrs)
        return elasticsearch

es = ElasticFactory(["xxx.xxx.xxx.xxx"], "9200", "username", "password").create()
print(es.indices.exists(index='test'))
```

Table 3-79 Variables

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them with commas (,).
port	Access port of the cluster. Enter **9200**.
username	Username for accessing the cluster.
password	Password of the user.

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

Connecting to a Security-Mode+HTTPS Cluster with the Elasticsearch Python Client

Use the Elasticsearch 7.10.2 Python client to connect to a security-mode OpenSearch cluster that uses HTTPS, and query whether the **test** index exists.

The following is an example of the code for creating a client instance using the cluster connection information:

```
from elasticsearch import Elasticsearch
import ssl

class ElasticFactory(object):

    def __init__(self, host: list, port: str, username: str, password: str):
        self.port = port
        self.host = host
        self.username = username
        self.password = password

    def create(self) -> Elasticsearch:
        context = ssl._create_unverified_context()

        addrs = []
        for host in self.host:
            addr = {'host': host, 'port': self.port}
            addrs.append(addr)

        if self.username and self.password:
            elasticsearch = Elasticsearch(addrs, http_auth=(self.username, self.password), scheme="https",
ssl_context=context)
        else:
            elasticsearch = Elasticsearch(addrs)
        return elasticsearch
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
es = ElasticFactory(["xxx.xxx.xxx.xxx"], "9200", "username", "password").create()
print(es.indices.exists(index='test'))
```

Table 3-80 Variables

Parameter	Description
host	IP address for accessing the cluster. If there are multiple IP addresses, separate them with commas (,).
port	Access port of the cluster. Enter **9200**.
username	Username for accessing the cluster.
password	Password of the user.

This piece of code checks whether the **test** index exists in the cluster. If **true** (the index exists) or **false** (the index does not exist) is returned, it indicates that the cluster is connected.

3.7.6 Connecting to a Cluster Using cURL Commands

To query or analyze data in OpenSearch, you often need to connect to your cluster from an external client (such as an ECS). This topic describes how to use cURL commands to access an OpenSearch cluster.

Prerequisites

- The target CSS OpenSearch cluster is available.
- An ECS has been created, and it is in the same VPC and security group as the CSS cluster.
  - If they are not in the same security group, modify the ECS security group, or configure its inbound and outbound rules to allow all access from the cluster. For details, see .
  - For details about how to use an ECS, see .

Accessing a Cluster

1. Obtain the private network address of the cluster. It is used to access the cluster.

   This topic uses a private IP address as an example to describe how to access a cluster. The cluster access address varies depending on the network configurations used. For details, see **Obtaining the Cluster Access Address**.

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

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

   c. In the cluster list, obtain the target cluster's internal network address from the **Internal Network Address** column. Typical address format: *<host>*:*<port>* or *<host>*:*<port>*,*<host>*:*<port>*. Example: **10.62.179.32:9200,10.62.179.33:9200**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

If the cluster has client nodes, only the IP addresses and ports of all the client nodes are displayed. Otherwise, the IP addresses and ports of all data nodes and cold data nodes are displayed.

2. Run one of the following commands on the ECS to access the cluster. The access command varies depending on the security mode of the cluster.
   - For a cluster with the security mode disabled:
     ```
     curl "http://<host>:<port>"
     ```
   - For a security-mode cluster that uses HTTP:
     ```
     curl -u <user>:<password> "http://<host>:<port>"
     ```
   - For a security-mode cluster that uses HTTPS:
     ```
     curl -u <user>:<password> -k "https://<host>:<port>"
     ```

Table 3-81 Variables

Variable	Description
host	IP address of each node in the cluster. If the cluster contains multiple nodes, there will be multiple IP addresses. You can use any of them.
port	Port number for accessing a cluster node. Generally, the port number is 9200.
user	Username for accessing the cluster.
password	Password of the user. If the password contains special characters, enclose the username and password in single quotation marks, for example, **curl -u "user:password!" "http://<host>:<port>"**.

Example command:

```
curl "http://10.62.176.32:9200"
```

Example response:

```
HTTP/1.1 200 OK
content-type: application/json; charset=UTF-8
content-length: 513

{
  "name" : "css-xxx-ess-esn-3-1",
  "cluster_name" : "css-xxx",
  "cluster_uuid" : "xxx_uuid",
  "version" : {
    "number" : "7.10.2",
    "build_type" : "tar",
    "build_hash" : "unknown",
    "build_date" : "unknown",
    "build_snapshot" : true,
    "lucene_version" : "9.12.1",
    "minimum_wire_compatibility_version" : "7.10.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "The OpenSearch Project: https://opensearch.org/"
}
```

For more commands, see the **OpenSearch documentation**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3.8 Ingesting Data

3.8.1 Data Ingestion Methods

When building an enterprise-grade search and analytics platform, you may need to ingest large volumes of heterogeneous data distributed across relational databases (such as MySQL and Oracle), messaging systems (such as Kafka), object storage services (such as OBS), and business applications into an OpenSearch cluster. To meet varying requirements for data latency, data volume, and development costs, CSS OpenSearch clusters provide multiple data ingestion methods. This topic briefly introduces and compares these methods, including their applicable scenarios and characteristics, to help you select an optimal option.

NOTE

Before ingesting large amounts of data, you can enhance the destination OpenSearch cluster's ingestion performance to improve throughput. For details, see **Enhancing Data Ingestion Performance**.

Table 3-82 Comparing different data ingestion methods for OpenSearch clusters

Method	When to Use	Supported Source/Format	Details
CSS Logstash	You want to perform complex data cleaning (filtering) without building and managing your own Logstash server. For example, you may need to clean Nginx logs from Kafka before ingesting them into OpenSearch.	MySQL, Kafka, and OBS	**Using Logstash to Synchronize Data to Elasticsearch**
Cloud Data Migration (CDM)	You need to perform a full migration of historical data. This method is code-free and guided by a step-by-step wizard. For example, you may ingest years of archived logs from OBS or historical orders from Oracle into OpenSearch.	OBS (JSON/CSV), Oracle, and MySQL	**Migrating Data Using CDM**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Method	When to Use	Supported Source/Format	Details
Open-source Logstash	You want to migrate data from an on-premises IDC to OpenSearch clusters on the cloud, during which you may use special plugins or deeply customized pipeline logic. For example, you may upload system logs from an on-premises data center to OpenSearch via an SSH tunnel.	Any source supported by Logstash input, such as JSON, CSV, and text	**Ingesting Data Using Self-Managed Logstash**
Open-source APIs	You need to quickly write small amounts of data to OpenSearch during development or debugging. For example, you can have Java/Python applications directly call OpenSearch APIs to write data.	JSON	**Ingesting Data Using Open-Source OpenSearch APIs**

3.8.2 Migrating Data Using CDM

Enterprises often operate heterogeneous platforms, with data distributed across traditional relational databases such as Oracle and storage systems like Object Storage Service (OBS). To enable high-performance full-text search and analytics, this data often needs to be migrated into OpenSearch. However, migrating large volumes of heterogeneous data using custom scripts is costly and often hindered by complex network configurations and low transfer efficiency. The Cloud Data Migration (CDM) service simplifies this process with a no-code, managed service that supports high concurrency and resumable transfers. Using CDM, you can quickly and securely migrate data from Oracle databases or OBS to CSS OpenSearch clusters.

Table 3-83 Ingesting data into CSS using CDM

Scenario	Data Source	Destination Cluster
Ingesting data from an Oracle database into CSS	A local or third-party Oracle database	OpenSearch 1.3.6

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Scenario	Data Source	Destination Cluster
Ingesting data from OBS to CSS	JSON/CSV files in OBS buckets	OpenSearch 1.3.6

Preparations

1. Connect the network.
   - Ensure the CDM cluster, CSS cluster, and OBS bucket reside in the same VPC. This enables data transmission over the internal network for optimal speed.
   - If the source is an on-premises Oracle database, make sure CDM can access it through a VPN, Direct Connect, or public IP address.
2. Obtain connection information
   - Obtain the private network address (for example, **192.168.xxx.xxx:9200**), username, and password of the CSS cluster. (The username and password are required only for a security-mode cluster.)
   - If the source is an Oracle database, you need to obtain the database's IP address, port, database name, username, and password.
   - If the source is OBS, obtain the OBS bucket name, domain name, port, AK, and SK.

Migrating Data

1. Log in to OpenSearch Dashboards and go to the command execution page.
   a. Log in to the **CSS management console**.
   b. In the navigation pane on the left, choose **Clusters > OpenSearch**.
   c. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
   d. In the left navigation pane, choose **Dev Tools**.

      The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.
2. (Optional) Create the destination index in the OpenSearch cluster. CDM can automatically create destination indexes. However, to ensure optimal query performance, you are advised to define the index mappings in the destination OpenSearch cluster in advance.

   For example, run the following command to create index **demo**:

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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        "type": "keyword"
      }
    }
  }
}
```

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

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

3. Ingest data from Oracle or OBS to the OpenSearch cluster using CDM.
   - If the data source is an Oracle database, see **Migrating Data from Oracle to CSS** for an operation guide.
   - If the data source is OBS, see **Migrating Data from OBS to CSS** for an operation guide.

   NOTE

   If the data source is MySQL, see **Creating a MySQL Connector** for how to connect to a MySQL database.
4. After the migration is complete, verify data integrity.
   a. Log in to the OpenSearch cluster via OpenSearch Dashboards, and navigate to the **Dev Tools** page.
   b. Run the following command to check the newly ingested data:

      ```
      GET demo/_count      # Check the number of records ingested.
      GET demo/_search     # Check the content of the ingested data.
      ```

      If the results are consistent with the source, data ingestion is successful.

3.8.3 Ingesting Data Using Self-Managed Logstash

Enterprises typically generate large volumes of service logs (in JSON or CSV formats), often distributed across local servers or third-party cloud platforms. For centralized search and real-time analytics, you may choose to aggregate such data into CSS OpenSearch clusters. However, for security reasons, CSS clusters are typically deployed within private networks (for example, VPCs), which restrict direct ingestion from external data sources. This topic describes how to use open-source Logstash to securely and efficiently ingest data into CSS clusters in both private and public network environments.

Solution

Logstash is an open-source, server-side data processing pipeline that ingests data from multiple sources simultaneously, processes and transforms the data, and then sends it to CSS clusters. For more information about Logstash, see the official document **Getting started with Logstash**.

Depending on where the Logstash server is deployed, this solution can use either of the following network architectures:

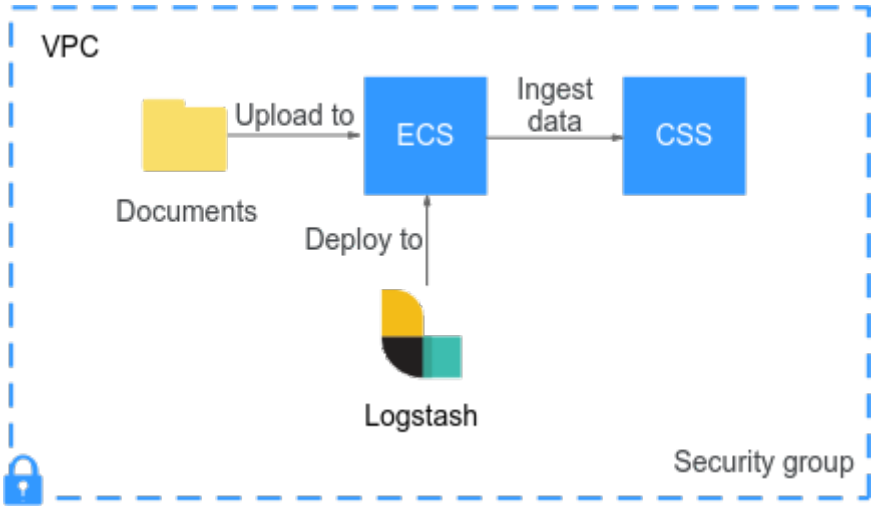
- Network architecture 1: Direct connection within a VPC (recommended)

  Logstash is deployed on an ECS that is in the same VPC as the destination CSS cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Characteristics: fast, easy to set up, and secure.

Figure 3-18 Logstash and CSS within the same VPC

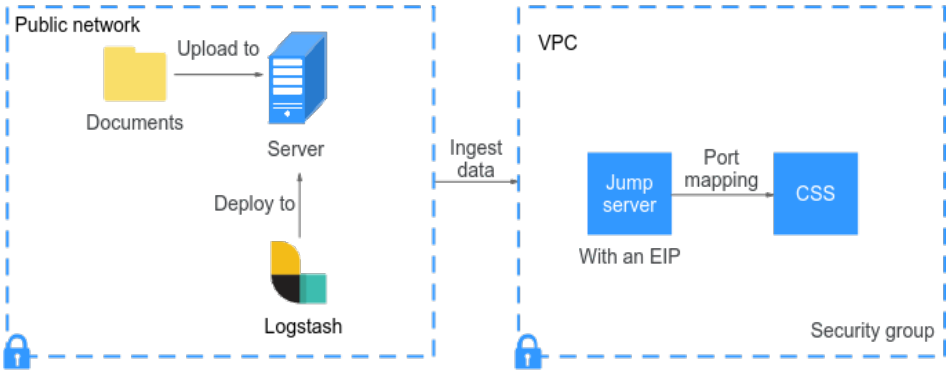


- Network architecture 2: SSH connection over the public network

  Logstash is deployed in an on-premises data center or on a server located on the public network, which prevents it from directly accessing the destination CSS cluster in a private network. To connect the two, we need to use an ECS allocated with a public IP address as a jump server. This jump server leverages SSH port forwarding to forward traffic coming from Logstash to the destination cluster on the cloud.

  Characteristics: suitable for migrating on-premises data to the cloud; port forwarding must be configured.

Figure 3-19 Logstash deployed on premises or in the public network



Prerequisites

- The server that deploys Logstash has JDK 1.8 or later installed. A Linux server is recommended.
- You have downloaded the OSS version (pure open-source) of Logstash. Logstash OSS 7.10.2 is recommended.

  Download Logstash OSS at **https://www.elastic.co/downloads/past-releases?product=logstash-oss**.
- You have obtained the connection information for both the data source and destination, including IP addresses, port numbers, accounts, and passwords.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- The source data is ready for ingestion.

  For example, the **access_20181029.log** is already in the **/tmp/access_log/** directory on the Logstash server. The following is an example of the 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 |
```

Scenario 1: Ingesting Data When Logstash and CSS Reside in the Same VPC

1. Prepare the network: Make sure the ECS that hosts Logstash and the destination CSS cluster are within the same VPC, and that the VPC's security group allows port 9200.

   Ping the CSS cluster's private network address from the ECS where you plan to deploy Logstash. If successful, the two are connected.

2. Install Logstash on the ECS. The following provides an installation guide for Red Hat Linux. For other operating systems, see **Installing Logstash**.

   a. Use yum to install the JDK first. This is because Logstash depends on Java.

   ```
   yum install java  # Replace it with the actual JDK version.
   ```

   b. Upload the downloaded Logstash package to the ECS.

   c. Use yum to install Logstash.

   ```
   yum install logstash-oss-7.10.0-x86_64.rpm
   ```

   Replace *logstash-oss-7.10.0-x86_64.rpm* with the actual Logstash installation package name.

   a. Modify the JVM configuration of Logstash to accelerate cluster data migration.

   The default heap memory of Logstash is 1 GB. You are advised to change it to half the cluster node memory.

   ```
   vim /etc/logstash/jvm.options
   -Xms4g
   -Xmx4g
   ```

3. (Optional) Prepare the security certificate. If the security mode and HTTPS are enabled for the destination CSS cluster, you must upload the cluster's security certificate to the ECS.

   a. **Obtaining the Security Certificate**.

   b. Upload the downloaded security certificate to the ECS.

4. Create a configuration file in the Logstash installation directory.

   a. Create a configuration file, for example, **logstash-simple.conf**.

   ```
   cd /logstash-x.x/ # Go to the Logstash installation directory.
   vi logstash-simple.conf     # Create a configuration file.
   ```

   b. Define input, filter, and output in the configuration file.

   ```
   input {
      # Read local log files.
      file {
         path => "/tmp/access_log/*.log" # Path to the log files
   ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        start_position => "beginning"   # Start position for reading data
    }
}

filter {
    # Add data filtering logic here, such as grok and mutate.
}

output {
    elasticsearch {
        # Enter the cluster's private network address. For a multi-cluster node, enter multiple
addresses to allow for load balancing.
        hosts => ["192.168.xxx.xxx:9200", "192.168.xxx.xxx:9200"]
        # Name of the index that events are written to.
        index => "my_import_index"
        # Configure the following for a security-mode cluster. (Delete them if security mode is
disabled.)
        # user => "username"           # CSS cluster account
        # password => "password"       # CSS cluster password
        # ssl => true                  # Whether SSL is enabled
        # ssl_certificate_verification => false              # Whether to enable certificate
authentication
        # cacert => "/logstash/config/CloudSearchService.cer" # Absolute path to the security
certificate
        # Commonly configured parameters to mitigate compatibility issues
        ilm_enabled => false             # Disable index lifecycle management to prevent permission
errors.
        manage_template => false         # Disable template management.
    }
}
```

c. After the configuration is complete, enter **:wq** to save the configuration file.

5. Execute the configuration file to ingest data into the CSS cluster.

```
./bin/logstash -f logstash-simple.conf     # Replace the file name with the actual configuration file
name.
```

If a message similar to **Successfully started Logstash API endpoint** is returned and no error is reported, the task is running properly.

6. After ingestion, verify data integrity.

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

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

c. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.

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

The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

e. Run the following command to check the newly ingested data:

```
GET my_import_index/_count        # Check the number of records ingested.
GET my_import_index/_search       # Check the content of the ingested data.
```

If the results are consistent with the source, data ingestion is successful.

Scenario 2: Ingesting Data When Logstash Is Deployed on the Public Network

1. Prepare a jump server. In the VPC where the CSS cluster is located, prepare a Linux ECS allocated with an EIP, and configure security group rules to allow a local port (for example, 9100).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. Establish an SSH tunnel.
   a. On the jump server, run the following command to configure port mapping, which redirects requests sent to the opened port on the jump server to the destination cluster:
      ```
      ssh -g -L <Local port on the jump server>:<Private IP address of a CSS cluster node>:9200 -N -f root@<The jump server's private network address>
      ```
      For example, if the opened local port (allowed by security group rules) on the jump server is 9100, the CSS cluster node's private IP address is 192.168.0.81, and the jump server's private IP address is 192.168.0.227, the port mapping command is as follows:
      ```
      ssh -g -L 9100:192.168.0.81:9200 -N -f root@192.168.0.227
      ```
      NOTE

      If the specified CSS cluster node is unavailable, the port mapping command will fail. If the cluster contains multiple nodes, replace <Private IP address of a CSS cluster node> with the IP address of another node. If the cluster contains only one node, you have to wait until the node recovers.

      For more information about SSH tunneling, see **SSH official documents**.
   b. Verify connectivity.
      ```
      curl http://<The jump server's public IP address>:<The jump server's local port>
      ```
      If the OpenSearch version information is returned, the tunnel is established. When you configure the Logstash configuration file later, set **hosts** to <The jump server's public IP address>:<The jump server's local port>.
3. (Optional) Install Logstash on a server deployed on an external network. The following provides an installation guide for Red Hat Linux.
   a. Use yum to install the JDK first. This is because Logstash depends on Java.
      ```
      yum install java  # Replace it with the actual JDK version.
      ```
   b. Upload the downloaded Logstash package to the ECS.
   c. Use yum to install Logstash.
      ```
      yum install logstash-oss-7.10.0-x86_64.rpm
      ```
      Replace *logstash-oss-7.10.0-x86_64.rpm* with the actual Logstash installation package name.
   a. Modify the JVM configuration of Logstash to accelerate cluster data migration.

      The default heap memory of Logstash is 1 GB. You are advised to change it to half the cluster node memory.
      ```
      vim /etc/logstash/jvm.options
      -Xms4g
      -Xmx4g
      ```
4. (Optional) Prepare the security certificate. If the security mode and HTTPS are enabled for the destination CSS cluster, you must upload the cluster's security certificate to the ECS.
   a. **Obtaining the Security Certificate**.
   b. Upload the downloaded security certificate to the ECS.
5. Create a configuration file in the Logstash installation directory.
   a. Create a configuration file, for example, **logstash-simple.conf**.
      ```
      cd /logstash-x.x/ # Go to the Logstash installation directory.
      vi logstash-simple.conf     # Create a configuration file.
      ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

b. Define input, filter, and output in the configuration file.

```
input {
    # Read local log files.
    file {
        path => "/tmp/access_log/*.log" # Path to the log files
        start_position => "beginning"   # Start position for reading data
    }
}

filter {
    # Add data filtering logic here, such as grok and mutate.
}

output {
    elasticsearch {
        # Enter the local address for the SSH tunnel.
        hosts => ["<The jump server's public IP address>:<The jump server's local port>"]
        # Name of the index that events are written to.
        index => "my_import_index"
        # Configure the following for a security-mode cluster. (Delete them if security mode is
disabled.)
        # user => "username"           # CSS cluster account
        # password => "password"       # CSS cluster password
        # ssl => true                  # Whether SSL is enabled
        # ssl_certificate_verification => false              # Whether to enable certificate
authentication
        # cacert => "/logstash/config/CloudSearchService.cer" # Absolute path to the security
certificate
        # Commonly configured parameters to mitigate compatibility issues
        ilm_enabled => false             # Disable index lifecycle management to prevent permission
errors.
        manage_template => false         # Disable template management.
    }
}
```

c. After the configuration is complete, enter **:wq** to save the configuration file.

6. Execute the configuration file to ingest data into the CSS cluster.

```
./bin/logstash -f logstash-simple.conf     # Replace the file name with the actual configuration file
name.
```

If a message similar to **Successfully started Logstash API endpoint** is returned and no error is reported, the task is running properly.

7. After ingestion, verify data integrity.

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

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

c. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.

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

The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

e. Run the following command to check the newly ingested data:

```
GET my_import_index/_count        # Check the number of records ingested.
GET my_import_index/_search       # Check the content of the ingested data.
```

If the results are consistent with the source, data ingestion is successful.

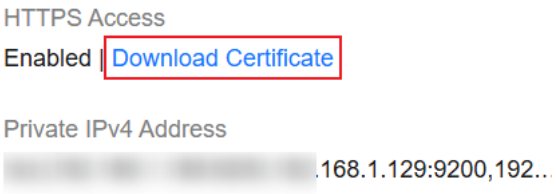
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Obtaining the Security Certificate

To access a security-mode OpenSearch cluster that uses HTTPS, perform the following steps to obtain the security certificate CloudSearchService.cer if it is required.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Overview** tab. In the **Network Information** area, click **Download Certificate** below **HTTPS Access**.

Figure 3-20 Downloading a security certificate



3.8.4 Ingesting Data Using Open-Source OpenSearch APIs

During application development, debugging, or small-scale data migration, developers frequently need to quickly write local JSON files, such as test datasets and configuration files, to an OpenSearch cluster. Compared with data migration tools such as Logstash and CDM, which are complex to configure, open-source OpenSearch APIs (such as the _bulk API) provide a more flexible and lightweight alternative, especially for ad hoc tasks. You can write data through OpenSearch Dashboards in a highly interactive manner, or run cURL commands on an ECS for batch ingestion. Both methods enable agile and efficient data ingestion into cloud-based OpenSearch clusters.

Method 1: Ingesting Data in OpenSearch Dashboards

On OpenSearch Dashboards, you can run POST commands to import data using an open-source OpenSearch API.

Applicable scenarios: development and debugging, small ad-hoc data writes, and syntax verification.

1. Log in to the OpenSearch Dashboards.
   a. Log in to the **CSS management console**.
   b. In the navigation pane on the left, choose **Clusters > OpenSearch**.
   c. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
   d. In the left navigation pane, choose **Dev Tools**.

      The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. (Optional) Create an index and specify a custom mapping to define field data types.

   For example, run the following command to create index **my_store**:

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

3. Write data to the index.

   For example, run the following command to write a record to the **my_store** index:

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

4. Verify the result.

   Check the result. If the value of **errors** is **false** and the value of **result** in each record in the **items** array is **created**, all records are successfully written.

Method 2: Ingesting Data by Running cURL Commands on an ECS

On an ECS server, you can run cURL commands to import files via an open-source OpenSearch API.

Applicable scenarios: batch ingestion of local JSON files using an automation script.

1. Prepare an ECS. Buy an ECS located in the same VPC as the destination OpenSearch cluster.

   For how to buy and use an ECS, see **Purchasing and Logging In to a Linux ECS**.

2. Obtain the private network address of the cluster.

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

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

   c. In the cluster list, obtain the target cluster's internal network address from the **Internal Network Address** column. Typical address format: *<host>*:*<port>* or *<host>*:*<port>*,*<host>*:*<port>*. Example: **10.62.179.32:9200,10.62.179.33:9200**.

      If the cluster has client nodes, only the IP addresses and ports of all the client nodes are displayed. Otherwise, the IP addresses and ports of all data nodes and cold data nodes are displayed.

3. Verify connectivity. Ping the OpenSearch cluster's private network address from the ECS. If successful, the two are connected.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

4. Prepare documents. Upload a JSON file to the ECS.

   For example, save the following data as a **test.json** file and upload it to the ECS:

   ```
   {"index": {"_index":"my_store"}}
   {"productName":"Autumn new woman blouses 2019","size":"M"}
   {"index": {"_index":"my_store"}}
   {"productName":"Autumn new woman blouses 2019","size":"L"}
   ```

5. (Optional) Create an index and specify a custom mapping to define field data types.

   For example, run the following command to create index **my_store**:

   - For a non-security mode cluster (HTTP): no authentication required.

     ```
     curl -X PUT "http://<host>:<port>/my_store" \
         -H 'Content-Type: application/json' \
         -d '{
           "settings": {
             "number_of_shards": 1
           },
           "mappings": {
             "properties": {
               "productName": {
                 "type": "text"
               },
               "size": {
                 "type": "keyword"
               }
             }
           }
         }'
     ```

   - For a security-mode cluster using HTTP: Authentication is required. If the password contains special characters like **!**, enclose the password with a pair of single quotation marks, for example, **-u 'admin':'Pass!word'**.

     ```
     curl -X PUT -u <user>:<password> "http://<host>:<port>/my_store" \
         -H 'Content-Type: application/json' \
         -d '{
           "settings": {
             "number_of_shards": 1
           },
           "mappings": {
             "properties": {
               "productName": {
                 "type": "text"
               },
               "size": {
                 "type": "keyword"
               }
             }
           }
         }'
     ```

   - For a security-mode cluster using HTTPS: Authentication is required. If the password contains special characters like **!**, enclose the password with a pair of single quotation marks, for example, **-u 'admin':'Pass!word'**. Furthermore, ignore SSL certificate authentication.

     ```
     curl -X PUT -k -u <user>:<password> "https://<host>:<port>/my_store" \
         -H 'Content-Type: application/json' \
         -d '{
           "settings": {
             "number_of_shards": 1
           },
           "mappings": {
             "properties": {
               "productName": {
                 "type": "text"
               },
     ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        "size": {
          "type": "keyword"
        }
      }
    }
  }'
```

6. Ingest the JSON file into the OpenSearch cluster:

   Run the following command in the ECS directory that stores the JSON file:

   - For a non-security mode cluster (HTTP): no authentication required.

     ```
     curl -X POST "http://<host>:<port>/_bulk" \
         -H 'Content-Type: application/json' \
         --data-binary @test.json
     ```

   - For a security-mode cluster using HTTP: Authentication is required. If the password contains special characters like **!**, enclose the password with a pair of single quotation marks, for example, **-u 'admin':'Pass!word'**.

     ```
     curl -X POST -u <user>:<password> "http://<host>:<port>/_bulk" \
         -H 'Content-Type: application/json' \
         --data-binary @test.json
     ```

   - For a security-mode cluster using HTTPS: Authentication is required. If the password contains special characters like **!**, enclose the password with a pair of single quotation marks, for example, **-u 'admin':'Pass!word'**. Furthermore, ignore SSL certificate authentication.

     ```
     curl -X POST -k -u <user>:<password> "https://<host>:<port>/_bulk" \
         -H 'Content-Type: application/json' \
         --data-binary @test.json
     ```

   NOTE

   If the designated cluster node is unavailable, the command will fail. If the cluster contains multiple nodes, replace <host> with the IP address of another node. If the cluster contains only one node, you have to wait until the node recovers.

7. Verify the result.

   After data ingestion, the following information is returned:

   ```
   {"took":204, "errors":false, "items":[...]}
   ```

   If the value of **errors** is **false**, all data has been successfully ingested.

3.9 Managing the Index Lifecycle

3.9.1 Configuring Index State Management (ISM) Policies

As a business expands, its time series data, such as logs and metrics, can grow exponentially. Storage costs can increase dramatically, and over-sized indexes can lead to poor query performance. To prevent these issues, operations teams traditionally needed to manually create new indexes, migrate old data, and delete obsolete data on a daily basis, which can be a tedious and error-prone process. Index lifecycle automation addresses these challenges by automatically managing index state transitions. This ensures read and write performance for hot data, reduces cold storage costs, and automatically deletes obsolete data. CSS OpenSearch clusters provide the Index State Management (ISM) plugin for this purpose. By configuring ISM policies, you can automate key lifecycle operations, for example, to automatically roll over an index when it reaches 50 GB, transition it to cold storage when it is 30 days old, and delete it when it is 90 days old. This helps reduce costs and improve efficiency.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Additionally, CSS provides advanced ISM features such as skipping auto rollover for empty indexes and automatic retry of failed ISM tasks. These features further reduce the operational workload and enhance ISM task reliability.

- ISM: the OpenSearch plugin that automates index lifecycle management. For more information, see the official document **Index State Management**.
- Policy: a set of rules for index lifecycle management. An example of a policy: If the number of documents in an index exceeds 10 million, the index rolls over.
- Rollover: When a write index grows beyond a specified size, age, or number of documents, a new index is automatically created to serve as the new write index. The old index changes to read-only or an archived state based on the ISM policy.

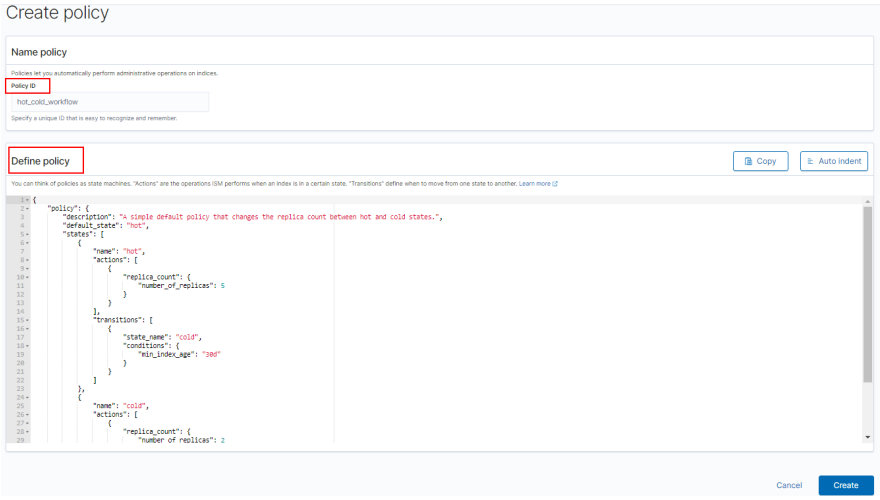
The OpenSearch Dashboards GUI may vary depending on the software version. OpenSearch Dashboards 1.3.6 is used as an example here.

Step 1: Creating an ISM Policy

Define one or more ISM policies on OpenSearch Dashboards to tell the cluster how to manage indexes.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. Expand the OpenSearch Dashboards menu in the upper-left corner, and choose **Index Management**.
5. Click **Index Policies** or **State management policies**.
6. Click **Create policy** on the right to create a policy. In the **Configuration method** dialog box, select **JSON editor** and click **Continue**. The page for creating an ISM policy is displayed.
   - **Policy ID**: Enter a custom policy name.
   - **Define policy**: Define the policy in JSON format. You can define a policy by modifying the default system example.

Figure 3-21 Configuring a policy



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

7. Click **Create**.

Step 2: Associating a Policy with Indexes

After an ISM policy is created, associate it with specific indexes to apply it. You are advised to associate a policy with an index template. This ensures that the policy is automatically applied to all indices created based on this template.

- **Method 1: Associate a policy with an index template (recommended)**

  Use this method for indexes that store time series data.

  If the **ism_template** field is configured in the JSON file in **Step 1: Creating an ISM Policy**, the created policy will be automatically applied to new indexes that meet the specified criteria (for example, any index whose name starts with logs-*). No extra step is required.

  In the following example, a policy named **policy1** is associated with an index template specified by **ism_template**. The policy will be automatically applied to new indexes whose name starts with **logs-**.

```
PUT _plugins/_ism/policies/policy1
{
  "policy": {
    "description": "an example policy",
    "default_state": "hot",
    "states": [
        // Ignore the states part.
    ],
    "ism_template": {
      "index_patterns": [
        "logs-*"
      ],
      "priority": 100
    }
  }
}
```

- **Method 2: Manually associating an existing index with a policy**

  Use this method to associate a policy with existing indexes that are currently not associated with any policy.

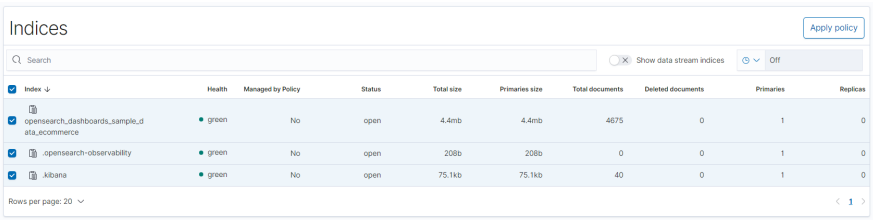
  a. Expand the OpenSearch Dashboards menu in the upper-left corner, and choose **Index Management**.

  b. Click **Indices**.

  c. In the **Indices** list, select one or more indexes.

  d. Click **Apply policy** or **Actions > Apply policy** in the upper right corner. In the displayed dialog box, select **Policy ID**.

  Figure 3-22 Apply policy



  e. Click **Apply**.

  f. On the **Managed Indices** or **Policy managed indexes** page, locate the indexes, and make sure their **Policy** is the associated policy ID.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

After you associate a policy with indexes, ISM automatically creates a job that runs every 5 minutes to execute the policy, check criteria, and change index states based on the policy.

Step 3: Managing ISM Policies

To manage deployed policies, choose **Index Management > Managed Indices** or **Index Management > Policy managed indexes** on OpenSearch Dashboards.

- View status: In the **Managed Indices** or **Policy managed indexes** list, check the policy associated with each index and the policy status.
- Retry policy: If the policy associated with an index is not working properly, select this index and click **Retry policy**.
- Change policy: To update the policy for an index, click **Change policy**.
- Remove policy: To stop lifecycle management for an index, click **Remove policy**.

(Advanced Feature) Disabling Auto Rollover for Empty Indexes

A lifecycle policy can be configured to enable automatic index rollover, that is, automatically creating a new index when an existing index meets certain conditions (for example, when it reaches a predefined age in terms of days). The new index then receives subsequent writes. By default, even if an existing index contains no documents, new indexes are still created. Over time, this can result in a large number of empty indexes. To prevent this, CSS allows you to disable automatic rollover for empty indexes. This is achieved by configuring a lifecycle policy to allow automatic rollover only for indexes that contain documents. By doing so, you can prevent excessive empty indexes from being created, thus optimizing storage utilization and enhancing overall system efficiency.

This feature is supported only by the following OpenSearch cluster versions: 1.3.6 (image version: no earlier than x.x.x_25.9.0_xxx), 2.19.0 (image version: no earlier than x.x.x_25.9.0_xxx), or 3.4.0 (image version: no earlier than 3.4.0_26.6.0_xxx).

You can disable automatic rollover for empty indexes at either the index level or the cluster level. Index-level settings take precedence over cluster-level settings. **You must configure this manually.** The commands are as follows:

- Disable empty index automatic rollover for a **specified index**:

```
PUT {index_name}/_settings
{
  "index.plugins.index_state_management.rollover.only_if_has_documents": true
}
```

- Disable empty index automatic rollover at the **cluster** level:

```
PUT _cluster/settings
{
  "persistent": {
    "plugins.index_state_management.rollover.only_if_has_documents": true
  }
}
```

(Advanced Feature) Configuring Auto Retry for ISM Tasks

ISM tasks (such as converting hot data to cold data and deleting expired indexes) can fail due to transient cluster issues like resource constraints, node restarts, or network interruptions. To address this, CSS automatically retries (reactivates)

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

failed ISM tasks at configured intervals until they succeed. This ensures the continuity and reliability of ISM tasks.

This feature is supported only by the following OpenSearch cluster versions: 1.3.6 (image version: no earlier than x.x.x_25.9.0_xxx), 2.19.0 (image version: no earlier than x.x.x_25.9.0_xxx), or 3.4.0 (image version: no earlier than 3.4.0_26.6.0_xxx).

Automatic retry of ISM tasks is enabled by default for OpenSearch clusters that meet the conditions above.

For OpenSearch clusters that do not meet these conditions, upgrade them to enable this feature. However, ISM tasks that have already failed before the upgrade will not be retried after the cluster is upgraded. You will need to manually retry them before the automatic retry can occur. The command for retrying failed ISM tasks is as follows:

```
POST _opendistro/_ism/retry/{index_name}
```

Run the following command to modify automatic retry settings for index lifecycle management tasks:

```
PUT _cluster/settings
{
  "persistent": {
    "plugins.index_state_management.coordinator.css.reactivate": true,
    "plugins.index_state_management.coordinator.css.reactivate_period": "5m",
    "plugins.index_state_management.coordinator.css.reactivate_duration": "30m",
    "plugins.index_state_management.coordinator.css.reactivate_max_duration": "12h",
    "plugins.index_state_management.coordinator.css.max_inflight_reactivate_tasks": 1000
  }
}
```

Table 3-84 Configuration parameters

Parameter	Type	Default Value	Description
plugins.index_state_management.coordinator.css.reactivate	Boolean	true	Whether to enable the Reactivate (auto retry) feature. • **true**: Enables automatic retry of index lifecycle management tasks. • **false**: Disables automatic retry of automatic retry of index lifecycle management tasks.
plugins.index_state_management.coordinator.css.reactivate_period	Time	30m	Reactivate interval. Value range: ≥ 5m
plugins.index_state_management.coordinator.css.reactivate_duration	Time	1h	Initial Reactivate interval, that is, the wait time for another retry after the initial attempt fails. It is not affected by the exponential backoff mechanism. Value range: ≥ 1m

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
plugins.index_state_management.coordinator.css.reactivate_max_duration	Time	24h	Maximum Reactivate interval, that is, the maximum wait time between retry attempts regardless of the effect of the exponential backoff mechanism. This parameter ensures that the retry interval does not increase indefinitely. Value range: ≥ 1m
plugins.index_state_management.coordinator.css.max_inflight_reactivate_tasks	Long	10000	Maximum number of tasks that can be retried at the same time. Value range: 1 to 100000

Idempotence considerations for ISM task retry

- Idempotent operations: All steps are retried, starting from the beginning, to ensure data consistency.
- Non-idempotent operations: The retry begins from where the failure occurred. If the original failure cause persists, retries may fail repeatedly, in which case, manual intervention is required.

  Examples of non-idempotent operations in OpenSearch's index lifecycle management tasks: ForceMerge, Notification, Snapshot, and Shrink. Shrink is available only in OpenSearch 2.19.0 and OpenSearch 3.4.0 (image version 3.4.0_26.6.0_xxx or later).

In the **ISM task details API**, CSS has added fields that record ISM task execution details. You can run the following command to check ISM task records, including the total number of failures, failure occurrence time, activation time, and failure causes:

```
GET /_opendistro/_ism/explain/{index_name}
```

Example response:

```
{
  "rollover-0000001" : {
    "index.opendistro.index_state_management.policy_id" : "logs-policy-rollover",
    "index" : "rollover-0000001",
    ...
    "reactivate_info" : {
      "count": 2, //Total number of failures.
      "latest_failed_time": 1764301006792, //Latest failure time.
      "latest_reactivate_time": 1764301486800, //Latest activation time.
      "failed_infos": [  //Historical execution failure records. Only the latest 10 records are retained.
        {
          "start_time": 1764299686801, //Task execution time. The earliest execution time is the index creation time.
          "failed_time": 1764299686814, //Task failure time.
          "info": {  //Task failure cause.
            "message" : "Missing rollover_alias index setting [index=rollover-0000001]"
          }
        },
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
      {
        "start_time" : 1764300286841,
        "failed_time" : 1764301006792,
        "info" : {
          "message" : "Missing rollover_alias index setting [index=rollover-0000001]"
        }
      }
    ]
  },
  ...
 }
}
```

3.9.2 Automating Index Rollover Using ISM

Time series data processing workloads, such as log analytics and real-time monitoring, often involve rapid data growth over time. If data continues to be written to a single index, the index can become excessively large, leading to degraded query performance and increased storage consumption due to obsolete data. Traditionally, managing this growth required custom scripts to split indexes, which was complex and costly. Now, by configuring index state management (ISM) policies for indexes in OpenSearch clusters, you can automate index rollover based on index size or age and have obsolete indexes deleted automatically. This helps balance query performance and storage costs.

- ISM: a plugin that automates index lifecycle management.
- Rollover: When the size or age of an index reaches a predefined threshold, a new index is automatically created, and writes are redirected to the new index.
- Alias: the logical index name. During writes, the alias always points to the current active index; and during queries, it points to the current active index plus all historical indexes.

Solution

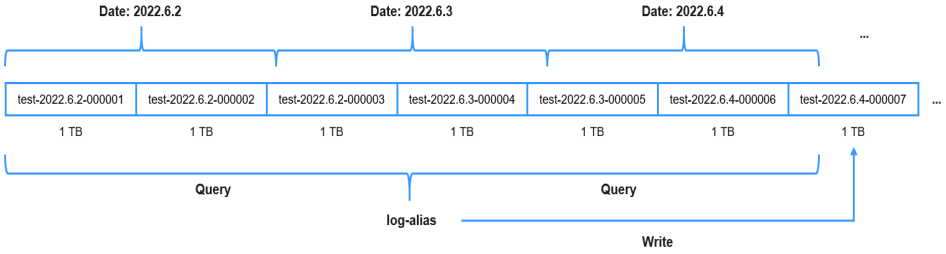
This topic describes how to configure an ISM policy to automate index rollover.

- Hot phase: Indexes handle high-concurrency writes. When the index size reaches 1 TB or its age reaches one day, a new index is automatically created, and writes are redirected to the new index.
- Warm phase: After seven days, index replicas are disabled to save resources.
- Delete phase: After 30 days, the index is automatically deleted to reclaim storage space.

Assume that an application generates around 2.4 TB of logs every day. An ISM policy can be created to optimize log storage and query performance. In this example, we define the index alias as *log-alias*. **Figure 1 log-alias data organization** shows the organization of data in this index. During queries, the alias points to all indexes whose name starts with *test*. During writes, the alias points only to the latest index.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-23 log-alias data organization



NOTE

The one day in the rollover time refers to 24 hours following the index creation time, not a calendar day.

Configuring an ISM Policy

1. Log in to OpenSearch Dashboards and go to the command execution page.
   a. Log in to the **CSS management console**.
   b. In the navigation pane on the left, choose **Clusters > OpenSearch**.
   c. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
   d. In the left navigation pane, choose **Dev Tools**.

      The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.
2. Define an ISM policy to automate index rollover, state transitions, and deletion.

   For example, create a policy named "rollover workflow": When the index size reaches 1 TB or the index age reaches one day, it rolls over automatically. When the index age reaches seven days, replicas are disabled. When it reaches 30 days, the index is deleted. Additionally, all newly created indexes whose name starts with **test** will be automatically associated with this policy.

```
PUT _plugins/_ism/policies/rollover_workflow
{
  "policy": {
    "description": "rollover test",
    "default_state": "hot",
    "states": [
      {
        "name": "hot",
        "actions": [
          {
            "rollover": {
              "min_size": "1tb",
              "min_index_age": "1d"
            }
          }
        ],
        "transitions": [
          {
            "state_name": "warm",
            "conditions": {
              "min_index_age": "7d"
            }
          }
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
      ]
    },
    {
      "name": "warm",
      "actions": [
        {
          "replica_count": {
            "number_of_replicas": 0
          }
        }
      ],
      "transitions": [
        {
          "state_name": "delete",
          "conditions": {
            "min_index_age": "30d"
          }
        }
      ]
    },
    {
      "name": "delete",
      "actions": [
        {
          "delete": {}
        }
      ]
    }
  ],
  "ism_template": {
    "index_patterns": [
      "test*"
    ],
    "priority": 100
  }
 }
}
```

After a lifecycle policy is created, run the following command to query the policy details:

```
GET _plugins/_ism/policies/rollover_workflow
```

3. Associate the policy with an index template so that new indexes created using this template will automatically inherit the ISM policy and alias.

   For example, create an index template named **template_test**, and use **log_alias** for index rollover.

```
PUT _template/template_test
{
  "index_patterns": "test*",
  "settings": {
    "number_of_replicas": 1, // Number of index replicas
    "number_of_shards": 1,    // Number of index shards
    "index.opendistro.index_state_management.rollover_alias": "log_alias" // Index lifecycle policy name
  },
  "mappings": {
    "properties": {
      "name": {
        "type": "text"
      }
    }
  }
}
```

   After an index template is created, you can run the following command to query the template details:

```
GET _template/template_test
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

4. Manually create the first index that will start the automatic rollover process.

   For example, create the first index, set the date format, specify **aliases**, and set **is_write_index** to **true**. The index will automatically use the index template *template_test* and inherit the lifecycle policy *rollover_workflow* through the template.

   ```
   PUT %3Ctest-%7Bnow%2Fd%7D-000001%3E
   {
     "aliases": {
       "log_alias": {
         "is_write_index": true // Tells the customer to direct all write requests aiming at log_alias to this index.
       }
     }
   }
   ```

   The index name above is the result of URL-coding **<test-{now/d}-000001>**. By default, the index name contains the creation date. For example, if an index was created on 2022-06-02, the index name is **test-2022.06.02-000001**.

5. Write in data. The alias **log_alias** is used during data writes, and **log_alias** always points to the last index.

   ```
   POST log_alias/_bulk
   {"index":{}}
   {"name":"name1"}
   {"index":{}}
   {"name":"name2"}
   {"index":{}}
   {"name":"name3"}
   {"index":{}}
   {"name":"name4"}
   {"index":{}}
   {"name":"name5"}
   {"index":{}}
   {"name":"name6"}
   ```

6. Query data and check whether the rollover has taken effect.

   - One day after the indexes are created, check the indexes starting with **test**.

     ```
     GET _cat/indices/test*?s=i
     ```

     There are supposed to be at least two indexes, for example:

     ```
     green open test-<Date>-000001 r8ab5NX6T3Ox_hoGUanogQ 1 1 6 0 416b 208b
     green open test-<Date>-000002 sfwkVgy8RSSEw7W-xYjM2Q 1 1 0 0 209b 209b
     ```

     In the preceding information, **test-<Date>-000001** is the index created in **4**, and **test-<Date>-000002** is the index generated through rollover.

   - To query the index associated with the alias **log_alias**, run the following command:

     ```
     GET _cat/aliases/log_alias?v
     ```

     The alias is supposed to point to multiple indexes, for example:

     ```
     alias     index                filter routing.index routing.search is_write_index
     log_alias test-<Date>-000001   -      -             -              false
     log_alias test-<Date>-000002   -      -             -              true
     ```

FAQ: How Do I Skip Rollover for an Index?

Scenarios

- A manual rollover is performed: A time-based ISM policy was configured to automate index rollover. Before the rollover criteria were met, however, a manual rollover was performed. When the auto rollover time configured by ISM was reached, an automatic rollover was attempted but failed because the

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

index had already been rolled over manually. As a result, the ISM task stopped. To solve this problem, you need to skip the index rollover and then retry the ISM task to resume automatic index rollover.

- For an index that has already been rolled over automatically, a remove or add policy operation is performed: After the rollover policy is modified (through the remove policy or add policy operation), the ISM task restarts from the beginning. When rollover starts, the task fails. To solve this, you also need to skip index rollover and then retry the ISM task to resume automatic index rollover.

WARNING

After rollover is skipped for an index, ISM will no longer attempt rollover or generate rolled-over indexes. This means skipping index rollover may lead to data loss. Please exercise caution.

Procedure

1. If an ISM task stops due to an index rollover failure, run the following command to skip index rollover:
   ```
   PUT {index_name}/_settings
   {
     "index.plugins.index_state_management.rollover_skip": true
   }
   ```
   If **true** is returned, the configuration is successful.
2. After index rollover is skipped, run the following command to retry the ISM task:
   ```
   POST _plugins/_ism/retry/{index_name}
   ```
   If the following information is returned, the retry is successful:
   ```
   {
     "updated_indices": 1,
     "failures": false,
     "failed_indices": []
   }
   ```

3.9.3 Decoupling Storage and Compute Using ISM

In log analytics and monitoring scenarios, storage costs for historical data accumulate over time. However, most cold data is accessed only occasionally for auditing or backtracking purposes. Storing such data on expensive local SSDs over extended periods does not make economic sense. To balance storage costs with long-term retention requirements, CSS provides a storage-compute decoupling feature for OpenSearch clusters. By configuring an index state management (ISM) policy, coupled with index freezing, you can have eligible indexes automatically transitioned from local SSDs to lower-cost object storage. The data remains searchable, although query latency may increase.

Solution

This topic describes how to configure an ISM policy to decouple storage and compute.

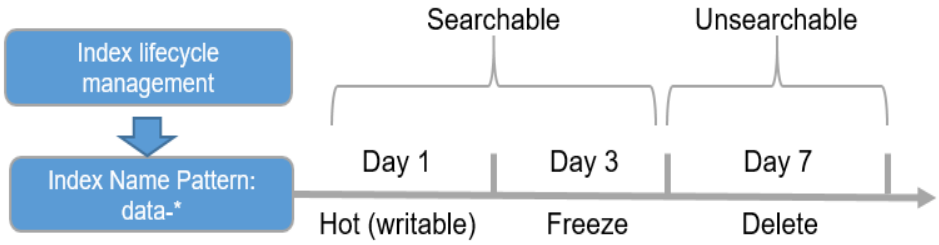
- Hot phase: Data is stored on high-performance local disks to support high-concurrency read and write operations.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- Warm phase: The index is frozen, and data is dumped to OBS. Local disks only retain necessary metadata, and query requests are directly sent to OBS. (Initial queries may experience higher latency, but subsequent queries can accelerate with the use of a cache.)
- Delete phase: When the data retention period expires, the index is automatically deleted to reclaim storage space.

In this example, an ISM policy is created for an OpenSearch cluster. According to this policy, a newly created index is automatically frozen in three days, with data dumped to OBS; and deleted in seven days.

Figure 3-24 Storage-compute decoupling



Constraints

Only OpenSearch 1.3.6, 2.19.0, and 3.4.0 clusters support decoupled storage and compute.

Configuring an ISM Policy

1. Log in to OpenSearch Dashboards and go to the command execution page.
   a. Log in to the **CSS management console**.
   b. In the navigation pane on the left, choose **Clusters > OpenSearch**.
   c. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
   d. In the left navigation pane, choose **Dev Tools**.

      The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.
2. Define an ISM policy and associate it with an index template to automate storage-compute decoupling for new indexes created using that template.

   For example, create a policy named *hot_warm_policy*. According to this policy, a newly created index is automatically frozen in three days, with data dumped to OBS; and deleted permanently in seven days. Associate the policy with an index template so that new indexes whose name starts with **data** will automatically inherit this policy.

```
PUT _plugins/_ism/policies/hot_warm_policy
{
  "policy": {
    "description": "hot warm delete workflow",
    "error_notification": null,
    "default_state": "hot",
    "states": [
      {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
      "name": "hot",
      "actions": [],
      "transitions": [
        {
          "state_name": "warm",
          "conditions": {
            "min_index_age": "3d"
          }
        }
      ]
    },
    {
      "name": "warm",
      "actions": [
        {
          "freeze_low_cost": {} // Freeze the index.
        }
      ],
      "transitions": [
        {
          "state_name": "delete",
          "conditions": {
            "min_index_age": "7d"
          }
        }
      ]
    },
    {
      "name": "delete",
      "actions": [
        {
          "delete": {}
        }
      ],
      "transitions": []
    }
  ],
  "ism_template": {      // Associate with an index template.
    "index_patterns": [
      "data*"
    ],
    "priority": 100
  }
 }
}
```

3. Write data to an index.

   For example, create an index named data-2022-06-06 and write data to it in batches.

```
POST data-2022-06-06/_bulk
{"index":{}}
{"name":"name1"}
{"index":{}}
{"name":"name2"}
{"index":{}}
{"name":"name3"}
{"index":{}}
{"name":"name4"}
{"index":{}}
{"name":"name5"}
{"index":{}}
{"name":"name6"}
```

   The index will automatically apply the ISM policy **hot_warm_policy**.

4. Query data and check whether the ISM policy is automatically applied.

   a. Three days after index creation, query the list of frozen indexes and check whether indexes created three days ago have been frozen.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
GET _cat/freeze_indices?s=i&v
```

If the returned result contains **data-2022-06-06**, the index has been dumped to OBS.

```
health status index                uuid                   pri rep docs.count docs.deleted store.size
pri.store.size
green  open   data-2022-06-06 x8ab5NX6T3Ox_xoGUanogQ   1   1          6            0
7.6kb          3.8kb
```

b. Seven days after index creation, query the index list and check whether indexes created seven days ago have been deleted.

```
GET /_cat/indices?expand_wildcards=open,closed
```

If the returned result does not contain **data-2022-06-06**, the index has been deleted.

3.10 Search and Analytics

3.10.1 Using DSL for Search

When you need to interact with an OpenSearch cluster to perform complex full-text search or data analysis, the domain-specific language (DSL) is your best choice. As a JSON-based query language native to OpenSearch, DSL enables you to express sophisticated query logic with clarity and precision while giving you fine-grained control over how your queries are executed.

What Is DSL?

The OpenSearch Query DSL is a JSON-based query language that defines the structure and semantics of search and data retrieval requests. It contains two contexts:

- Query context: In the query context, a query clause answers the question "How well does this document match this query clause?" Besides deciding whether or not the document matches, the query clause also calculates a relevance score in the **_score** metadata field. One example of such query clauses: match.
- In a filter context, a query clause answers the question "Does this document match this query clause?" The answer is a simple Yes or No. No scores are calculated, but frequently used filters will be cached automatically by OpenSearch, to speed up performance. Examples of such queries: term and range.

DSL queries are executed in Dev Tools of OpenSearch Dashboards. Both the request body and returned information are in JSON.

This topic lists some of the most commonly used DSL query clauses. For more, see **Query DSL**.

Basic Query: match_all

Use **match_all** to match all documents in the index. It is equivalent to **SELECT * FROM table** in SQL. Use it when you want to search all documents.

For example, run the following command to match all documents in the **test** index:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
GET /test/_search
{
  "query": {
    "match_all": {}
  }
}
```

Compound Queries: Combining Multiple Query Clauses

Use a bool query with clauses such as must and filter to construct compound query conditions. This is similar to the **where** clause in SQL. Use this query when you need to apply multiple conditions to filter documents.

For example, run the following command to retrieve all documents whose status is **published** and whose publish_date is later than **2015-01-01** (filter condition), and whose title or content contains **Search** (search condition).

```
GET /_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "title": "Search"
          }
        },
        {
          "match": {
            "content": "search"
          }
        }
      ],
      "filter": [
        {
          "term": {
            "status": "published"
          }
        },
        {
          "range": {
            "publish_date": {
              "gte": "2015-01-01"
            }
          }
        }
      ]
    }
  }
}
```

The differences between **must** and **filter** are as follows:

- **must**: Specifies conditions that documents must meet. These conditions affect the relevance score of documents, and documents with higher scores are ranked higher in the results.
- **filter**: Also specifies conditions that documents must meet, but these conditions do not affect the relevance score. **filter** is typically more efficient than **must**, making it better suited for filtering structured data.

For conditions that do not require relevance scoring (such as status, time range, and category), use **filter** to improve query performance.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Aggregations

The **aggs (such as the terms aggregation)** structure is used to perform aggregation queries. It is similar to the **Group by** clause in SQL. Use it when you want to group documents to calculate metrics.

For example, run the following command to count how many times different titles appear in the **test** index:

```
GET /test/_search
{
  "aggs": {
    "titles": {
      "terms": {
        "field": "title.keyword"
      }
    }
  }
}
```

The reason for using **title.keyword** is as follows:

- Text fields are used for full-text retrieval (they undergo tokenization). For example, "Hello World" will be split into "hello" and "world". If you attempt to aggregate on a text field directly, an error (Fielddata is disabled) will be returned. This is because it is an extremely memory-intensive operation.
- Keyword fields are used for exact match and aggregations. It keeps "Hello World" as a single value.

By default, the cluster automatically creates multi-fields for strings. That means **title** is used for search, and **title.keyword** for aggregations.

3.10.2 Using SQL for Search

For developers and data analysts who are familiar with relational databases like MySQL, the native, JSON-based OpenSearch query DSL may present a steep learning curve. In particular, constructing complex aggregation queries can be an error-prone process. To address this challenge, CSS OpenSearch clusters integrate the Open Distro for OpenSearch SQL plugin. This plugin enables users to query, aggregate, and analyze data in OpenSearch using standard SQL syntax. It also supports interconnection with popular BI tools like Tableau through a JDBC driver, enabling a seamless data exploration experience.

How the Feature Works

Open Distro for OpenSearch SQL is essentially a translator between SQL and OpenSearch DSL. It works as follows:

1. Parsing: It receives and validates SQL queries submitted by users.
2. Translation: It converts SQL queries into native OpenSearch DSL queries.
3. Execution: It sends DSL queries to the cluster for execution.
4. Return: It returns the results in JSON or CSV.

For more information, see **SQL Functions**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

SQL Query Guide

- **Method 1: Running SQL Queries in OpenSearch Dashboards (Recommended)**

  Use this method for development, debugging, and ad-hoc queries.

  – Perform a basic query.

    For example, run the following command to search the my-index index for documents where the age field exceeds 20, returning up to 10 results:

    ```
    POST _plugins/_sql
    {
      "query": "SELECT * FROM my-index WHERE age > 20 LIMIT 10"
    }
    ```

  – Export data in CSV. To export data to an Excel document for analysis, specify **format=csv**.

    For example, run the following command to retrieve 10 documents from the my-index index based on the name and age fields and return the results in CSV format:

    ```
    POST _plugins/_sql?format=csv
    {
      "query": "SELECT name, age FROM my-index LIMIT 10"
    }
    ```

  – Convert SQL to DSL. To learn how to write DSL queries or analyze the performance of SQL queries, use the **_explain** API to check the translated DSL queries.

    For example, run the following command to convert a SQL query into the equivalent DSL query:

    ```
    POST _plugins/_sql/_explain
    {
      "query": "SELECT * FROM my-index WHERE age > 20"
    }
    ```

- **Method 2: Running SQL Queries via cURL Commands on a Server**

  Use this method when you need to incorporate OpenSearch queries into automated scripts or scheduled tasks.

  For example, run the following command to retrieve 10 records from the **my-index** index. The following is an example of the code used in a security-mode cluster that uses HTTP:

  ```
  curl -XPOST http://<cluster IP address>:9200/_plugins/_sql \
    -u <username>:<password> \
    -H 'Content-Type: application/json' \
    -d '{"query": "SELECT * FROM my-index LIMIT 10"}'
  ```

Supported SQL Syntax

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

- Statements

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-85 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

CAUTION

Large UNION and MINUS queries can cause high resource consumption, potentially leading to degrading cluster performance or even service interruptions. To mitigate this risk, you are advised to optimize the query structure or split large queries into smaller batches.

- Conditions

Table 3-86 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 ('Bob', 'David')
Not	SELECT * FROM my-index WHERE name NOT IN ('Bob')
Between	SELECT * FROM my-index WHERE age BETWEEN 20 AND 30
Aliases	SELECT avg(age) AS Average_Age FROM my-index

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Condition	Example
Date	SELECT * FROM my-index WHERE birthday='1990-11-15'
Null	SELECT * FROM my-index WHERE name IS NULL

- Aggregations

  Table 3-87 Aggregations

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

- Include and exclude fields

  Table 3-88 Include and exclude fields

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

- Functions

  CAUTION

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

  Table 3-89 Functions

Function	Example
floor	SELECT floor(number) AS Rounded_Down FROM my-index

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

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

- Joins

CAUTION

Supported join types: INNER JOIN, LEFT JOIN, and CROSS JOIN.

Hard limits:

- Only two tables (indexes) can be joined.
- An alias must be specified for each table, for example, **FROM people p JOIN index b**.
- The ON condition supports only AND combinations, not OR or nested conditions.
- GROUP BY or ORDER BY cannot be used in a JOIN query.
- LIMIT and OFFSET cannot be used together. A negative example: **LIMIT 25 OFFSET 25**
- Fields from multiple indexes cannot be combined in the WHERE clause.

  A positive example:

  ```
  WHERE (a.type1 > 3 OR a.type1 < 0) AND (b.type2 > 4 OR b.type2 < -1)
  ```

  A negative example:

  ```
  WHERE (a.type1 > 3 OR b.type2 < 0) AND (a.type1 > 4 OR b.type2 < -1)
  ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-90 Joins

Join	Example
Inner join	SELECT s.firstname, s.lastname, s.gender, sc.name FROM student s JOIN school sc ON sc.name = s.school_name WHERE s.age > 20
Left outer join	SELECT s.firstname, s.lastname, s.gender, sc.name FROM student s LEFT JOIN school sc ON sc.name = s.school_name
Cross join	SELECT s.firstname, s.lastname, s.gender, sc.name FROM student s CROSS JOIN school sc

- Show

  Show commands display indexes and mappings that match an index pattern. You can use ***** or **%** for wildcards.

  Table 3-91 Show

Show	Example
Show tables like	SHOW TABLES LIKE logs-*

BI Integration via JDBC Driver

If you are using Tableau, Power BI, or a self-managed Java application, you can connect to a CSS cluster using the JDBC driver.

Download the JDBC driver from the **GitHub repository**.

3.10.3 Configuring Custom Dictionaries

In daily search scenarios, OpenSearch's built-in dictionaries often fail to recognize domain-specific terms. Such terms may be split into separate words (or separate characters in Chinese), resulting in low search recall and inaccurate results. Additionally, meaningless high-frequency words (such as stop words), if not filtered out, can introduce noise and reduce search efficiency. Furthermore, when users search for "cell phone", they may expect documents containing synonyms such as "mobile phone" to also be returned, but built-in dictionaries do not support synonym expansion. To address these issues, CSS provides the IK analysis plugin (analysis-ik) and dynamic synonym plugin (analysis-dynamic-synonym). By configuring custom dictionaries, you can enable OpenSearch to accurately recognize domain-specific terms, filter irrelevant words, and expand synonyms, delivering a smarter search experience tailored to business use cases.

How the Feature Works

Dictionaries are core resources that guide analyzers in recognizing special terms. CSS supports the following three types of dictionaries:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- Main word dictionary: Defines special terms that should be recognized as independent terms rather than split into separate words by the analyzer.
- Stop word dictionary: Defines meaningless words that should be filtered out in order to reduce noise.
- Synonym dictionary: Defines groups of semantically equivalent terms to enable synonym-based search expansion.

Dictionaries take effect only when used together with analyzers. CSS provides a built-in standard analyzer, IK analyzers, and a synonym filter. IK analyzers are implemented through the IK analysis plugin, and the synonym filter is implemented through the dynamic synonym plugin. **Table 3-92** describes the mapping between dictionaries and analyzers.

Table 3-92 Mapping between dictionaries and analyzers

Analyzer	Description	Dictionary Dependency
Default analyzer: Standard Analyzer	A standard analyzer based on Unicode text segmentation. It automatically converts text to lowercase and splits Chinese text into individual characters. • For example, **Hello World** is split into **hello,world**. • Advantage: Simple configuration; no dictionary maintenance required. • Disadvantage: Poor performance in Chinese segmentation. • Applicable scenario: Indexing Western languages or scenarios with low requirements for Chinese text analysis.	No dictionary required
IK Analyzer ik_smart	A coarse-grained analyzer that generates only the most reasonable segmentation results without enumerating all possible combinations. • Example: Text is segmented into the most likely word combinations rather than all possible combinations. • Advantage: Provides good Chinese text segmentation, reduces noise, and improves search precision. • Disadvantage: May miss some less commonly used terms or word combinations. • Applicable scenario: Query analysis during the search phase.	Requires main word and stop word dictionaries (built-in dictionaries are provided; custom dictionaries are also supported)

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Analyzer	Description	Dictionary Dependency
IK Analyzer ik_max_word	A fine-grained analyzer that enumerates all possible word combinations and generates the finest-grained segmentation results. • Example: Text is segmented into all possible word combinations. • Advantage: Provides good Chinese segmentation, covers more term combinations, and improves search recall. • Disadvantage: Increases index size and slightly reduces write performance. • Applicable scenario: Index building for full-text search during data ingestion.	Requires main word and stop word dictionaries (built-in dictionaries are provided; custom dictionaries are also supported)
Synonym filter dynamic_synonym	A synonym expansion filter that dynamically loads synonym dictionaries and performs semantic expansion or correction on segmentation results. • Example: Searching for **cell phone** also returns documents containing **mobile phone** and **mobile device**. • Advantage: Enables synonym expansion search and improves search recall. • Disadvantage: Must be used together with the standard analyzer or IK analyzer. • Applicable scenarios: Semantic expansion search and term correction.	Requires a synonym dictionary (supports custom dictionaries only)

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

CAUTION

The IK Analyzer is designed primarily for Chinese text. When used to analyze English text, only the following special characters are treated as regular characters and preserved: # & + - . @ _. Other special characters are treated as segmentation points where text can be split into different parts. Example:

- **access address: example@domain.com** is split into **access,address,example@domain.com**.
- **set the parameter value to user|domain** is split into **set,parameter,value,user,domain**. (**the** and **to** are in the built-in Static Stop Word Dictionary and are filtered out during word segmentation.)

How Custom Dictionaries Work

The workflow of custom dictionaries is as follows:

1. **Dictionary loading**: After uploading dictionary files to an OBS bucket, configure the dictionaries for the cluster. The system then loads the dictionaries to all nodes in the cluster.
2. **Main word dictionary matching**: The analyzer first matches entries in the main word dictionary and keeps matched entries as complete terms during segmentation, preventing special terms from being split.
3. **Stop word filtering**: Entries matching the stop word dictionary in the segmentation results are discarded and are not indexed.
4. **Synonym expansion**: The segmentation results are processed by the synonym filter. Entries matching the synonym dictionary are expanded into additional synonym terms to enable semantic expansion search.
5. **Scope of update**: After a dictionary is updated, newly written data is automatically analyzed using the new dictionary. Historical index data is not rebuilt automatically. To re-analyze historical data using the updated dictionary, manually run the **_update_by_query** command.

Dictionary File Specifications

Prepare dictionary files according to the following specifications. Files that do not meet these requirements will fail to be loaded.

Table 3-93 Dictionary file specifications

Item	Requirement
File format	**.txt**
File encoding	UTF-8 without BOM

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Item	Requirement
Content format	• Main word and stop word dictionaries: One term per line. Example: cloud computing search engine • Synonym dictionary: One synonym group per line. Use commas (,) to separate bidirectional synonyms, and => to separate unidirectional synonyms. Example: happy,glad,pleased cell phone=>mobile phone With bidirectional synonyms, searching for **happy** returns documents containing **happy**, **glad**, and **pleased**. With unidirectional synonyms, searching for **cell phone** only returns documents containing **mobile phone**.
English terms	Must be in lowercase.
File size	Maximum 100 MB per file.

Prerequisites

- The target cluster is in the **Available** state and has no ongoing tasks.
- The account you are using has the following permissions (choose **Permissions** > **Policies/Roles** on the IAM console to check your permissions):
  - Permission to configure custom dictionaries:
    ```
    "css:IKThesaurus:*"
    ```
  - To update dictionaries, the account must also have read access to the OBS bucket and objects:
    ```
    obs:bucket:getBucketLocation
    obs:bucket:getBucketStoragePolicy
    obs:bucket:listAllMyBuckets
    obs:object:getObject
    ```
  - If the OBS bucket uses SSE-KMS encryption, the following KMS permissions are also required:
    ```
    "kms:cmk:create",
    "kms:dek:create",
    "kms:cmk:get",
    "kms:dek:decrypt",
    "kms:cmk:list"
    ```
- To update dictionaries, prepare dictionary files that comply with the required format and upload them to an OBS bucket. For details, see **Uploading an Object**.

  The OBS bucket must be in the same region as the CSS cluster, and its storage class must be **Standard**.

Accessing the Custom Dictionary Configuration Page

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Cluster Settings > Custom Word Dictionaries**.

Configuring Custom Dictionaries

If built-in dictionaries do not meet your business requirements for text analysis, you can configure custom dictionaries.

1. On the Custom Word Dictionaries tab, configure custom dictionaries as required.

Table 3-94 Configuring custom dictionaries

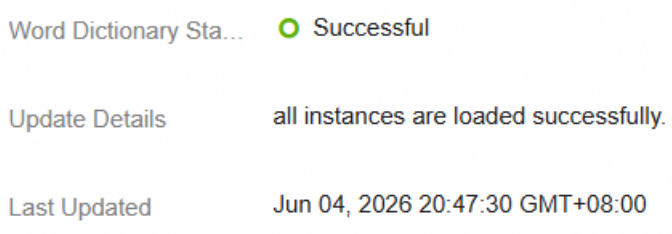
Parameter	Description
OBS Bucket	Select the OBS bucket that stores the dictionary file. This parameter is required only when you update dictionaries. If no OBS buckets meet your requirements, click **Create Bucket** to go to the OBS console and create one. For details, see **Creating a Bucket**.
Main Word Dictionary	A custom collection of main words. It is empty initially. • Default value: **No Update**, meaning not to configure this dictionary. • To create a custom dictionary or update an existing one, choose **Update > Select**, select a dictionary file that complies with the required format from the OBS bucket, and click **OK**. • To delete a dictionary, click **Do Not Use**.
Stop Word Dictionary	A custom collection of stop words. It is empty initially. • Default value: **No Update**, meaning not to configure this dictionary. • To create a custom dictionary or update an existing one, choose **Update > Select**, select a dictionary file that complies with the required format from the OBS bucket, and click **OK**. • To delete a dictionary, click **Do Not Use**.
Synonym Dictionary	A custom collection of synonyms. It is empty initially. • Default value: **No Update**, meaning not to configure this dictionary. • To create a custom dictionary or update an existing one, choose **Update > Select**, select a dictionary file that complies with the required format from the OBS bucket, and click **OK**. • To delete a dictionary, click **Do Not Use**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. Click **Save**. In the displayed dialog box, click **OK** to start updating dictionaries.

   The dictionary configuration information is displayed below. Wait for approximately 1 minute. If the dictionary status changes from **Updating** to **Successful**, the dictionary update is complete.

   Figure 3-25 Dictionary status

3. (Optional) After a dictionary is updated, newly written data uses the updated dictionary for analysis. Historical index data is not rebuilt automatically. To re-analyze and re-index historical data, manually run the following command:

   ```
   POST /{my_index}/_update_by_query
   ```

Updating Built-in Dictionaries

You can update built-in dictionaries if you need to modify or disable them.

1. On the **Custom Word Dictionaries** page, toggle on the switch next to **Modify Preset Word Dictionaries**.

   CAUTION

   If **Modify Preset Word Dictionaries** is not displayed, the current cluster version does not support built-in dictionary updates. In this case, you are advised to upgrade the cluster version or create a new cluster and then migrate data to it.

2. Update built-in dictionaries as required.

   Table 3-95 Updating built-in dictionaries

Parameter	Description
OBS Bucket	Select the OBS bucket that stores the dictionary file. This parameter is required only when you update dictionaries. If no OBS buckets meet your requirements, click **Create Bucket** to go to the OBS console and create one. For details, see **Creating a Bucket**.

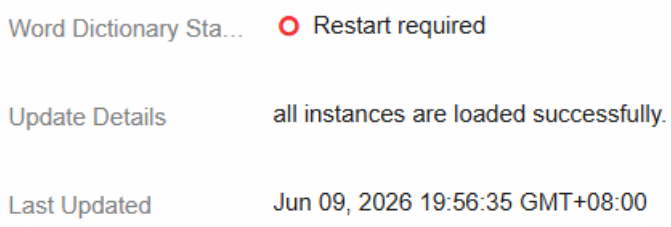
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Static Main Word Dictionary	A built-in collection of commonly used main words. • Default value: **No Update**, meaning not to configure this dictionary. • To update this dictionary, choose **Update** > **Select**, select a dictionary file that complies with the required format from the OBS bucket, and click **OK**. • To delete this dictionary, click **Do Not Use**.
Static Stop Word Dictionary	A built-in collection of commonly used stop words. • Default value: **No Update**, meaning not to configure this dictionary. • To update this dictionary, choose **Update** > **Select**, select a dictionary file that complies with the required format from the OBS bucket, and click **OK**. • To delete this dictionary, click **Do Not Use**.
Extra Main Word Dictionary	A built-in collection of uncommon main words. • Default value: **No Update**, meaning not to configure this dictionary. • To update this dictionary, choose **Update** > **Select**, select a dictionary file that complies with the required format from the OBS bucket, and click **OK**. • To delete this dictionary, click **Do Not Use**.
Extra Stop Word Dictionary	A built-in collection of uncommon stop words. • Default value: **No Update**, meaning not to configure this dictionary. • To update this dictionary, choose **Update** > **Select**, select a dictionary file that complies with the required format from the OBS bucket, and click **OK**. • To delete this dictionary, click **Do Not Use**.

3. Click **Save**. In the displayed dialog box, click **OK** to start updating dictionaries.

   The dictionary configuration information is displayed below. Wait for approximately 1 minute. If the dictionary status changes from **Updating** to **Restart required**, the dictionary update is complete.

   Figure 3-26 Dictionary status



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

4. Click **Restart** in the upper-right corner of the page to restart the cluster and apply changes made to the built-in dictionaries. For details, see **Restarting a Cluster**.
5. (Optional) After a dictionary is updated, newly written data uses the updated dictionary for analysis. Historical index data is not rebuilt automatically. To re-analyze and re-index historical data, manually run the following command:

```
POST /{my_index}/_update_by_query
```

Verifying Main Word and Stop Word Dictionaries

After configuring the main word dictionary and stop word dictionary, you can use the /_analyze API to test the text segmentation results.

Scenario: The custom main word dictionary and stop word dictionary have been configured with the necessary main words and stop words.

1. On the **Custom Word Dictionaries** page, click **Dashboards** in the upper-right corner to log in to OpenSearch Dashboards.
2. On OpenSearch Dashboards, expand the menu in the upper-left corner, and choose **Dev Tools**.
3. Run the following commands to check the segmentation results of **ik_smart** and **ik_max_word**:
   - **ik_smart (coarse-grained) segmentation**

     Example request:

     ```
     POST /_analyze
     {
       "analyzer":"ik_smart",
       "text":"Sample Chinese text"
     }
     ```

     Expected result: The main words in the provided text are recognized as independent terms, and stop words are filtered out.

     ```
     {
       "tokens": [
         {
           "token": "main_word_1",
           "start_offset": 0,
           "end_offset": 4,
           "type": "CN_WORD",
           "position": 0
         },
         {
           "token": "main_word_2",
           "start_offset": 5,
           "end_offset": 8,
           "type": "CN_WORD",
           "position": 1
         }
       ]
     }
     ```

   - **ik_max_word (fine-grained) segmentation**

     Example request:

     ```
     POST /_analyze
     {
       "analyzer":"ik_max_word",
       "text":"Sample Chinese text"
     }
     ```

     Expected result: The main words in the provided text are segmented into multiple possible terms, and stop words are filtered out.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
{
  "tokens": [
    { "token": "word_1", "start_offset": 0, "end_offset": 4, "type": "CN_WORD", "position": 0 },
    { "token": "word_2", "start_offset": 0, "end_offset": 2, "type": "CN_WORD", "position": 1 },
    { "token": "word_3", "start_offset": 0, "end_offset": 1, "type": "CN_WORD", "position": 2 },
    { "token": "word_4", "start_offset": 1, "end_offset": 3, "type": "CN_WORD", "position": 3 },
    { "token": "word_5", "start_offset": 2, "end_offset": 4, "type": "CN_WORD", "position": 4 },
    { "token": "word_6", "start_offset": 3, "end_offset": 4, "type": "CN_WORD", "position": 5 },
    { "token": "word_7", "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 }
  ]
}
```

Verifying Synonym Expansion

After configuring the synonym dictionary, you can test the dictionary through synonym search.

Scenario: The synonym dictionary contains synonyms "happy, glad".

1. On the **Custom Word Dictionaries** page, click **Dashboards** in the upper-right corner to log in to OpenSearch Dashboards.
2. On OpenSearch Dashboards, expand the menu in the upper-left corner, and choose **Dev Tools**.
3. Run the following command to create the **test** index and configure the synonym filter.

```
PUT test
{
    "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"
            }
        }
    }
}
```

4. Run the following command to verify synonym filter configuration.

```
GET /test/_analyze
{
  "analyzer": "ik_synonym",
  "text": ["I'm very happy"]
}
```

Expected result: Although the text uses "happy", the segmentation results also contain "glad" based on the synonym mapping.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
{
  "tokens": [
    { "token": "I'm", "start_offset": 0, "end_offset": 2, "type": "CN_WORD", "position": 0 },
    { "token": "happy", "start_offset": 2, "end_offset": 4, "type": "CN_WORD", "position": 1 },
    { "token": "glad", "start_offset": 2, "end_offset": 4, "type": "SYNONYM", "position": 1 }
  ]
}
```

Preset Dictionaries

To help you quickly get started with the IK analyzer, CSS provides four built-in dictionaries, as listed in **Table 3-96**. **If these preset main word and stop word dictionaries already meet your requirements, there is no need to configure custom dictionaries.**

Table 3-96 Preset dictionaries

Preset Dictionary	Description	Download URL
Static main word dictionary	A collection of common main words (default main words of the IK analyzer)	**https://github.com/infinilabs/analysis-ik/blob/master/config/main.dic**
Static stop word dictionary	A collection of common stop words (default stop words of the IK analyzer)	**https://github.com/infinilabs/analysis-ik/blob/master/config/stopword.dic**
Extra main word dictionary	A collection of uncommon main words (extended main words of the IK analyzer)	**https://github.com/infinilabs/analysis-ik/blob/master/config/extra_main.dic**
Extra stop word dictionary	A collection of uncommon stop words (extended stop words of the IK analyzer)	**https://github.com/infinilabs/analysis-ik/blob/master/config/extra_stopword.dic**

Related Documents

- **Improving Keyword Search Accuracy with the IK Analyzer and Custom Dictionaries**: Configure custom main word and stop word dictionaries with the IK analyzer to preserve domain-specific terms and filter irrelevant words, enabling more precise keyword search.
- **Improving Search Recall with the Synonym Filter and a Custom Synonym Dictionary**: Configure a custom synonym dictionary and use the

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

dynamic_synonym synonym filter to automatically expand synonyms during query and indexing, enabling synonym-based search across equivalent terms.

3.10.4 Configuring Simplified-Traditional Chinese Conversion Search

In cross-region business scenarios, users may need to search Traditional Chinese content using Simplified Chinese keywords, or search Simplified Chinese content using Traditional Chinese keywords. For example, a user may enter a Simplified Chinese keyword and expect documents containing its Traditional Chinese equivalent to be returned. However, due to differences in character forms between Simplified and Traditional Chinese, direct searches may fail to accurately retrieve the expected documents, resulting in a poor user experience. To address this issue, CSS provides the Simplified-Traditional Chinese conversion plugin (analysis-stconvert) for OpenSearch clusters. By automatically converting between Simplified and Traditional Chinese during text analysis, the plugin eliminates the need to modify business data or implement custom conversion logic. You only need to configure the corresponding analyzer when creating an index to enable cross-form Chinese search, such as searching Traditional Chinese documents with Simplified Chinese keywords or vice versa.

How the Feature Works

The Simplified-Traditional Chinese conversion plugin (analysis-stconvert) supports bidirectional conversion between Simplified and Traditional Chinese. The plugin is pre-installed in CSS and ready for use out of the box. Its core capability is to convert characters during text analysis, unifying Simplified and Traditional Chinese at the index level to enable cross-form search and matching.

Table 3-97 Supported conversion types

Conversion Type	Description	Example
s2t	Converts Simplified Chinese to Traditional Chinese.	**A Simplified Chinese term** > **Its Traditional Chinese equivalent**
t2s	Converts Traditional Chinese to Simplified Chinese. This is the commonly used type. The IK analyzer dictionaries are primarily based on Simplified Chinese, so IK analyzers generally provide better analysis results for Simplified Chinese text.	**A Traditional Chinese term** > **Its Simplified Chinese equivalent**

The Simplified-Traditional Chinese conversion plugin can be flexibly integrated into different stages of text analysis. The following four usage modes are supported:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-98 Usage modes of the Simplified-Traditional Chinese conversion plugin

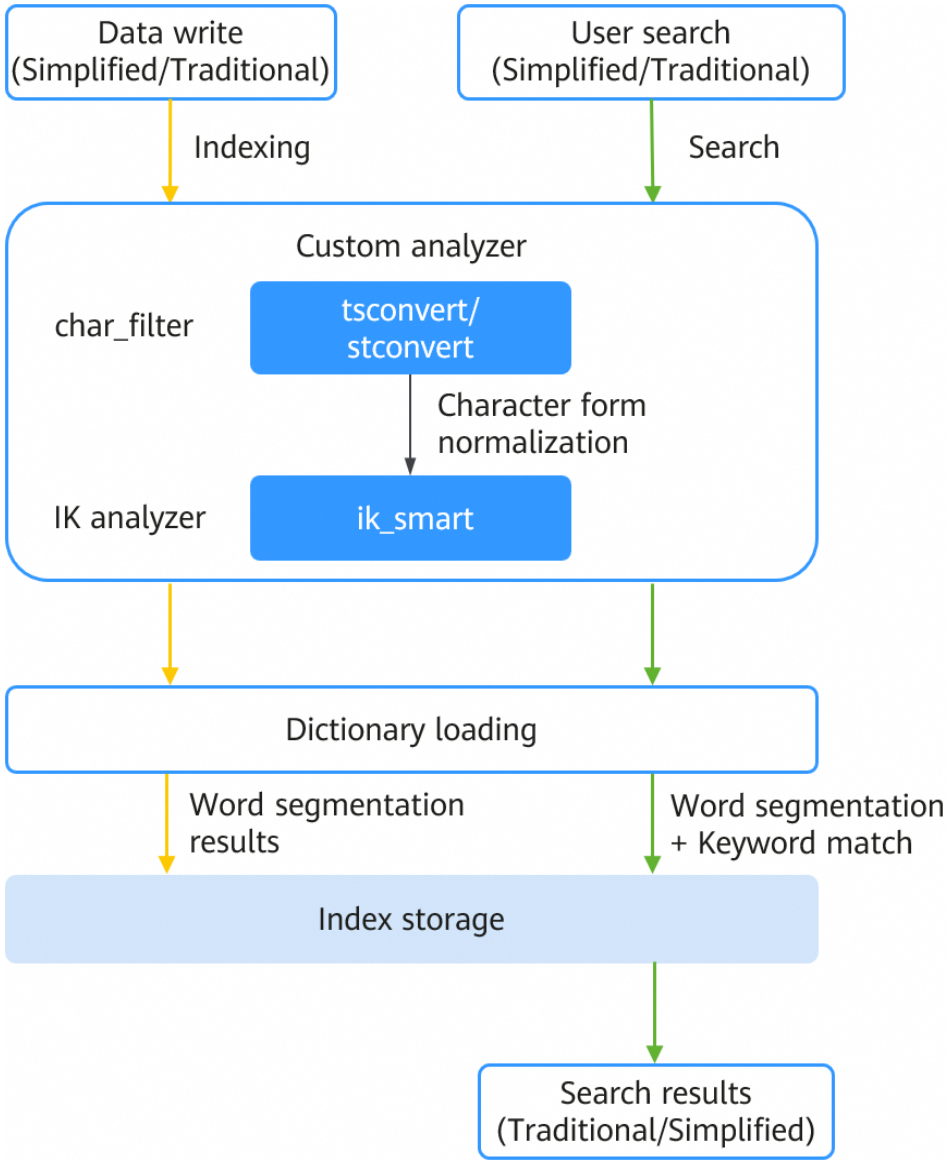
Usage	Description	Applicable Scenario
char_filter	Converts the original string before tokenization. • tsconvert: Traditional Chinese to Simplified Chinese (t2s) • stconvert: Simplified Chinese to Traditional Chinese (s2t)	Recommended mode. Used when character forms need to be unified before tokenization. Requires a tokenizer.
tokenizer	Combines conversion and tokenization into a single step.	Lightweight scenarios where no additional tokenizer is required.
token_filter	Converts tokens after tokenization.	Used when text must be tokenized first before being normalized to the same character form. Requires a tokenizer.
analyzer	Uses the plugin directly as a complete analyzer.	Only Simplified-Traditional Chinese conversion is required. There is no need for Chinese text analysis capabilities such as IK.

How Simplified-Traditional Chinese Conversion Works

This topic uses the char_filter (tsconvert) mode to integrate Simplified-Traditional Chinese conversion before the IK analyzer, enabling character form normalization before word segmentation.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-27 Solution architecture



The key workflow is as follows:

- Indexing phase: Original text is first processed by the char_filter (tsconvert) to convert Traditional Chinese characters into Simplified Chinese characters. The converted text is then analyzed by the IK analyzer (ik_smart), and the results are stored in the inverted index.
- Query phase: Search keywords undergo the same character normalization process before segmentation. The resulting terms are matched against the inverted index, enabling cross-form Simplified-Traditional Chinese search.
- Custom analyzer: A custom analyzer is created by combining the ik_smart analyzer with the tsconvert character filter. The same analyzer is used for both indexing and query processing.

Prerequisites

- The target cluster is in the **Available** state and has no ongoing tasks.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- If the custom analyzer depends on custom dictionaries, configure the custom dictionaries in advance. For details, see **Configuring Custom Dictionaries**.

Logging In to OpenSearch Dashboards

Log in to OpenSearch Dashboards and go to the command execution page. OpenSearch clusters support multiple access methods. This topic uses OpenSearch Dashboards as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Creating an Index

Create an index and define a custom analyzer with Simplified-Traditional Chinese conversion logic. The analyzer normalizes character forms before word segmentation during both indexing and query processing.

Run the following command to create the **test** index and configure the custom analyzer **ts_ik** (which combines **ik_smart** and **tsconvert**).

```
PUT /test
{
  "settings": {
    "number_of_shards": 1,
    "number_of_replicas": 0,
    "analysis": {
      "analyzer": {
        "ts_ik": {
          "tokenizer": "ik_smart",
          "char_filter": [
            "tsconvert"
          ]
        }
      },
      "char_filter": {
        "tsconvert": {
          "type": "stconvert",
          "convert_type": "t2s"
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "desc": {
        "type": "text",
        "analyzer": "ts_ik"
      }
    }
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-99 Key parameters

Parameter	Description
ts_ik	A custom analyzer that first uses the **tsconvert** character filter to convert Traditional Chinese characters into Simplified Chinese characters, and then uses **ik_smart** for word segmentation.
char_filter.tsconvert	A custom character filter based on the **stconvert** conversion plugin. The **convert_type: t2s** parameter specifies Traditional-to-Simplified Chinese conversion.
convert_type: t2s	Converts Traditional Chinese into Simplified Chinese characters before word segmentation. Because the IK analyzer provides better segmentation results for Simplified Chinese, you are advised to convert Traditional Chinese to Simplified Chinese.
analyzer: ts_ik	Associates the **desc** field with the custom analyzer. The analyzer is used for both indexing and query processing.

Expected result:

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

Verifying Character-Form Conversion and Text Analysis

After creating the index, verify the custom analyzer to ensure that Simplified-Traditional Chinese conversion and word segmentation are configured correctly before importing data.

Run the following command to check how the ts_ik custom analyzer converts and segments Traditional Chinese text.

```
POST /test/_analyze
{
  "analyzer": "ts_ik",
  "text": "Text in traditional Chinese"
}
```

Expected result:

```
{
  "tokens" : [
    { "token": "Simplified Chinese term 1", "start_offset": 0, "end_offset": 2, "type": "CN_WORD", "position": 0 },
    { "token" : "Simplified Chinese term 1", "start_offset" : 2, "end_offset" : 5, "type" : "CN_WORD", "position" : 1 }
  ]
}
```

Result interpretation: The Traditional Chinese text is converted into Simplified Chinese text first. The ik_smart analyzer then segments the converted text into terms. Both conversion and tokenization are working as expected.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Importing Data

Import both Traditional and Simplified Chinese content into the index to verify cross-form search with mixed Simplified and Traditional Chinese data.

Run the following command to write test data to the **test** index:

```
POST /test/_bulk
{"index":{"_id":"1"}}
{"desc":"Traditional Chinese sample text"}
{"index":{"_id":"2"}}
{"desc":"Simplified Chinese sample text"}
{"index":{"_id":"3"}}
{"desc":"Unrelated text"}
```

Expected result:

```
{
  "took" : 6,
  "errors" : false,
  "items" : [
    { "index" : { "_index" : "test", "_type" : "_doc", "_id" : "1", "_version" : 1, "result" : "created", "_shards" :
{ "total" : 1, "successful" : 1, "failed" : 0 }, "_seq_no" : 0, "_primary_term" : 1, "status" : 201 } },
    { "index" : { "_index" : "test", "_type" : "_doc", "_id" : "2", "_version" : 1, "result" : "created", "_shards" :
{ "total" : 1, "successful" : 1, "failed" : 0 }, "_seq_no" : 1, "_primary_term" : 1, "status" : 201 } },
    { "index" : { "_index" : "test", "_type" : "_doc", "_id" : "3", "_version" : 1, "result" : "created", "_shards" :
{ "total" : 1, "successful" : 1, "failed" : 0 }, "_seq_no" : 2, "_primary_term" : 1, "status" : 201 } }
  ]
}
```

Verifying Simplified-Traditional Chinese Conversion Search

Search using both Simplified Chinese and Traditional Chinese keywords to verify cross-form search.

Run the following command to search using a Simplified Chinese keyword.

```
GET /test/_search
{
  "query": {
    "match": {
      "desc": "Simplified Chinese keyword"
    }
  }
}
```

Run the following command to search using the corresponding Traditional Chinese keyword.

```
GET /test/_search
{
  "query": {
    "match": {
      "desc": "Traditional Chinese keyword"
    }
  }
}
```

The results of the two searches are expected to be the same:

```
{
...
  "hits" : {
    "total" : {
      "value" : 2,
      "relation" : "eq"
    },
    "max_score" : 0.43445712,
    "hits" : [
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
    {
      "_index" : "test",
      "_type" : "_doc",
      "_id" : "1",
      "_score" : 0.43445712,
      "_source" : {
        "desc": "Traditional Chinese document"
      }
    },
    {
      "_index" : "test",
      "_type" : "_doc",
      "_id" : "2",
      "_score" : 0.43445712,
      "_source" : {
        "desc" : "Simplified Chinese document"
      }
    }
   ]
  }
}
```

Result interpretation: Whether you search using the Simplified Chinese keyword or its Traditional Chinese equivalent, both Traditional Chinese and Simplified Chinese documents containing the keyword are returned. This indicates that bidirectional Simplified-Traditional Chinese search is working. Documents that do not contain the keyword are not returned.

FAQ

- **Question 1: How do I configure Simplified Chinese search to match Traditional Chinese documents?**

  When creating an index, replace the tsconvert (t2s) character filter with the stconvert (s2t) character filter. In this configuration, Simplified Chinese content is converted to Traditional Chinese during indexing, and user queries are also converted to Traditional Chinese during query processing. Configure the analyzer as follows:

  ```
  PUT /test1
  {
    "settings": {
      "number_of_shards": 1,
      "number_of_replicas": 0,
      "analysis": {
        "analyzer": {
          "ts_ik": {
            "tokenizer": "ik_smart",
            "char_filter": [
              "stconvert"
            ]
          }
        },
        "char_filter": {
          "stconvert": {
            "type": "stconvert",
            "convert_type": "s2t"
          }
        }
      }
    },
    "mappings": {
      "properties": {
        "desc": {
          "type": "text",
          "analyzer": "ts_ik"
        }
  ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  }
 }
}
```

- **Question 2: Do existing indexes need to be rebuilt to support Simplified-Traditional Chinese conversion?**

  Analyzer configurations can only be specified when an index is created and cannot be modified for existing indexes. Whether an index needs to be rebuilt depends on the existing data:

  - If all existing data is in Simplified Chinese, no index rebuild is required after adding a Traditional-to-Simplified Chinese analyzer. Existing data is already in Simplified Chinese, and newly ingested Traditional Chinese data will be automatically converted.
  - If existing data contains Traditional Chinese content, the index must be rebuilt after adding a Traditional-to-Simplified Chinese analyzer. Otherwise, historical Traditional Chinese data cannot be retrieved by Simplified Chinese keywords.

  The following two methods can be used to rebuild an index:

  - Method 1: Update in place (where there is only a small amount of data)

    Run the following command:

    ```
    POST /{my_index}/_update_by_query
    ```

    This command re-analyzes existing documents but does not modify the analyzer configuration of the index. This method works only if the index was originally created with the Simplified-Traditional Chinese conversion analyzer.

  - Method 2: Switch indexes seamlessly using an alias

    i. Create a new index configured with the Simplified-Traditional Chinese conversion analyzer.

    ii. Migrate data from the old index to the new index using **_reindex**.

    ```
    POST /_reindex
    {
      "source": { "index": "old_index" },
      "dest": { "index": "new_index" }
    }
    ```

    iii. Point the alias to the new index.

    ```
    POST /_aliases
    {
      "actions": [
        {
          "remove": {
            "index": "old_index",
            "alias": "my_alias"
          }
        },
        {
          "add": {
            "index": "new_index",
            "alias": "my_alias"
          }
        }
      ]
    }
    ```

    iv. (Optional) Delete the old index to release storage space.

    ```
    DELETE /old_index
    ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3.10.5 Implementing OpenSearch AI Search with a Search Foundation Model

By integrating the embedding capabilities of a search foundation model, OpenSearch can automatically generate embeddings when documents are indexed and perform semantic or hybrid search at query time, combining exact matching with semantic retrieval.

Application Scenarios

Traditional keyword search (BM25) relies on term matching. It cannot understand synonyms, related terms, or semantic relationships. As a result, it may fail to retrieve relevant documents when users phrase queries differently. For example, a user may ask "What should I do if I don't have permission to update a file?", while the relevant document may describe the issue as "the operation is restricted" or "the modification was rejected." Keyword search may fail to match these different expressions.

By integrating the embedding capabilities of a search foundation model, OpenSearch can automatically convert text into vector embeddings when documents are indexed. At query time, it uses semantic search (the neural query) to convert the query text into an embedding and retrieve semantically similar documents, enabling intent-aware search and improving precision and recall. Furthermore, hybrid search combines keyword search with semantic search, delivering more comprehensive search results by combining exact matching with semantic understanding.

OpenSearch AI Search is useful for semantic search use cases such as Retrieval-Augmented Generation (RAG), enterprise document search, e-commerce search, and intelligent question answering.

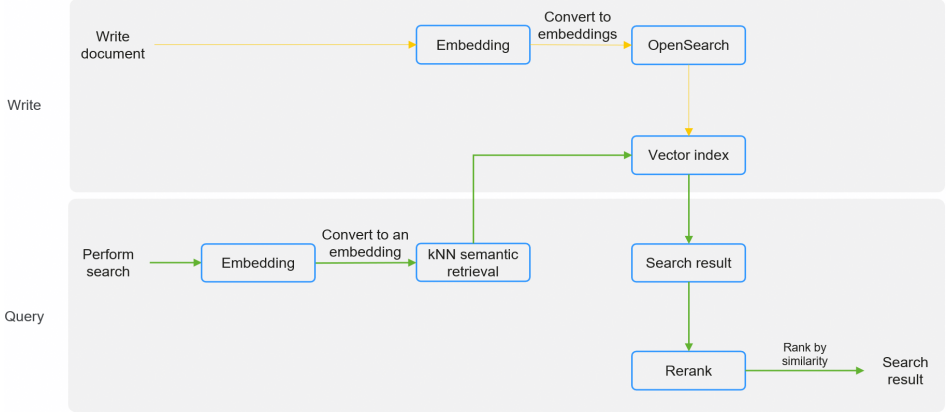
Solution Architecture

The core components of OpenSearch AI search are as follows:

- **Search foundation model**: Provides embedding and rerank model services. Three sources are supported: a CSS dedicated search foundation model cluster (under Search Model > Exclusive Clusters, restricted availability), a MaaS service, or an external model service. The model can be accessed through either a private or public network address.
- **ML Commons plugin**: The built-in machine learning plugin in OpenSearch. It can be used to register external models, create connectors, and manage model groups.
- **Connector**: Defines how OpenSearch connects to an external model service, including the access protocol, input/output formats, and authentication.
- **Ingest pipeline**: Automatically invokes the embedding model when documents are indexed and converts text fields into vector embeddings.
- **Semantic search (neural search)**: Automatically invokes the embedding model at query time to convert the query text into a vector embedding and performs k-NN (k-Nearest Neighbors) semantic search.
- **Hybrid search**: Combines keyword search and semantic search, using a search pipeline to normalize and combine scores and leverage the strengths of both retrieval methods.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-28 Solution architecture



The overall procedure is as follows:

1. Prepare the search foundation model and obtain the model service endpoint.
2. Prepare the OpenSearch cluster and complete the initial configuration.
3. Register a model group, create a connector, and register and deploy an embedding model.
4. Create an ingest pipeline and associate it with the embedding model to automatically generate embeddings.
5. Create an index containing text and vector fields and associate it with the ingest pipeline.
6. Index documents. The ingest pipeline automatically generates embeddings.
7. Perform semantic search (neural query) and hybrid search (hybrid query).

If the search model is deployed on a private network, we recommend deploying it in the same VPC as the OpenSearch cluster and accessing it through a private address without exposing it to the public network. If the model service is accessed through a public network, use HTTPS and an authentication key to secure communications. Configure the authentication key in the connector's **credential** field so that only authorized OpenSearch clusters can invoke the model service.

Advantages

- Semantic understanding: Traditional keyword search cannot understand synonyms or semantic relationships. An embedding model converts text into embeddings, enabling intent-aware search and overcoming the limitations of keyword matching.
- Automatic embedding generation: Traditional vector search requires embeddings to be generated and stored in advance. By associating an ingest pipeline with an embedding model, OpenSearch can automatically generate embeddings when documents are indexed, simplifying data processing.
- Dual-path retrieval: Pure semantic search may miss documents containing exact keywords, while pure keyword search lacks semantic understanding. Hybrid search combines both retrieval methods to cover more relevant documents and improve recall.
- Flexible weighting: The **weights** parameter adjusts the relative contribution of keyword and semantic search to meet different search requirements.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- End-to-end integration: Traditional solutions often require middleware to connect different components and coordinate the model invocation and search processes. This solution uses OpenSearch's native capabilities throughout the process, from model registration and data ingestion to retrieval, reducing integration complexity.

Constraints

- The OpenSearch cluster version must be 3.4.0.
- Each query clause in a hybrid query is executed independently. We recommend using no more than five query clauses to avoid additional performance overhead.
- The dimension of the vector field must match the output dimension of the embedding model.
- The connector's **pre_process_function** and **post_process_function** must be correctly set. Otherwise, the connector cannot be integrated with the ingest or search pipeline. These functions do not affect calls to the **_predict** API.
- Invoking an embedding model when indexing documents increases write latency. Depending on the model response time, the additional latency is approximately 50–200 ms per document. For batch indexing, control the batch size to prevent a backlog in the ingest pipeline.
- Subqueries in a hybrid query are executed independently. Adding more subqueries increases latency. Determine the appropriate value of *k* and number of subqueries through testing in your production environment.

Resource and Cost Planning

Table 3-100 Resources and costs

Resource	Description	Quantity	Billing
Search foundation model	Provides embedding and rerank model services. An exclusive CSS cluster can be accessed through a private network. MaaS or an external model service can be accessed through a private or public network, depending on the deployment.	1	Pay-per-use
OpenSearch cluster	OpenSearch 3.4.0 cluster running ML Commons and ingest/search pipelines for data indexing and search.	1	Pay-per-use

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Operation Procedure

Table 3-101 Procedure

Step	Description
Step 1: Create a Model Group	Creates a logical group for the search foundation model. This step is required only when fine-grained access control is required for models in a cluster with security mode enabled.
Step 2: Create a Model Connector	Defines how OpenSearch connects to an external model service.
Step 3: Register and Deploy a Model	Registers a remote model through the connector and deploys it.
Step 4: Test Model Availability	Uses **_predict** to verify that the model is available.
Step 5: Create an Ingest Pipeline	Creates an ingest pipeline and configures it to automatically invoke the embedding model to generate embeddings when documents are indexed.
Step 6: Create a Vector Index	Creates an index that contains a vector field named **knn_vector** and a text field.
Step 7: Index Documents	The ingest pipeline automatically generates embeddings when documents are indexed.
Step 8: Perform Semantic Search	Uses the neural query to verify semantic search.
Step 9: Create a Search Pipeline	Creates a search pipeline to normalize, weight, and combine scores from different query clauses. This step is required only for hybrid search.
Step 10: Perform Hybrid Search	Uses the hybrid query to combine keyword and semantic search.

Prerequisites

- You have prepared a search foundation model. The following three sources are supported:
  - An exclusive cluster under **Search Model** on the CSS management console: Set **Model Version** to **text-embedding-rank**. The exclusive cluster provides both embedding and rerank models.

    Access to exclusive clusters for search foundation models is controlled. To use it, submit a .
  - MaaS: Subscribe to an embedding model and rerank model through the MaaS service and obtain the API endpoint and authentication information.
  - External model service: Deploy an embedding model and rerank model yourself. For example, you can deploy an open-source model using vLLM.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

The model service must provide an OpenAI-compatible **/v1/embeddings** API.

Regardless of the deployment method, you must record the model service address, port number, model ID (or name), and authentication key. They will be needed when you configure the connector.

- An OpenSearch 3.4.0 cluster has been created, and the cluster status is **Available**.
- If the model service is accessed through a private network, deploy the model service and OpenSearch cluster in the same VPC. Configure inbound security group rules to allow traffic to the model service port and port 9200. The default model service port is 18088 for an exclusive search model cluster. For MaaS or an external model service, use the actual service port. Port 9200 is used for OpenSearch cluster communication. If the model service is accessed through a public network, make sure the OpenSearch cluster have access to the public network.

Logging In to OpenSearch Dashboards

Log in to OpenSearch Dashboards and go to the command execution page. OpenSearch clusters support multiple access methods. This topic uses OpenSearch Dashboards as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Step 1: Create a Model Group

A model group usually contains multiple models of the same type, or multiple versions of the same model. Backend roles can be granted access to a model group using the Security plugin for fine-grained permission control.

- For a non-security mode cluster (**Security Mode** is **Disabled**), no access control is required. You can skip this step.
- For a security-mode cluster (**Security Mode** is **Enabled**), creating a model group is optional. If no model group is created, model access is controlled using the default permissions defined by the ML Commons plugin. If a model group is specified, backend roles can be used to implement fine-grained access control.

Run the following command to create an embedding model group:

```
POST /_plugins/_ml/model_groups/_register
{
  "name": "embedding_model_group",
  "description": "A model group for external embedding models"
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-102 Model group parameters

Parameter	Description	Example Value
name	Custom name of the model group.	embedding_model_group
description	Description of the model group.	A model group for external embedding models

Example response:

```
{
  "model_group_id": "DHCWnZgBEIau0nVrgXDm",
  "status": "CREATED"
}
```

Record the returned **model_group_id**. It will be needed for model registration later.

Step 2: Create a Model Connector

A connector defines how OpenSearch connects to an external model service, including the communication protocol, input/output format, and authentication.

- Run the following command to create an embedding connector:

```
POST /_plugins/_ml/connectors/_create
{
  "name": "embedding connector",
  "description": "The connector to public embedding models",
  "version": 1,
  "protocol": "http",
  "parameters": {
    "endpoint": "<search model address>:<port>,
    "model": "query2doc"
  },
  "credential": {
    "api_key": "abc"
  },
  "actions": [
    {
      "action_type": "predict",
      "method": "POST",
      "url": "http://${parameters.endpoint}/v1/embeddings",
      "headers": {
        "Content-Type": "application/json"
      },
      "request_body": "{ \"input\": ${parameters.input}, \"model\": \"${parameters.model}\" }",
      "pre_process_function": "connector.pre_process.openai.embedding",
      "post_process_function": "connector.post_process.openai.embedding"
    }
  ]
}
```

- Run the following command to create a rerank connector:

```
POST /_plugins/_ml/connectors/_create
{
  "name": "pangu rerank connector",
  "description": "The connector to public rerank models",
  "version": 1,
  "protocol": "http",
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
"parameters": {
  "endpoint": "<search model address>:<port>,
  "model": "rerank",
  "response_filter": "$.results"
},
"credential": {
  "api_key": "abc"
},
"actions": [
  {
    "action_type": "predict",
    "method": "POST",
    "url": "http://${parameters.endpoint}/v1/rerank",
    "headers": {
      "Content-Type": "application/json"
    },
    "request_body": "{ \"documents\":${parameters.documents}, \"query\": \"${parameters.query}\",
\"model\": \"${parameters.model}\", \"top_n\": ${parameters.top_n}}",
    "pre_process_function": "connector.pre_process.default.rerank",
    "post_process_function": "connector.post_process.default.rerank"
  }
 ]
}
```

Table 3-103 Connector parameters

Parameter	Description	Example Value (Embedding)
name	Custom name of the connector.	pangu embedding connector
description	Description of the connector.	The connector to public embedding models
protocol	Communication protocol used by the connector to access the model service. The value can be **http** or **https**, depending on the protocol supported by the model service.	http
parameters.endpoint	Address and port number for accessing the model service, which are obtained from the model service. For an exclusive search model cluster, the default port is 18088. For MaaS or an external model service, the port varies.	10.10.1.117:18088
parameters.model	Model identifier. For an exclusive search model cluster, use the model version selected during cluster creation. For a MaaS service, use the name of the subscribed model. For an external model service, use the name specified during deployment.	query2doc

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description	Example Value (Embedding)
credential.api_key	Authentication key for connecting to the model service. Required when the model service requires authentication, such as MaaS. Omit this parameter if authentication is not required, for example, an exclusive search model cluster connected via a private network.	abc
actions[0].url	Request URL of the model service. The URL references the endpoint defined in **parameters.endpoint**.	http://${parameters.endpoint}/v1/embeddings
actions[0].pre_process_function	Preprocessing function, which converts input data into the format required by the model.	connector.pre_process.openai.embedding
actions[0].post_process_function	Postprocessing function, which converts model responses into the format required by OpenSearch.	connector.post_process.openai.embedding

CAUTION

- **pre_process_function** and **post_process_function** must be configured correctly. Otherwise, mismatched input and output formats may cause errors when the connector is integrated with the ingest or search pipeline. For configuration details, see **Connector blueprints - OpenSearch Documentation**.
- OpenSearch provides built-in OpenAI-compatible preprocessing and postprocessing functions when connecting to embedding and rerank models. If the model service uses a different API format, you can use Painless scripts to customize the processing functions.

Example response:

```
{
  "connector_id": "23CmnZgBEIau0nVrGHCp"
}
```

Record the returned **connector_id**. It will be needed for model registration later.

Step 3: Register and Deploy a Model

Use the connector to register a remote model and deploy it immediately so that it can be invoked by the ingest and search pipelines.

- Run the following command to register and deploy an embedding model:

```
POST /_plugins/_ml/models/_register?deploy=true
{
  "name": "embedding model",
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  "function_name": "remote",
  "model_group_id": "<model group ID>",
  "description": "embedding model",
  "connector_id": "<connector ID>"
}
```

- Run the following command to register and deploy a rerank model:

```
POST /_plugins/_ml/models/_register?deploy=true
{
  "name": "rerank model",
  "function_name": "remote",
  "model_group_id": "<model group ID>",
  "description": "rerank model",
  "connector_id": "<connector ID>"
}
```

Table 3-104 Model registration parameters

Parameter	Description	Example Value (Embedding)
deploy	URL parameter, which specifies whether to deploy the model immediately after registration.	true
name	Custom display name of the model in OpenSearch.	embedding model
function_name	Fixed to **remote** when accessing an external model service through a connector.	remote
model_group_id	Model group ID, that is, **model_group_id** returned in **Step 1: Create a Model Group**. This parameter is optional depending on the cluster's security mode setting. For a security-mode cluster, this parameter allows fine-grained model access control. Skip this parameter if security mode is disabled or if no model group is created.	DHCWnZgBEIau0nVrgXDm
description	Description of the registered model.	embedding model
connector_id	Connector ID, that is, **connector_id** returned in **Step 2: Create a Model Connector**.	23CmnZgBEIau0nVrGHCp

Example response:

```
{
  "task_id": "cHCxnZgBEIau0nVrb3G_",
  "status": "CREATED",
  "model_id": "aEJ7P58BSQOU_3XGorlP"
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

> **NOTE**
>
> Model deployment is an asynchronous operation. After **task_id** is returned, wait for approximately 1–10 seconds. Then, run **GET /_plugins/_ml/tasks/<task_id>** to check the deployment task status. If the value of **state** is **COMPLETED**, the deployment is complete.

Record the returned **model_id**. It will be needed when creating an ingest pipeline and performing searches later.

Step 4: Test Model Availability

Use the **_predict** API to verify that the model is correctly configured, that is, whether it can be reached and whether the output format is as expected.

- Run the following command to test the availability of the embedding model:

```
POST /_plugins/_ml/models/<model_id>/_predict
{
  "parameters": {
    "input": ["sun", "sunny"]
  }
}
```

Table 3-105 Model test parameters

Parameter	Description	Example Value
model_id	Model ID, that is, **model_id** of the embedding model returned in **Step 3: Register and Deploy a Model**.	aEJ7P58BSQOU_3XGorlP

Example response (partial):

```
{
  "inference_results": [
    {
      "output": [
        {
          "name": "response",
          "dataAsMap": {
            "embeddings": [
              {
                "values": [0.0123, -0.0456, 0.0789, ...]
              }
            ]
          }
        }
      ]
    }
  ]
}
```

Result verification: If embeddings in the form of a floating-point number array are returned, the model is correctly configured. You can proceed to the next step. If an error occurs, check the model service **endpoint** in the connector configuration, and whether inbound security group rules allows the model service port as well as port 9200.

- Run the following command to test the availability of the rerank model:

```
POST /_plugins/_ml/models/<model_id>/_predict
{
  "parameters": {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
    "query": "Is elementary school math difficult?",
    "top_n": 4,
    "documents": [
      "Elementary school math mainly covers addition, subtraction, multiplication, and division. It is generally not very difficult and provides a foundation for later study.",
      "Middle school math introduces algebra and geometry, and becomes gradually more difficult.",
      "University-level math covers subjects such as calculus and linear algebra and is considerably more difficult.",
      "Elementary school Chinese mainly covers Pinyin and basic Chinese characters."
    ]
  }
}
```

Example response:

```
{
  "inference_results": [
    {
      "output": [
        {
          "name": "similarity",
          "data_type": "FLOAT32",
          "shape": [
            1
          ],
          "data": [
            0.9980276
          ]
        },
        {
          "name": "similarity",
          "data_type": "FLOAT32",
          "shape": [
            1
          ],
          "data": [
            0.5402266
          ]
        },
        {
          "name": "similarity",
          "data_type": "FLOAT32",
          "shape": [
            1
          ],
          "data": [
            0.03015741
          ]
        },
        {
          "name": "similarity",
          "data_type": "FLOAT32",
          "shape": [
            1
          ],
          "data": [
            0.01433703
          ]
        }
      ],
      "status_code": 200
    }
  ]
}
```

Result verification: A successful response indicates that the rerank model is correctly configured and you can proceed to the next step.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Step 5: Create an Ingest Pipeline

Create an ingest pipeline that automatically invokes the embedding model when documents are indexed.

Run the following command to create an ingest pipeline named **nlp-ingest-pipeline** for the embedding model:

```
PUT /_ingest/pipeline/nlp-ingest-pipeline
{
  "description": "A text embedding pipeline",
  "processors": [
    {
      "text_embedding": {
        "model_id": "<Embedding model ID>"
        "field_map": {
          "passage_text": "passage_embedding"
        }
      }
    }
  ]
}
```

Table 3-106 Ingest pipeline parameters

Parameter	Description	Example Value
nlp-ingest-pipeline	Custom name of the ingest pipeline.	nlp-ingest-pipeline
description	Description of the ingest pipeline.	A text embedding pipeline
text_embedding	Fixed processor name of the OpenSearch ingest pipeline. At index time, the pipeline automatically runs the **text_embedding** processor to convert the text in the **passage_text** field into an embedding and stores the result in the **passage_embedding** field.	-
model_id	Embedding model ID, that is, **model_id** of the embedding model returned in **Step 3: Register and Deploy a Model**.	aEJ7P58BSQOU_3XGorlP
field_map	Maps the input text field to the output vector field. In this example, **passage_text** is converted into an embedding stored in **passage_embedding**.	{"passage_text": "passage_embedding"}

Result verification: If the result contains **"acknowledged": true**, the execution is successful.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Step 6: Create a Vector Index

Create an index containing both vector and text fields, and associate the created ingest pipeline with it as the default pipeline.

Run the following command to create a vector index named **my-nlp-index**:

```
PUT /my-nlp-index
{
  "settings": {
    "index.knn": true,
    "default_pipeline": "nlp-ingest-pipeline"
  },
  "mappings": {
    "properties": {
      "passage_embedding": {
        "type": "knn_vector",
        "dimension": 768,
        "method": {
          "engine": "faiss",
          "space_type": "innerproduct",
          "name": "hnsw"
        }
      },
      "passage_text": {
        "type": "text"
      }
    }
  }
}
```

Table 3-107 Parameters for configuring a vector index

Parameter	Description	Example Value
my-nlp-index	Custom index name.	my-nlp-index
index.knn	Whether to enable k-NN search. The value is fixed to **true**.	true
default_pipeline	Associates the index with the ingest pipeline created in **Step 5: Create an Ingest Pipeline**. At index time, the ingest pipeline automatically invokes the embedding model to convert text into embeddings.	nlp-ingest-pipeline
passage_embedding.type	Vector field type. The value is fixed to **knn_vector**.	knn_vector
passage_embedding.dimension	Vector dimension. This value must match the output dimension of the embedding model.	768
passage_embedding.method.engine	Vector search engine. Select **faiss** or **lucene** based on the model and service requirements.	faiss

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description	Example Value
passage_embedding.method.space_type	Distance measurement method. Select **l2** (Euclidean distance) or **innerproduct** (inner product) based on the model and service requirements. When **innerproduct** is used, the vectors must be normalized first.	innerproduct
passage_embedding.method.name	Index algorithm name. The value is fixed to **hnsw**.	hnsw
passage_text.type	Text field type. The value is fixed to **text**.	text

Result verification: If the result contains **"acknowledged": true**, the execution is successful.

Step 7: Index Documents

Write documents to the vector index. Because the index has been associated with the ingest pipeline, you only need to provide the passage_text field. The ingest pipeline will automatically convert text into embeddings.

- Index a single document:
```
POST /my-nlp-index/_doc/1
{
  "passage_text": "The system returned error 403: The user does not have administrator privileges. Check the policy permissions configured for the IAM role."
}
```
- Index multiple documents in a single batch:
```
POST /my-nlp-index/_bulk
{"index": {"_id": 2}}
{"passage_text": "If you cannot save a file or your changes are rejected, check whether you have permission to edit and modify the content."}
{"index": {"_id": 3}}
{"passage_text": "The Finance Department officially released its financial report for the previous quarter. The report details the company's spending on policy subsidies and tax compliance."}
{"index": {"_id": 4}}
{"passage_text": "If the system indicates that an update could not be performed due to restrictions, contact the team leader to request permission to modify the content."}
```

After the documents are indexed, the ingest pipeline automatically invokes the **text_embedding** processor to generate an embedding for each passage_text field and stores the embedding in passage_embedding. The indexed documents contain both the original passage_text field and the generated passage_embedding vector.

Result verification:

- Run the following command. If the number of returned documents matches the number of indexed documents, indexing was successful.
```
GET /my-nlp-index/_count
```
- Run the following command to view the indexed documents and vector embeddings and check whether the ingest pipeline has taken effect.
```
GET /my-nlp-index/_search
```
  Example response (partial):

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
 "hits": {
   "total": {
     "value": 4,
     "relation": "eq"
   },
   "max_score": 1,
   "hits": [
     {
       "_index": "my-nlp-index-2",
       "_id": "4",
       "_score": 1,
       "_source": {
         "passage_embedding": [
           -0.10803223,
           -0.31982422,
           0.023406982,
           ......
         ],
         "passage_text": "If the system indicates that an update could not be performed due to restrictions, contact the team leader to request permission to modify the content."
       }
     },
     ......
   ]
 }
```

passage_text indicates original text, and **passage_embedding** indicates the generated embeddings.

Step 8: Perform Semantic Search

Use the neural query to perform semantic search. The neural query automatically invokes the embedding model to convert the query text into an embedding and then performs k-NN search.

Run the following command to perform a semantic search:

```
GET /my-nlp-index/_search
{
  "_source": {
    "excludes": ["passage_embedding"]
  },
  "query": {
    "neural": {
      "passage_embedding": {
        "query_text": "What should I do if I don't have permission to update a file?",
        "model_id": "<Embedding model ID>",
        "k": 1
      }
    }
  }
}
```

Table 3-108 Neural query parameters

Parameter	Description	Example Value
_source.excludes	Excludes the vector field from the response to make the search results easier to read.	["passage_embedding"]

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description	Example Value
neural	Performs semantic search based on vector similarity (nearest neighbor search). **passage_embedding** indicates the vector field name in the index.	-
neural.query_text	Natural-language query text. The embedding model converts the query text into a vector embedding.	What should I do if I don't have permission to update a file
neural.model_id	Embedding model ID, that is, **model_id** of the embedding model returned in **Step 3: Register and Deploy a Model**.	aEJ7P58BSQOU_3XGorlP
neural.k	The number of most relevant documents to retrieve.	1

Example response:

```
{
  "took": 21,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 2,
      "relation": "eq"
    },
    "max_score": 104.77277,
    "hits": [
      {
        "_index": "my-nlp-index",
        "_id": "4",
        "_score": 104.77277,
        "_source": {
          "passage_text": "If the system indicates that an update could not be performed due to restrictions, contact the team leader to request permission to modify the content."
        }
      },
      {
        "_index": "my-nlp-index",
        "_id": "2",
        "_score": 103.32463,
        "_source": {
          "passage_text": "If you cannot save a file or your changes are rejected, check whether you have permission to edit and modify the content."
        }
      }
    ]
  }
}
```

Result verification: If semantically relevant results are returned, the semantic search is successful.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Result analysis: As shown in the example response, documents 2 and 4 that are most relevant to the query intent are successfully retrieved. Although these documents do not use exactly the same wording as the query, semantic search can identify them based on their semantic similarity. This is something that traditional keyword search cannot do.

Step 9: Create a Search Pipeline

A search pipeline can be used to process and combine the results of multiple query clauses in a hybrid query.

Two types of processors are available for hybrid search:

- Normalization processor: Normalizes and combines scores from multiple query clauses.
- Score ranker processor: Combines and re-ranks hybrid search results using rank fusion.

For hybrid search, you can add the rerank processor to the search pipeline to process search results in two phases. The normalization processor in phase_results_processors combines the scores of multiple subqueries; and then the rerank processor in response_processors performs second-phase reranking on the combined results.

- Run the following command to create the search pipeline **my-search-pipeline** using the normalization processor:

```
PUT /_search/pipeline/my-search-pipeline
{
  "description": "Post processor for hybrid search",
  "phase_results_processors": [
    {
      "normalization-processor": {
        "normalization": {
          "technique": "min_max"
        },
        "combination": {
          "technique": "arithmetic_mean",
          "parameters": {
            "weights": [
              0.3,
              0.7
            ]
          }
        }
      }
    }
  ]
}
```

- Run the following command to create the search pipeline **my-search-pipeline** using the normalization processor and rerank processor:

```
PUT /_search/pipeline/my-search-pipeline
{
  "description": "Post processor for hybrid search",
  "phase_results_processors": [
    {
      "normalization-processor": {
        "normalization": {
          "technique": "min_max"
        },
        "combination": {
          "technique": "arithmetic_mean",
          "parameters": {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
          "weights": [
            0.3,
            0.7
          ]
        }
      }
    }
  }
 ],
 "response_processors": [
  {
    "rerank": {
      "ml_opensearch": {
        "model_id": "<Rerank model ID>"
      },
      "context": {
        "document_fields": [ "passage_text" ]
      }
    }
  }
 ]
}
```

Table 3-109 Search pipeline parameters

Parameter	Description	Example Value
my-search-pipeline	Custom name of the search pipeline.	my-search-pipeline
description	Description of the search pipeline.	Post processor for hybrid search
normalization-processor	Fixed processor name for score normalization and combination.	-
normalization.technique	Normalizes the scores of each subquery to the range [0, 1]. Value range: • **min_max**: linear mapping, which maps the maximum score to 1, and the minimum score to 0, with other scores scaled proportionally. • **l2**: L2 norm normalization, where each score is divided by the square root of the sum of squared scores.	min_max
combination.technique	Method for combining normalized scores. The value can be: • **arithmetic_mean**: Weighted arithmetic mean. • **geometric_mean**: Weighted geometric mean. • **harmonic_mean**: Weighted harmonic mean.	arithmetic_mean

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description	Example Value
combination.parameters.weights	Weights assigned to each query clause, expressed as decimal values. The number of weights must match the number of query clauses in the hybrid query, and the order must be the same. For details about how to configure this parameter, see **Q1: How do I adjust the weights in a hybrid search when creating a search pipeline?**.	[0.3, 0.7]
rerank	Fixed processor name used for second-phase reranking. Configured in response_processors, the rerank processor runs after score normalization and combination.	-
rerank.ml_opensearch	Fixed configuration block for the rerank processor.	-
rerank.ml_opensearch.model_id	Rerank model ID, that is, **model_id** of the rerank model returned in **Step 3: Register and Deploy a Model**.	bFJ7P58BSQOU_3XGorlR
rerank.context	Rerank context configuration block, which specifies the data source sent to the rerank model.	-
rerank.context.document_fields	List of document fields involved in reranking. The rerank model reads text from these fields and scores their semantic relevance with the query.	["passage_text"]

Step 10: Perform Hybrid Search

Hybrid search normalizes and combines document scores from different query clauses with configurable weights through a search pipeline, delivering more comprehensive search results. This section introduces several hybrid search options:

- **Option 1: Combine keyword search with semantic vector search** (search pipeline without a rerank processor)

  Run the following command to perform a hybrid query that combines keyword search (match) and semantic vector search (neural), and specify the newly created search pipeline using the **search_pipeline** parameter:

```
GET /my-nlp-index/_search?search_pipeline=my-search-pipeline
{
  "_source": {
    "excludes": ["passage_embedding"]
  },
  "query": {
    "hybrid": {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
    "queries": [
      {
        "match": {
          "passage_text": "What should I do if I don't have permission to update a file?"
        }
      },
      {
        "neural": {
          "passage_embedding": {
            "query_text": "What should I do if I don't have permission to update a file?",
            "model_id": "<Embedding model ID>",
            "k": 5
          }
        }
      }
    ]
  }
 }
}
```

Table 3-110 Hybrid search parameters (match + neural)

Parameter	Description	Example Value
search_pipeline	URL parameter that specifies the name of the search pipeline. The value must be the same as the name of the search pipeline specified in **Step 9: Create a Search Pipeline**. The search pipeline is used to normalize and combine hybrid search results with configurable weights. You can set a default search pipeline so that you don't have to specify this parameter for every query. For details, see **Q2: How do I set a default search pipeline?**.	my-search-pipeline
_source.excludes	Excludes the vector field from the response to make the search results easier to read.	["passage_embedding"]
hybrid.queries	An array of one or more query clauses. The scores of these clauses are normalized and combined according to the configured weights. The order of the array corresponds to the order of the **weights** array in the search pipeline.	-
match	A keyword search clause that performs full-text search based on the BM25 algorithm.	-

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description	Example Value
match.passage_text	Text field (**passage_text**) to search.	What should I do if I don't have permission to update a file
neural	Performs semantic search based on vector similarity (nearest neighbor search). **passage_embedding** indicates the vector field name in the index.	-
neural.passage_embedding.query_text	Natural-language query text. The embedding model converts the query text into a vector embedding.	What should I do if I don't have permission to update a file
neural.passage_embedding.model_id	Embedding model ID, that is, the value of **model_id** returned in **Step 3: Register and Deploy a Model**.	aEJ7P58BSQOU_3XGorlP
neural.passage_embedding.k	The number of most relevant documents to retrieve.	5

Example response:


```
{
  "took": 24,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 4,
      "relation": "eq"
    },
    "max_score": 1,
    "hits": [
      {
        "_index": "my-nlp-index",
        "_id": "4",
        "_score": 1,
        "_source": {
          "passage_text": "If the system indicates that an update could not be performed due to restrictions, contact the team leader to request permission to modify the content."
        }
      },
      {
        "_index": "my-nlp-index",
        "_id": "2",
        "_score": 0.7050488,
        "_source": {
          "passage_text": "If you cannot save a file or your changes are rejected, check whether you have permission to edit and modify the content."
        }
      },
```


 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
    {
      "_index": "my-nlp-index",
      "_id": "3",
      "_score": 0.083689965,
      "_source": {
        "passage_text": "The Finance Department officially released its financial report for the previous quarter. The report details the company's spending on policy subsidies and tax compliance."
      }
    },
    {
      "_index": "my-nlp-index",
      "_id": "1",
      "_score": 0.001,
      "_source": {
        "passage_text": "The system returned error 403: The user does not have administrator privileges. Check the policy permissions configured for the IAM role."
      }
    }
  ]
 }
}
```


Verification: If the query successfully returns results ranked by normalized combined scores and max_score is 1.0 (the maximum value after min_max normalization), the hybrid search configuration is successful.

Result analysis: When the query is **What should I do if I don't have permission to update a file**, hybrid search returns four results, as shown in **Table 3-111**.

Table 3-111 Hybrid search ranking (match + neural)

Rank	Document ID	Normalized Score	Content Summary	Matching Analysis
1	4	1.0	Document update disallowed. You should request for change and operation permissions.	Strong semantic match with the query, with some keywords matched.
2	2	0.705	Request to save or change the file rejected. Confirm your permission.	Some degree of semantic match with the query, with some keywords matched.
3	3	0.084	The financial report mentions policy subsidies.	Weak keyword matching only; not semantically relevant.
4	1	0.001	System error 403 is returned. Check the policy permissions configured for the IAM role.	Weak keyword matching only; not semantically relevant.

Compared with pure semantic search (**Step 8: Perform Semantic Search**), hybrid search has the following advantages:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

  - Retains the advantage of semantic matching: Documents 4 and 2 rank among the top two based on semantic similarity, consistent with the results of pure semantic search.
  - Provides complementary recall through keyword matching: Documents 3 and 1 are retrieved because they contain the keyword **policy**. Although their semantic relevance is low, they broaden the search coverage.
  - Ranks results effectively using normalized scores: Semantically irrelevant documents (Documents 3 and 1) receive lower scores and rank at the bottom, balancing recall and precision.

- **Option 2: Combine keyword search and exact matching** (search pipeline without a rerank processor)

  This option only uses the search pipeline to normalize and combine the scores of two text query clauses. Use it if you need to combine multiple text matching strategies without relying on semantic search.

  Run the following command to perform a hybrid query that combines keyword search (match) and exact matching (term), and specify the newly created search pipeline using the **search_pipeline** parameter.

```
GET /my-nlp-index/_search?search_pipeline=my-search-pipeline
{
  "_source": {
    "excludes": ["passage_embedding"]
  },
  "query": {
    "hybrid": {
      "queries": [
        {
          "match": {
            "passage_text": "Permission"
          }
        },
        {
          "term": {
            "passage_text": {
              "value": "policy"
            }
          }
        }
      ]
    }
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-112 Hybrid query parameters (match + term)

Parameter	Description	Example Value
search_pipeline	URL parameter that specifies the name of the search pipeline. The value must be the same as the name of the search pipeline specified in **Step 9: Create a Search Pipeline**. The search pipeline is used to normalize and combine hybrid search results with configurable weights. You can set a default search pipeline so that you don't have to specify this parameter for every query. For details, see **Q2: How do I set a default search pipeline?**.	my-search-pipeline
_source.excludes	Excludes the vector field from the response to make the search results easier to read.	["passage_embedding"]
hybrid.queries	An array of one or more query clauses. The scores of these clauses are normalized and combined according to the configured weights. The order of the array corresponds to the order of the **weights** array in the search pipeline.	-
match	A keyword search clause that performs full-text search based on the BM25 algorithm.	-
match.passage_text	Text field (**passage_text**) to search.	Permission
term	An exact matching clause that performs exact term matching on the specified field.	-
term.passage_text.value	**passage_text** is the name of the text field in the index, and **value** is the term for exact matching.	policy

Example response:

```
{
  "took": 10,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
 },
 "hits": {
  "total": {
   "value": 3,
   "relation": "eq"
  },
  "max_score": 1,
  "hits": [
   {
    "_index": "my-nlp-index",
    "_id": "1",
    "_score": 1,
    "_source": {
     "passage_text": "The system returned error 403: The user does not have administrator privileges. Check the policy permissions configured for the IAM role."
    }
   },
   {
    "_index": "my-nlp-index",
    "_id": "3",
    "_score": 0.5,
    "_source": {
     "passage_text": "The Finance Department officially released its financial report for the previous quarter. The report details the company's spending on policy subsidies and tax compliance."
    }
   },
   {
    "_index": "my-nlp-index",
    "_id": "4",
    "_score": 0.3,
    "_source": {
     "If the system indicates that an update could not be performed due to restrictions, contact the team leader to request permission to modify the content."
    }
   }
  ]
 }
}
```


Verification: If the query successfully returns results ranked by normalized combined scores and max_score is 1.0, the hybrid search configuration is successful.

Result analysis: When **permission** (match) and **policy** (term) are queried, hybrid search returns three results, as shown in the table below.

Table 3-113 Hybrid search ranking (match + term)

Rank	Document ID	Normalized Score	Content Summary	Matching Analysis
1	1	1.0	System error 403 is returned. Check the policy permissions configured for the IAM role.	**permission** is matched using the match query, and **policy** is matched using the term query.
2	3	0.5	The financial report mentions policy subsidies.	The term query matches **policy**, but the match query fails to match **permission**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Rank	Document ID	Normalized Score	Content Summary	Matching Analysis
3	4	0.3	Document update disallowed. You should request for change and operation permissions.	The match query matches **permission** (or **authorization** which has similar semantics), but the term query fails to match **policy**.

- **Option 3: Combine keyword search with semantic vector search** (search pipeline with a rerank processor)

  Run the following command to perform a hybrid query that combines keyword search (match) and semantic vector search (neural), and specify the search pipeline with a rerank processor using the **search_pipeline** parameter:

```
GET /my-nlp-index/_search?search_pipeline=my-search-pipeline
{
  "_source": {
    "excludes": ["passage_embedding"]
  },
  "query": {
    "hybrid": {
      "queries": [
        {
          "match": {
            "passage_text": "What should I do if I don't have permission to update a file?"
          }
        },
        {
          "neural": {
            "passage_embedding": {
              "query_text": "What should I do if I don't have permission to update a file?",
              "model_id": "<Embedding model ID>",
              "k": 5
            }
          }
        }
      ]
    }
  },
  "ext": {
    "rerank": {
      "query_context": {
        "query_text": "What should I do if I don't have permission to update a file?"
      }
    }
  }
}
```

Table 3-114 Description of the rerank parameter in a hybrid query

Parameter	Description	Example Value
query_context.query_text	Query value received by the rerank model. It should be the same as the value in **query**.	What should I do if I don't have permission to update a file

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Verification and result analysis: Result ranking is different from when there is no rerank processor, and the _score values also change (the rerank scores have replaced the normalized combined scores).

FAQ

- **Q1: How do I adjust the weights in a hybrid search when creating a search pipeline?**

  Adjust the weights parameter according to the desired balance between semantic and keyword search.

  - Prioritize semantic matching: Increase the weight of vector search, for example, [0.2, 0.8]. Use this setting in scenarios with rich synonyms where semantic understanding is prioritized.
  - Prioritize keyword matching: Increase the weight of keyword search, for example, [0.7, 0.3]. Use this setting in scenarios where precise term matching is prioritized.
  - Balance the two: Set the weights to [0.5, 0.5] to give equal weight to both search methods.

- **Q2: How do I set a default search pipeline?**

  You can set a default search pipeline for an index to avoid specifying the **search_pipeline** parameter for every query.

  Run the following command to set **my-search-pipeline** as the default search pipeline for the **my-nlp-index** index:

  ```
  PUT /my-nlp-index/_settings
  {
    "index.search.default_pipeline": "my-search-pipeline"
  }
  ```

  After the default search pipeline is set, queries on the index automatically use this search pipeline.

- **Q3: What should I do if the "vector dimension mismatch" error is returned during indexing?**

  Cause: The dimension parameter of the knn_vector field in the index does not match the actual output dimension of the embedding model.

  Solution: Use the _predict API to obtain the embedding output and determine its dimension. Then recreate the index and set the **dimension** parameter to the value that matches the model output dimension.

- **Q4**: **What should I do if the "input/output format mismatch" error is returned for the ingest or search pipeline?**

  Cause: The connector does not have a pre_process_function or post_process_function, resulting in a mismatch between the input/output format and the format required by the pipeline.

  Solution: Add preprocessing and postprocessing functions (connector.pre_process.openai.embedding and connector.post_process.openai.embedding) compatible with the OpenAI format to the connector configuration. If the model service uses a different API format, customize the preprocessing and postprocessing functions using Painless scripts.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Related Documents

- **OpenSearch AI Search**: Official OpenSearch community documentation for semantic search, including detailed syntax for neural and hybrid queries.
- **OpenSearch Machine Learning**: Official OpenSearch community documentation for the ML Commons plugin, including the APIs for connectors, model registration, and model deployment.
- **CSS Vector Database**: Learn about the vector index algorithms and parameter settings supported by CSS.

3.11 Core Service Enhancements

3.11.1 CSS Enhancements Over Open-Source OpenSearch

In enterprise-grade search and analytics scenarios, open-source OpenSearch, despite its powerful features, often struggles to meet demands such as searching through tens of millions of high-dimensional vectors, managing PB-scale data storage, handling out-of-memory (OOM) exceptions caused by large queries, and maintaining stability during traffic peaks. To overcome these challenges, CSS offers a range of enhancements over the open-source OpenSearch, such as decoupled storage and compute, large query isolation, and enhanced ingestion performance. These enhancements go beyond performance. They also improve production-environment stability. With simple configurations, your cluster can support high concurrency and cost-effective storage of massive datasets, allowing you to focus on your core applications.

Table 3-115 CSS enhancements over open-source OpenSearch

Category	Feature	Description	Supported Version	Details
Cost optimization	Storage-compute decoupling	With decoupled storage and compute, hot data that is frequently accessed is stored in high-performance storage media, while cold data that is infrequently accessed is migrated to low-cost storage media—Object Storage Service (OBS). This ensures real-time query performance for hot data while reducing long-term storage costs.	1.3.6, 2.19.0, 3.4.0	**Configuring Decoupled Storage and Compute**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Category	Feature	Description	Supported Version	Details
	Switching between hot and cold storage tiers	Data is allocated to nodes of different performance standards based on data temperature (that is, how often data is accessed). The goal is to achieve optimal storage costs and query performance.	All versions that support cold data nodes	**Switching Between Hot and Cold Data**
Enhanced stability	Read/write splitting	This feature relies on collaboration between the leader and follower clusters. Writes are directed to the leader and queries to the follower, ensuring optimal write performance while supporting high-concurrency, scalable queries. There is no more resource contention, and peak loads are reduced.	2.19.0, 3.4.0	**Configuring Read/Write Splitting Between Leader and Follower**
	Flow control	This feature protects clusters from overload through flow control policies, such as client request throttling, shard indexing backpressure, and traffic pattern analysis, ensuring proper resource allocation and cluster stability.	2.19.0, 3.4.0	**Configuring Flow Control**
	Query traffic isolation	You can configure specific nodes as low-priority nodes or isolated nodes and define an index whitelist for exceptions. For shard query requests to indexes that are not on this whitelist, the system preferentially or completely bypasses these nodes, ensuring that faulty nodes do not affect services.	2.19.0, 3.4.0	**Configuring Query Traffic Isolation**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Category	Feature	Description	Supported Version	Details
	Large query isolation	Large query isolation can be configured to manage queries that have high memory usage or take too long to complete. This helps improve the stability of OpenSearch clusters and prevent out-of-memory (OOM) exceptions.	2.19.0, 3.4.0	**Configuring Large Query Isolation**
	OpenSearch Dashboards query response circuit breaker	The OpenSearch Dashboards query circuit breaker estimates heap usage before buffering the response body. It automatically discards oversized responses to prevent the OpenSearch Dashboards backend process from crashing due to out-of-memory (OOM) errors.	2.19.0, 3.4.0	**Configuring the OpenSearch Dashboards Query Response Circuit Breaker**
	Automatic retry for failed shard allocation	By configuring automatic retry for failed shard allocation with an exponential backoff strategy, you can enable automatic recovery from shard allocation failures. Such failures may occur due to high node load or a triggered circuit breaker.	2.19.0, 3.4.0	**Configuring Automatic Retry for Failed Shard Allocation**
Enhanced performance	Enhanced data ingestion performance	CSS provides several ways to enhance ingestion performance: bulk routing, bulk aggregation, text indexing acceleration, and merge optimization. They can help reduce performance overhead and significantly improve write stability and throughput in heavy-load scenarios.	2.19.0, 3.4.0	**Enhancing Data Ingestion Performance**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Category	Feature	Description	Supported Version	Details
	Enhanced aggregation performance	CSS enhances aggregation performance in the face of large data volumes by leveraging vectorization and optimized clustering techniques, enabling faster analytics and decision-making in complex situations.	2.19.0, 3.4.0	**Configuring Enhanced Aggregation**
Enhanced operations	Index recycle bin	To prevent data loss caused by accidental deletion, CSS provides an index recycle bin. When it is enabled, deleted indexes are temporarily stored in the recycle bin, allowing for recovery before they are permanently removed. This feature improves data reliability and operational security.	2.19.0, 3.4.0	**Enabling the Index Recycle Bin**
	Query resource tracker	O&M personnel can call an API to obtain top queries with the highest latency, CPU usage, or memory consumption, filter the queries by time range, and quickly identify problematic queries. This can significantly improve troubleshooting efficiency and accuracy.	2.19.0, 3.4.0	**Tracking Query Resource Consumption**

3.11.2 Configuring Decoupled Storage and Compute

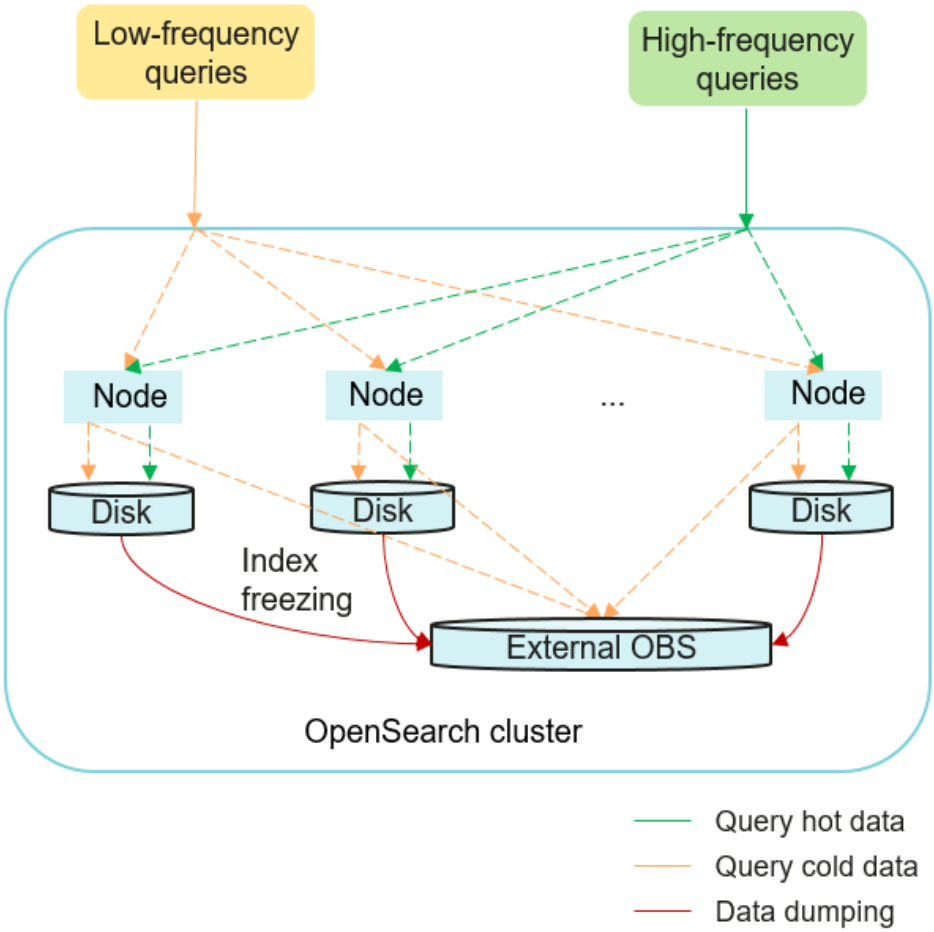
As data grows exponentially, traditional monolithic storage architectures struggle to balance performance and cost efficiency. To support high-speed real-time queries while lowering the cost of storing massive quantities of data, CSS offers the storage-compute decoupling feature for OpenSearch clusters.

With decoupled storage and compute, hot data that is frequently accessed is stored in high-performance storage media, while cold data that is infrequently accessed is migrated to low-cost storage media—Object Storage Service (OBS). This ensures real-time query performance for hot data while reducing long-term storage costs.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

How the Feature Works

Figure 3-29 Decoupled storage and compute architecture



This is how storage-compute decoupling works:

1. Data writing

   Hot data is written to high-performance local disks (SSDs) to ensure real-time query performance.

2. Hot-to-cold transition

   Infrequently accessed data is migrated from SSDs to cost-effective OBS (OBS buckets are invisible to users), turning it from hot data to cold data. You can call the index freezing API to manually turn hot data to cold data, or define a lifecycle policy (for example, a 90-day data retention period) to have it triggered automatically. Cold data itself is stored in OBS for cost efficiency, but its metadata and date fields are maintained on local disks to enable rapid search.

3. Query processing
   - Hot data is read directly from local disks with millisecond latency.
   - For cold data, the system uses metadata retained on local disks to quickly locate it in OBS, load it, and return it to the user.

In CSS, indexes may go through the following stages during their entire lifecycle.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- Hot index (open): Writes are allowed. Query latency is in milliseconds.
- Frozen index (freeze_low_cost): Rarely accessed data is stored in OBS. The index becomes read-only, and query latency is in seconds or even minutes.
- Deleted index: Frozen indexes are deleted periodically to reclaim storage resources.

Decoupled storage and compute can be automated via index lifecycle management policies. For details, see **Decoupling Storage and Compute Using ISM**.

Impact on Billing

Upon freezing, indexes are dumped to OBS, and the storage class is Standard. OBS buckets that store frozen indexes are invisible to users—users cannot view or manage them on the OBS console. Note that storing indexes in OBS buckets incurs additional fees, billed on a pay-per-use basis according to consumption of Standard storage capacity. For details, see **Cloud Search Service Price Calculator**.

Constraints

- Only OpenSearch 1.3.6, 2.19.0, and 3.4.0 clusters support decoupled storage and compute.
- While an index is being frozen, the system sets the index to the read-only state. Even after the data in the index is dumped to OBS, the index remains read-only and no data can be written into the index.
- During index freezing, the data in it can still be queried. After the freezing is complete, the index is closed and then re-opened. During this transition period, the cluster may be in the red state temporarily, and the index becomes unsearchable. The index becomes searchable again after being re-opened.
- After an index is frozen, its data is dumped to an OBS bucket, and any index data is deleted from local disks. A frozen index has an increased query latency. During an aggregation operation, the latency becomes even longer because the query is complex and a large amount of data needs to be retrieved.
- A frozen index with data already dumped to OBS cannot be unfrozen. That is, a read-only index cannot be rolled back to writable.
- In the case of two clusters configured with read/write splitting, the secondary cluster cannot set storage-compute decoupling for indexes synchronized from the primary cluster.
- Vector indexes do not support storage-compute decoupling.
- When an index is frozen, its data is dumped to an OBS bucket. This process will consume network bandwidth. Make sure data transmission between your cluster and OBS does not exhaust the maximum bandwidth and QPS supported by OBS. If these constraints are violated, the performance of queries that involve OBS will deteriorate. For example, the speed of restoring shards and querying data will be slow.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Logging In to OpenSearch Dashboards

Log in to OpenSearch Dashboards and go to the command execution page. OpenSearch clusters support multiple access methods. This topic uses OpenSearch Dashboards as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Freezing an Index

Freezing an index means to move cold data from solid state drives (SSDs) to OBS, freeing up SSD storage for hot, frequently accessed data. A frozen index becomes read-only.

1. Freeze an index.

   Run the following command to dump the data of a specified index to OBS and set its storage class to Standard:

```
POST {index_name}/_freeze_low_cost
```

Table 3-116 Parameter description

Parameter	Type	Default Value	Description
index_name	String	N/A	Specifies one or more indexes. • Single index: Enter the index name, for example, **my_index**. • Multiple indexes: Enter multiple index names and use a comma (,) to separate them, for example, **my_index1,my_index2**. • Wildcard: Use the wildcard (*) to match multiple indexes. For example, **myindex*** matches all indexes whose name starts with **myindex**.

Example response:

```
{
   "freeze_uuid": "pdsRgUtSTymVDWR_HoTGFw" // ID of the index freezing task, which you can use to query the task progress.
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

CAUTION

If the OpenSearch cluster version is 2.19.0 or 3.4.0 and the image version is no earlier than x.x.x_25.9.0_xxx, index freezing tasks are not executed immediately upon submission. Instead, they are temporarily cached for a configurable period (default: one minute) and then processed in batches. The caching rules for asynchronous index freezing tasks can be customized. For details, see **Modify batch index freezing settings**.

2. Check the progress of the index freezing task.

   Run the following command to check the index freezing progress by specifying the task ID:

   ```
   GET _freeze_low_cost_progress/{freeze_uuid}
   ```

   If the OpenSearch cluster version is 2.19.0 or 3.4.0 and the image version is no earlier than x.x.x_25.9.0_xxx, the progresses of multiple index freezing tasks of the same batch are returned.

   Example response:

   ```
   {
     "stage" : "STARTED",
     "shards_stats" : {
       "INIT" : 0,
       "FAILURE" : 0,
       "DONE" : 0,
       "STARTED" : 1,
       "ABORTED" : 0
     },
     "indices" : {
       "data1" : [
         {
           "uuid" : "7OS-G1-tRke2jHZPlckexg",
           "index" : {
             "name" : "data1",
             "index_id" : "4b5PHXJITLaS6AurImfQ9A",
             "shard" : 2
           },
           "start_ms" : 1611972010852,
           "end_ms" : -1,
           "total_time" : "10.5s",
           "total_time_in_millis" : 10505,
           "stage" : "STARTED",
           "failure" : null,
           "size" : {
             "total_bytes" : 3211446689,
             "finished_bytes" : 222491269,
             "percent" : "6.0%"
           },
           "file" : {
             "total_files" : 271,
             "finished_files" : 12,
             "percent" : "4.0%"
           },
           "rate_limit" : {
             "paused_times" : 1,
             "paused_nanos" : 946460970
           }
         }
       ]
     }
   }
   ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-117 Response parameters

Parameter	Description
stage	Status of the index freezing task. Value range: • **INIT**: The task has just started or is being initialized. • **FAILURE**: failed • **DONE**: complete • **STARTED**: started • **ABORTED**: canceled. This field is reserved.
shards_stats	Number of shards in each state.
indices	Status details of each index, as described in **Table 3-118**.

Table 3-118 indices parameters

Parameter	Description
uuid	ID of the index freezing task.
index	Index and shard information.
start_ms	Task start time.
end_ms	Task end time. If the task is still ongoing, **-1** is displayed.
total_time	How long the task has been ongoing.
total_time_in_millis	How long the task has been ongoing, in milliseconds.
stage	Shard status.
failure	Task failure cause. If no failure has occurred, **null** is returned.
size.total_bytes	Total bytes to be frozen.
size.finished_bytes	Number of bytes that have been frozen.
size.percent	Freezing progress in terms of bytes that have been frozen vs total bytes to freeze.
file.total_bytes	Total bytes to be frozen.
file.finished_bytes	Number of bytes that have been frozen.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
file.percent	Freezing progress in terms of the number of documents that have been frozen vs. total number of documents to freeze.
rate_limit.paused_times	Number of times freezing has been suspended due to rate limiting.
rate_limit.paused_nanos	Duration of freezing task suspension due to rate limiting, in nanoseconds.

When index freezing is complete, the settings parameters will be returned, as described in **Table 3-119**.

Table 3-119 settings parameters

Parameter	Description
index.frozen_low_cost	Frozen index. The value is fixed to **true**.
index.blocks.write	Writes disallowed to a frozen index. The value is fixed to **true**.
index.store.type	Index storage type. The value is fixed to **obs**.

3. Query the list of frozen indexes.

   NOTE

   Here, only some of the key parameters for querying the frozen index list are described. For more information about the API, see **CAT indices**.

   Run the following command to query the frozen index list based on index freezing status:

```
GET _cat/freeze_indices?stage={STAGE}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-120 Parameter description

Parameter	Type	Default Value	Description
STAGE	String	N/A	Index freezing stage. Value range: • **start**: Freezing has started and is ongoing. • **done**: Freezing has completed. • **unfreeze**: Freezing has not started. • Empty or any other value: Freezing is ongoing or has completed.

Example response:

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

Optimizing Cache Settings for Frozen Indexes

After an index is frozen, its data is dumped to OBS. To reduce direct data retrieval from OBS and thus improve query performance, some data is cached in the cluster. Data that is requested for the first time is directly retrieved from OBS. The retrieved data is then cached in the cluster memory. In response to subsequent queries, the system searches the cache first. CSS allows you to query cache statistics for frozen indexes that are stored in OBS buckets. You can also reset the cache status and modify cache configuration.

- Query cache statistics for frozen indexes stored in OBS buckets.
  - Run the following command to query cache statistics for frozen indexes on all nodes:
    ```
    GET _frozen_stats
    ```
  - Run the following command to query cache statistics for frozen indexes on specified nodes:
    ```
    GET _frozen_stats/{node_id}
    ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-121 Parameter description

Parameter	Type	Default Value	Description
node_id	String	N/A	Specifies one or more cluster nodes. • Single node: Enter the node ID. • Multiple nodes: Enter multiple node IDs and use a comma (,) to separate them. You can run the following command to obtain node IDs: `GET _cat/nodes?s=n&h=n,id&v=true&full_id=true`

The following is an example of the returned information for querying frozen index cache statistics on all nodes:

```
{
  "_nodes" : {
    "total" : 1,     // Total number of nodes
    "successful" : 1,      // Successful nodes
    "failed" : 0      // Failed nodes
  },
  "cluster_name" : "css-zzz1",             //Cluster name
  "nodes" : {
    "7uwKO38RRoaON37YsXhCYw" : {
      "name" : "css-zzz1-ess-esn-2-1",        //Node name
      "transport_address" : "10.0.0.247:9300",    //Node transport address
      "host" : "10.0.0.247",             //Node host
      "ip" : "10.0.0.247",             //Node IP address
      "block_cache" : {
        "default" : {
          "type" : "memory",             //Cache type. memory indicates in-memory cache.
          "block_cache_capacity" : 8192,     //Cache capacity
          "block_cache_blocksize" : 8192,     //Single block size in the cache, in bytes. In the example, the block size is 8 KB.
          "block_cache_size" : 12,          //Cache capacity used
          "block_cache_hit" : 14,          //Number of cache hits
          "block_cache_miss" : 0,          //Number of cache misses
          "block_cache_eviction" : 0,          //Number of cache evictions
          "block_cache_store_fail" : 0          //Number of cache storage failures, which occur when the cache is full.
        }
      },
      "obs_stats" : {
        "list" : {
          "obs_list_count" : 17,          //Number of times the OBS list API was called.
          "obs_list_ms" : 265,             //Total length of time spent calling the OBS list API.
          "obs_list_avg_ms" : 15          //Average time spent calling the OBS list API.
        },
        "get_meta" : {
          "obs_get_meta_count" : 79,     //Number of times the OBS get metadata API was called.
          "obs_get_meta_ms" : 183,     //Total length of time spent calling the OBS get metadata API.
          "obs_get_meta_avg_ms" : 2     //Average time spent calling the OBS get metadata API.
        },
        "get_obj" : {
          "obs_get_obj_count" : 12,     //Number of times the OBS get object API was called.
          "obs_get_obj_ms" : 123,     //Total length of time spent calling the OBS get object API.
          "obs_get_obj_avg_ms" : 10     //Average time spent calling the OBS get object API.
        },
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
      "put_obj" : {
        "obs_put_obj_count" : 12,     //Number of times the OBS put object API was called.
        "obs_put_obj_ms" : 2451,     //Total length of time spent calling the OBS put object API.
        "obs_put_obj_avg_ms" : 204     //Average time spent calling the OBS put object API.
      },
      "obs_op_total" : {
        "obs_op_total_ms" : 3022,     //Total length of time spent calling OBS APIs.
        "obs_op_total_count" : 120,     //Total number of times calling OBS APIs.
        "obs_op_avg_ms" : 25          //Average time spent calling OBS APIs.
      }
    },
    "reader_cache" : {
      "hit_count" : 0,
      "miss_count" : 1,
      "load_success_count" : 1,
      "load_exception_count" : 0,
      "total_load_time" : 291194714,
      "eviction_count" : 0
    }
  }
 }
}
```

- Reset the cache status for frozen indexes stored in OBS buckets.

  Run the following command to reset the cache status for frozen indexes:

  ```
  POST _frozen_stats/reset
  ```

  This command is used to troubleshoot performance issues. For example, if you reset the cache status before executing a query, you can see the cache hit results clearly. It is not advisable to use this command during runtime.

  Example response:

  ```
  {
   "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
   },
   "cluster_name" : "Es-0325-007_01",
   "nodes" : {
    "mqTdk2YRSPyOSXfesREFSg" : {
     "result" : "ok"
    }
   }
  }
  ```

- Modify cache configuration for frozen indexes.

  OpenSearch accesses different types of files using different methods. The cache system supports multi-level caches and uses blocks of different sizes to cache different types of files. For example, typically, a large number of small blocks are used to cache .fdx and .tip files, while a small number of large blocks are used to cache .fdt files. You can modify the cache settings based on service needs. **Table 3-122** describes the parameters that can be configured.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-122 Configuring cache settings for frozen indexes

Parameter	Type	Default Value	Description
low_cost.obs.blockcache.names	Array	["default"]	A list of the names of multi-level caches. The cache system supports multi-level caches for data of different access granularities. There is a default cache named **default**. If you configure this parameter, the value must include at least the **default** cache in addition to other custom cache names.
low_cost.obs.blockcache.<NAME>.type	ENUM	memory	Storage type of the cache named <NAME>. Value range: • **memory**: The cache uses in-memory storage, which means it consumes memory resources. • **file**: The cache uses disk storage. Its capacity is limited by available disk space. In this case, you are advised to use ultra-high I/O disks to improve cache performance.
low_cost.obs.blockcache.<NAME>.blockshift	Integer	13 (indicating 8 KB)	Block size in the cache named <NAME> (in bytes). Formula: $2^{blockshift}$. For example, if the value is 16, the block size is 2^{16} bytes, that is, 65536 bytes or 64 KB. Value: number of left-shifts
low_cost.obs.blockcache.<NAME>.bank.count	Integer	1	Number of partitions in the cache named <NAME>. Each partition is an independent unit in the cache.
low_cost.obs.blockcache.<NAME>.number.blocks.perbank	Integer	8192	Number of blocks in each partition of the cache named <NAME>.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
low_cost.obs.blockcache.<NAME>.exclude.file.types	Array	N/A	List of file name extensions for files that cannot be cached in the cache named <NAME>. If a file's extension is not in **exclude.file.types** or **<NAME>.file.types** (if configured), the default cache policy applies.
low_cost.obs.blockcache.<NAME>.file.types	Array	N/A	List of file name extensions for files that can be cached in the cache named <NAME>. If a file's extension is not in **exclude.file.types** or **<NAME>.file.types** (if configured), the default cache policy applies.
index.frozen.obs.max_bytes_per_sec	String	150MB	Maximum rate at which the documents of frozen indexes are uploaded to OBS. Value format: number + unit • Number: a natural number • Unit: B, K, KB, M, MB, G, GB, T, TB, P, or PB (case-insensitive) Value range: 0 to (2^{63} – 1) bytes The value **0** indicates no limit. The change takes effect immediately.
low_cost.obs.index.upload.threshold.use.multipart	String	1GB	Multipart upload threshold. In the process of dumping frozen indexes to OBS, if the size of a single document exceeds this threshold, a multipart upload is performed. Value format: number + unit • Number: a natural number • Unit: B, K, KB, M, MB, G, GB, T, TB, P, or PB (case-insensitive) Value range: 0 to (2^{63} – 1) bytes The value **0** indicates there is no multipart upload.

The following is a typical cache configuration example. A two-level cache system is used: **default** and **large**. The **default** cache has a total of 30 x 4096 64-KB blocks. It is used to cache non-.fdt files. The **large** cache has 5 x 1000 2-MB blocks. It is used to cache .fdx, .dvd, and .tip files.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
low_cost.obs.blockcache.names: ["default", "large"]
low_cost.obs.blockcache.default.type: file
low_cost.obs.blockcache.default.blockshift: 16
low_cost.obs.blockcache.default.number.blocks.perbank: 4096
low_cost.obs.blockcache.default.bank.count: 30
low_cost.obs.blockcache.default.exclude.file.types: ["fdt"]

low_cost.obs.blockcache.large.type: file
low_cost.obs.blockcache.large.blockshift: 21
low_cost.obs.blockcache.large.number.blocks.perbank: 1000
low_cost.obs.blockcache.large.bank.count: 5
low_cost.obs.blockcache.large.file.types: ["fdx", "dvd", "tip"]
```

- Modify batch index freezing settings.

  If the OpenSearch cluster version is 2.19.0 or 3.4.0 and the image version is no earlier than x.x.x_25.9.0_xxx, index freezing tasks are not executed immediately upon submission. Instead, they are temporarily cached for a configurable period and then processed in batches. The caching rules for asynchronous index freezing tasks can be customized. **Table 3-123** describes relevant parameters.

  Table 3-123 Parameters for configuring batch index freezing

Parameter	Type	Default Value	Description
low_cost.freeze.job.interval	TimeValue	1m	Batch index freezing task execution interval. Value range: 10s to 10m If this parameter is set to 1m, the system caches index freezing requests for 1 minute and then executes them all at once.
low_cost.freeze.job.batch_size	Integer	20	The maximum number of indexes that can be included in a single batch index freezing task. Value range: 1 to 100 If this parameter is set to 20, each batch can contain 20 indexes. For example, if an Index Lifecycle Management (ILM) task submits 100 indexes for freezing at once, they will be divided into five batches and frozen one batch at a time.
low_cost.freeze.job.start_time	String	00:00 (executable all day)	Start time for batch index freezing tasks, meaning no such tasks can be started before the specified time. If a submitted index freezing task has not yet started, it will have to wait until the specified start time. Value format: hour:minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
low_cost.freeze.job.end_time	String	23:59 (executable all day)	Start end time for batch index freezing tasks, meaning no such tasks can be started after the specified time. Ongoing tasks are not affected by this setting. Format: hour:minute

For example, run the following command to modify the default batch index freezing settings:

```
PUT _cluster/settings
{
  "persistent": {
    "low_cost.freeze.job.interval": "5m",
    "low_cost.freeze.job.batch_size": 30,
    "low_cost.freeze.job.start_time": "22:00",
    "low_cost.freeze.job.end_time": "07:00"
  }
}
```

Enhancing Query Performance for Frozen Indexes

When frozen indexes are queried on the **Discover** page of OpenSearch Dashboards for the first time, all data needs to be retrieved from OBS because there is no data in the cache. If a large number of documents need to be returned, it takes a long time to retrieve the corresponding time fields and document metadata from OBS. By caching this part of data in the cluster, query performance can be improved significantly. This is how CSS improves query performance for frozen indexes. Local cache settings are preset. You can review them and modify them as needed.

1. Modify local cache settings for frozen indexes. **Table 3-124** describes these settings. All of them can be modified dynamically, and the changes take effect immediately.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-124 Local cache settings

Parameter	Type	Default Value	Description
low_cost.local_cache.max.capacity	Integer	500	Maximum number of cold data caches that can be made available on each node. (Each shard corresponds to a cache.) Value range: 10–5000 • If the heap memory usage remains high, decrease this value. • If the value of **load_overflow_count** keeps increasing rapidly, increase this value.
index.low_cost.local_cache.threshold	Integer	50	The percentage of **date** fields in an index as a threshold for enabling or disabling the local caching of cold data. • If in an index, the percentage of **date** fields is lower than this value, cold data of the **date** type will be cached locally. Otherwise, such data will not be cached. • If date fields account for the vast majority of the total data size of the current index, you should not cache its data locally. Value range: 0–100 Unit: %
index.low_cost.local_cache.evict_time	String	30d	Retention duration of cold data in the local cache. The value is determined by **index.frozen_date** (time of index freezing). If **index.frozen_date** is unavailable, the value is determined by the index creation time. You are advised to adjust the retention duration based on your disk usage. Value range: 1 to 365 days Unit: d (days)

The following provides some examples:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- Run the following command to change the maximum number of cold data caches that can be made available on each node:

```
PUT _cluster/settings
{
  "persistent": {
    "low_cost.local_cache.max.capacity": 1000
  }
}
```

- Run the following command to change the threshold (in the form of the percentage of **date** fields in an index) for enabling or disabling the local caching of **date** fields as cold data:

```
PUT {index_name}/_settings
{
  "index.low_cost.local_cache.threshold": 20
}
```

- Run the following command to change the retention duration of cold data in the local cache:

```
PUT {index_name}/_settings
{
  "index.low_cost.local_cache.evict_time": "7d"
}
```

2. Query local cache statistics for frozen indexes.
   - Query local cache statistics for frozen indexes on all nodes:

     ```
     GET /_frozen_stats/local_cache
     ```

   - Query local cache statistics for frozen indexes on specified nodes:

     ```
     GET /_frozen_stats/local_cache/{node_id}
     ```

   Example response:

```
{
  "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  },
  "cluster_name" : "test",
  "nodes" : {
    "6by3lPy1R3m55Dcq3liK8Q" : {
      "name" : "node-1",
      "transport_address" : "127.0.0.1:9300",
      "host" : "127.0.0.1",
      "ip" : "127.0.0.1",
      "local_cache" : {
        "get_stats" : {
          "get_total_count" : 562,              //Total number of times data was retrieved from the local cold data cache.
          "get_hit_count" : 562,                //Total number of hits in the local cold data cache.
          "get_miss_count" : 0,                 //Total number of local cold data cache misses.
          "get_total_ns" : 43849200,            //Total duration for retrieving data from the local cold data cache.
          "get_avg_ns" : 78023                  //Average duration for retrieving data from the local cold data cache.
        },
        "load_stats" : {
          "load_count" : 2,                   //Number of times cold data was loaded from the local cache
          "load_total_ms" : 29,                 //Total duration for loading cold data from the local cache
          "load_avg_ms" : 14,                   //Average duration for loading cold data from the local cache
          "load_fail_count" : 0,                //Number of failures for loading cold data from the local cache
          "load_overflow_count" : 0         //Number of times the local cold data cache exceeds the cache pool size.
        },
        "reload_stats" : {
          "reload_count" : 0,                 //Number of times the local cold data cache was regenerated.
          "reload_total_ms" : 0,                //Total duration for regenerating the local cold data cache.
          "reload_avg_ms" : 0,                  //Average duration for regenerating the local cold data cache.
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        "reload_fail_count" : 0              //Number of failures in regenerating the local cold data
cache.
       },
       "init_stats" : {
         "init_count" : 0,                 //Number of times the local cold data cache was initialized.
         "init_total_ms" : 0,              //Total duration for initializing the local cold data cache.
         "init_avg_ms" : 0,                //Average duration for initializing the local cold data cache.
         "init_fail_count" : 0             //Number of failures in initializing the local cold data cache.
       }
     }
   }
  }
}
```

Querying the Real-Time Rates of OBS in Handling Cold Data

To help you understand how the storage-compute decoupling plug-in is working with OBS, an API for collecting statistics on the real-time rates of OBS has been added, and the real-time rates are recorded in the index **.freeze_obs_rate-YYYY.mm.dd**.

Calculation method: The average OBS operation rates in the last 5 seconds are calculated every 5 seconds.

The system index **.freeze_obs_rate-YYYY.mm.dd** records statistics on OBS real-time operation rates, helping you understand relevant trends about the OBS resources that store cold data. The default retention period of the index is 30 days.

1. Querying the real-time rates of OBS in handling cold data.
   - Run the following command to query the real-time OBS rates on all nodes:
     ```
     GET _frozen_stats/obs_rate
     ```
   - Run the following command to query the real-time OBS rates on specified nodes:
     ```
     GET _frozen_stats/obs_rate/{node_id}
     ```

   Example response:

```
{
  "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  },
  "cluster_name" : "test",
  "nodes" : {
    "dflDvcSwTJ-fkiIlT2zE3A" : {
      "name" : "node-1",
      "transport_address" : "127.0.0.1:9300",
      "host" : "127.0.0.1",
      "ip" : "127.0.0.1",
      "update_time" : 1671777600482,              // Time when the current statistics were updated.
      "obs_rate" : {
        "list_op_rate" : 0.0,                   // Rate of OBS list operations. Unit: times/s.
        "get_meta_op_rate" : 0.0,                 // Rate of OBS get meta operations. Unit: times/s.
        "get_obj_op_rate" : 0.0,                // Rate of OBS get operations. Unit: times/s.
        "put_op_rate" : 0.0,                    // Rate of OBS put operations. Unit: times/s.
        "obs_total_op_rate" : 0.0,              // Rate of all OBS operations. Unit: times/s.
        "obs_upload_rate" : "0.0 MB/s",           // OBS data upload rate. Unit: MB/s.
        "obs_download_rate" : "0.0 MB/s"          // OBS data download rate. Unit: MB/s.
      }
    }
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. Modify the retention period of the **.freeze_obs_rate-YYYY.mm.dd** index that stores the OBS real-time rates. The default retention period of this index is 30 days.

   Run the following command to change the index retention period to seven days:

```
PUT _cluster/settings
{
  "persistent": {
    "low_cost.obs_rate_index.evict_time": "7d"
  }
}
```

Table 3-125 Parameter description

Parameter	Type	Default Value	Description
low_cost.obs_rate_index.evict_time	String	30d	The retention period of the **.freeze_obs_rate-YYYY.mm.dd** index on each node. • Value range: 1 to 365 days • Unit: d (days) Any change takes effect immediately.

Related Documents

To ensure acceptable performance for cold data queries (for example, when historical data is queried periodically), use a tiered storage solution. For details, see **Switching Between Hot and Cold Data**.

3.11.3 Switching Between Hot and Cold Data

In an OpenSearch cluster, switching between hot and cold data storage means to allocate data to nodes of different performance standards based on data temperature (that is, how often data is accessed). The goal is to achieve optimal storage costs and query performance.

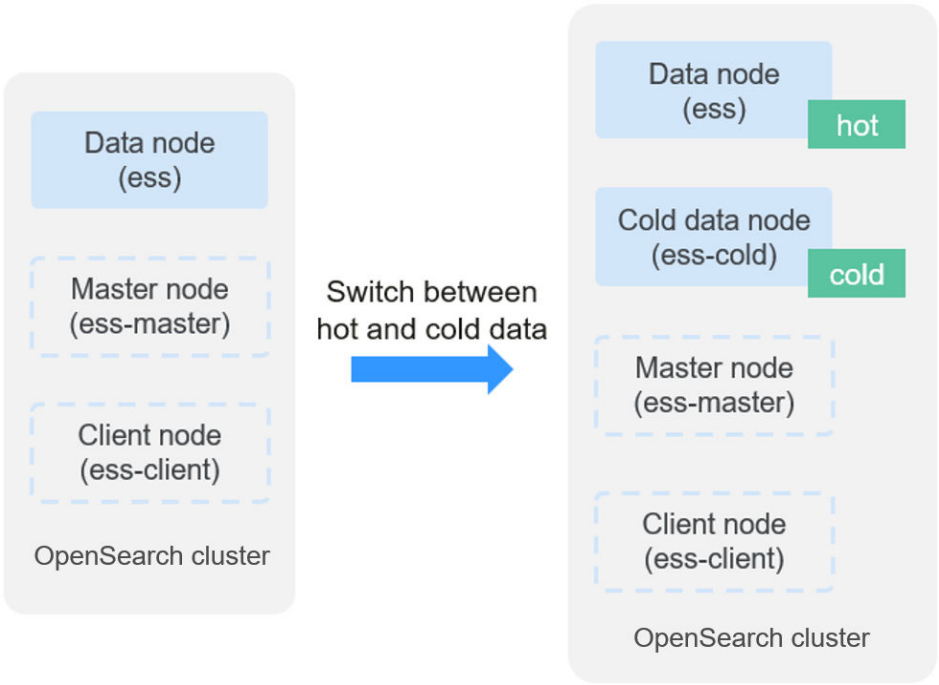
- Hot data: frequently accessed data, low latency required, stored using high-performance hardware (such as SSDs) to ensure fast write and query performance.
- Cold data: rarely accessed data, cost-optimized storage (such as HDDs).

If your cluster stores data used for different purposes, such as real-time analytics, log analytics, and monitoring data archives, you can switch between hot and cold storage for specified indexes to balance performance and costs.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

How the Feature Works

Figure 3-30 How cold/hot storage switchover works



The key is to allocate index data storage by node labels and index allocation policies.

- Node labels:
  - Data node (hot): Stores real-time data (hot data) by default and supports high-concurrency read/write requests.
  - Cold data node (cold): Stores historical data (cold data). Cold data nodes use less expensive hardware and deliver lower query performance than regular data nodes.
- Index allocation: You can configure index templates or directly specify index names to allocate specific indexes to data nodes or cold data nodes.

Procedure: Enable cold data nodes, and configure an index template or specific index settings to allocate index data storage. The cluster automatically allocates data based on your settings.

CAUTION

In comparison with data nodes, cold data nodes deliver lower query performance. Determine which type of node to use based on service needs.

Switching Over Between Hot and Cold Storage

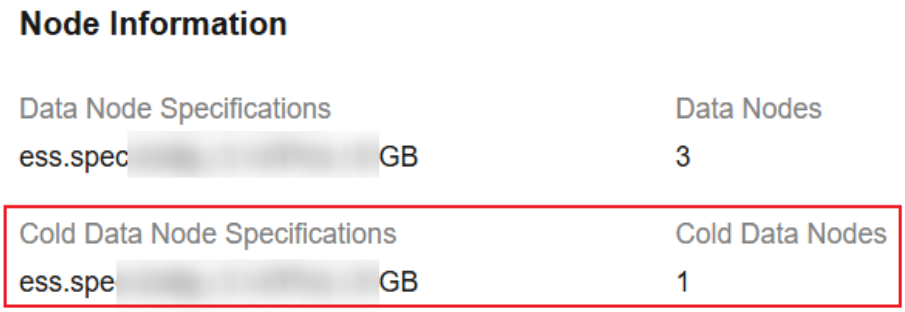
Step 1 Check whether cold data nodes are enabled in the target cluster.

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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, check whether the **Node Information** area contains cold data node information.

Figure 3-31 Cold data node information



   - If there is information about cold data nodes, the cluster has cold data nodes. Go to the next step.
   - If not, the cluster does not have cold data nodes. Add cold data nodes for the cluster and then go to the next step. For details about how to add cold data nodes, see **Adding New Node Types**.

Step 2 Log in to OpenSearch Dashboards.

1. On the cluster information page, click **Dashboards** in the upper-right corner to log in to OpenSearch Dashboards.
2. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Step 3 On OpenSearch Dashboards, configure index allocation policies.

You can configure index templates or directly specify index names to allocate specific indexes to data nodes or cold data nodes.

- **Configuring an index template**

  Configure an index template to allocate matching indexes to cold or hot data nodes. For example, run the following command to set a template to store newly created indexes whose name starts with **myindex** to cold data nodes:

```
PUT _template/{template_name}
{
  "order": 1,
  "index_patterns": "myindex*",
  "settings": {
    "refresh_interval": "30s",
    "number_of_shards": "3",
    "number_of_replicas": "0",
    "routing.allocation.require.box_type": "cold"
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-126 Key parameters

Parameter	Type	Description
index_patterns	String	Defines the index name matching pattern. The wildcard (*) can be used to match multiple indexes. For example, **myindex*** indicates all indexes whose name starts with **myindex**.
routing.allocation.require.box_type	String	Specifies the data node type for shard allocation. The value can be: – **cold**: Shards are allocated to cold data nodes. – **hot**: Shards are allocated to regular data nodes. If this parameter is not specified, shards will be evenly and randomly distributed among cold and hot data nodes.

- **Configuring specific indexes**

  To change the node type for an existing index, run the following command:

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

Table 3-127 Parameter description

Parameter	Type	Description
index_name	String	Specifies one or more indexes. – Single index: Enter the index name, for example, **my_index**. – Multiple indexes: Enter multiple index names and use a comma (,) to separate them, for example, **my_index1,my_index2**. – Wildcard: Use the wildcard (*) to match multiple indexes. For example, **myindex*** indicates all indexes whose name starts with **myindex**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Description
routing.allocation.require.box_type	String	Specifies the data node type for shard allocation. The value can be: – **cold**: Shards are allocated to cold data nodes. – **hot**: Shards are allocated to regular data nodes. If this parameter is not specified, shards will be evenly and randomly distributed among cold and hot data nodes.

Step 4 Verify the switchover between hot and cold storage.

Run the following command to check the distribution of index shards:

```
GET _cat/shards/{index_name}?v
```

When the data volume is large, the switchover may take a long time. There may be an intermediate state where the data of an index resides on both cold data nodes and data nodes.

As shown in the following figure, all shards of the **myindex** index are stored on the cold data node **css-e668-ess-cold-esn-1-1**.

```
index    shard prirep state      docs  store ip             node
myindex 1     p      STARTED 14085446 17.8gb 192.168.91.188 css-e668-ess-cold-esn-1-1
myindex 2     p      STARTED 14094005 17.9gb 192.168.91.188 css-e668-ess-cold-esn-1-1
myindex 0     p      STARTED 14094742 17.8gb 192.168.91.188 css-e668-ess-cold-esn-1-1
```

Step 5 Roll back the cold-hot switchover configuration.

To cancel the cold-hot switchover configuration, run the following command:

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

After the rollback, index data will be evenly and randomly allocated to both cold data nodes and data nodes.

----End

Related Operations

- You can scale cold data nodes by adding or reducing nodes or their storage capacity, or adding cold data nodes dynamically. For details, see **Managing Cluster Changes**.
- If your cluster does not have cold data nodes but you wish to cut storage costs, you can try using decoupled storage and compute. For details, see **Configuring Decoupled Storage and Compute**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3.11.4 Configuring Read/Write Splitting Between Leader and Follower

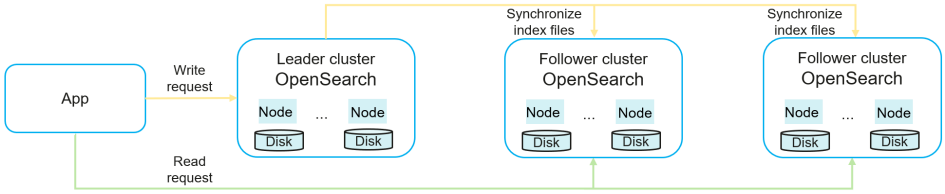
As business expands, and data volumes and access requests grow exponentially, a monolithic architecture where a single cluster handles both write and query requests faces typical challenges like resource contention and overload. To address these challenges, CSS introduces read/write splitting between OpenSearch clusters.

OpenSearch read/write splitting works by having leader and follower clusters collaborate together. The feature delivers the following benefits:

- Decoupled read and write loads: The leader cluster ensures data ingestion performance, while the follower clusters deliver scalable, high-concurrency query performance. There is no more resource contention, and peak loads are reduced.
- Flexible scalability: The write and query clusters can be scaled horizontally and independently. Cross-region cluster deployment is supported.
- Data consistency guarantee: Data can be synchronized in real time, with low latency. Incremental synchronization is also supported.

How the Feature Works

Figure 3-32 Read/write splitting



How read/write splitting between OpenSearch clusters works:

1. Data writes: Users send write requests, and the leader cluster handles these requests.
2. Data synchronization.

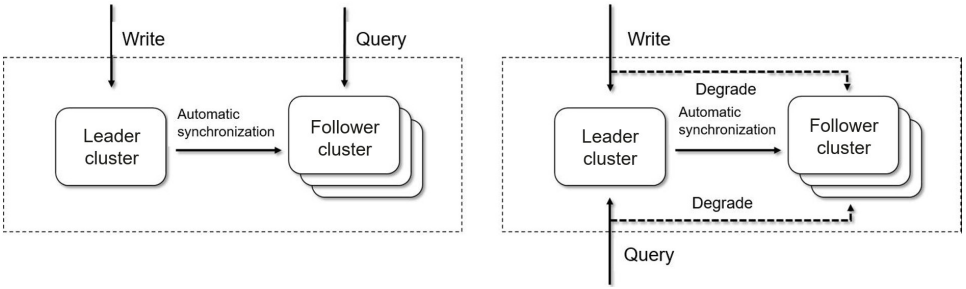
   The leader synchronizes data changes to the follower through a REST API. Two synchronization methods are supported:

   - Synchronizing a specified index.
   - Synchronizing matched indexes, for example, indexes that match a wildcard expression.
3. Query processing: Users send query requests. The follower cluster handles these requests and returns the results.

Figure 3-33 illustrates how read/write splitting works.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-33 How read/write splitting works



- Read and write split when both clusters are available (left): The leader handles writes, and the follower handles queries.
- Leader-to-follower switchover when the leader fails (right): If the leader becomes unavailable, the follower automatically upgrades to leader to ensure service continuity. For details, see **Case: Leader/Follower Switchover**.

Constraints

- Read/write splitting between leader and follower clusters is supported only in OpenSearch 2.19.0 and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).
- The leader and follower clusters must use the same software version.

Prerequisites

Two OpenSearch clusters of the same version have been created. One functions as the leader cluster, and the other the follower cluster. The follower cluster must be able to access the REST API (default port: 9200) of the leader cluster.

Logging In to OpenSearch Dashboards

Log in to OpenSearch Dashboards and go to the command execution page. OpenSearch clusters support multiple access methods. This topic uses OpenSearch Dashboards as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Connecting the Leader and Follower Clusters

1. In the **follower cluster**, run the following command to configure the leader cluster's information to connect them, so that data can be synchronized from the leader to the follower.

```
PUT /_cluster/settings
{
  "persistent" : {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  "cluster" : {
    "remote.rest" : {
      "{leader_name}" : {
        "seeds" : [
          "http://10.0.0.1:9200",
          "http://10.0.0.2:9200",
          "http://10.0.0.3:9200"
        ] ,
          "username": "test",
          "password": "*****"
      }
    }
  }
}
```

Table 3-128 Parameters for connecting the leader and follower clusters

Parameter	Mandatory	Type	Description
leader_name	Yes	String	Custom name of the leader cluster configuration task, used to identify the leader cluster connection in subsequent index synchronization configuration. Only letters (both uppercase and lowercase), digits, underscores (_), and hyphens (-) are allowed. Spaces and special characters (such as . # @) are not allowed.
seeds	Yes	String	List of addresses for accessing the leader cluster. • For a non-security mode cluster or a security-mode cluster that uses HTTP, the address format is http://xx.xx.xx.xx:9200. • For a security-mode cluster that uses HTTPS, the address format is https://xx.xx.xx.xx:9200.
username	Yes	String	Username of the leader cluster. This parameter is required only when security mode is enabled for the leader cluster.
password	Yes	String	Password of the leader cluster. This parameter is required only when security mode is enabled for the leader cluster.

Example response:

```
{
  "acknowledged" : true,  //Whether the operation is successful
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
"persistent" : {
  "cluster" : {
    "remote" : {
      "rest" : {
        "leader1" : {
          "seeds" : [
          "http://10.0.0.1:9200",
          "http://10.0.0.2:9200",
          "http://10.0.0.3:9200"
          ] ,
          "username": "test",
           "password": "*****"
        }
      }
    }
  }
 },
 "transient" : { }
}
```

2. After the configuration is complete, run the following command in the follower cluster to check the connection between the follower and leader clusters:

```
GET _remote/rest/info
```

Example response:

```
{
  "leader1" : {
    "connected" : true  //The two clusters are connected.
  }
}
```

Configuring Index Synchronization

1. Create an index synchronization task. There are two synchronization modes: synchronizing a specified index; and synchronizing matched indexes.

Table 3-129 Comparing two different synchronization modes

Synchronization Mode	Synchronizing a specified index	Synchronizing matched indexes
Synchronization Logic	Manually specify an index that you wish to synchronize from the leader cluster to the follower.	Specify an index matching pattern. For each index (including any new indexes created in the future) that matches this pattern in the leader cluster, the system automatically creates a task that will synchronize it to the follower cluster.
Scenario	When you want to precisely control the index synchronization scope (for example, to synchronize mission-critical indexes only)	When you want to synchronize many dynamically created indexes (for example, indexes automatically named by timestamp or predefined rules).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Synchronization Policy Modification	A synchronization task can be re-delivered for the same index. That is when you can modify the synchronization policy.	After a synchronization task is started, the synchronization policy can no longer be modified. You can only create a new pattern-matching index synchronization task to replace an existing one. Procedure: **Delete an existing task** > **Stopping Index Synchronization** > Create a new task.

The following describes how to configure a task of each type:

- **Synchronizing a specified index**

  Run the following command in the follower cluster to create a single-index synchronization task.

```
PUT start_remote_sync
{
  "remote_cluster": "{leader_name}",
  "remote_index": "{index_name_leader}",
  "local_index": "{index_name_follower}",
  "settings": {
    "number_of_replicas": 4
  },
  "settings_sync_enable": true,
  "settings_sync_patterns": ["*"],
  "settings_sync_exclude_patterns": ["index.routing.allocation.*"],
  "alias_sync_enable": true,
  "state_sync_enable": true
}
```

Table 3-130 Parameters for creating a single-index synchronization task

Parameter	Mandatory	Type	Default Value	Description
remote_cluster	Yes	String	N/A	The name of the leader cluster configuration task. The value must be consistent with the value of **leader_name** set in **Connecting the Leader and Follower Clusters**, for example, **leader1**.
remote_index	Yes	String	N/A	Name of the index to be synchronized in the leader cluster. Enter the name of an existing index in the leader cluster, for example, **data_leader**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
local_index	Yes	String	N/A	The name of the index in the follower cluster after synchronization from the leader, for example, **data_follower**. You can keep it the same as **remote_index**, but doing so is not recommended. This is because using identical names for indexes in the leader and follower clusters can lead to confusion during index management.
settings	No	Map	N/A	Index settings to be modified and their target values after synchronization to the follower cluster. This parameter allows you to flexibly configure index settings (such as the number of replicas and refresh interval) to match the hardware resources or service requirements in the follower cluster. If not configured, the index settings of the leader cluster will be used by default. Key and value: • **key**: the name of the setting, for example, number_of_replicas. • **value**: its new value, for example, 4. The following settings cannot be modified: number_of_shards, version.created, uuid, creation_date, and soft_deletes.enabled.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
settings_sync_enable	No	Boolean	false	Whether to periodically synchronize index settings from the leader cluster. The value can be: • **true**: Synchronize index settings from the leader cluster. • **false**: Not to synchronize index settings from the leader cluster when they are updated. Instead, keep the initial settings.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
settings_sync_patterns	No	String	* (matches all settings)	Leader-cluster index settings to be synchronized to the follower cluster. Specify the names of the settings to be synchronized. ● A single setting: Enter the setting name (for example, **index.max_regex_length**). ● Multiple settings: Enter multiple setting names and separate them with a comma (,), for example, **index.max_regex_length,index.max_refresh_listeners**. ● Wildcard: Use the wildcard * to match multiple settings. For example, **index*** matches all settings whose name starts with **index**. Constraints: ● This parameter takes effect only when **settings_sync_enable** is set to **true**. ● If the same setting is configured both in **settings** and **settings_sync_patterns**, the setting in **settings_sync_patterns** takes effect. ● The following settings cannot be synchronized to the follower cluster: index.number_of_shards, index.version.created, index.uuid, index.creation_date, index.history.uuid, index.number_of_replica

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
				s, index.auto_expand_replicas, and index.version.upgraded.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
settings_sync_exclude_patterns	No	String	Null (excludes none)	Leader-cluster index settings not to be synchronized to the follower cluster. Specify the names of the settings not to be synchronized. ● A single setting: Enter the setting name (for example, **index.max_regex_length**). ● Multiple settings: Enter multiple setting names and separate them with a comma (,), for example, **index.max_regex_length,index.max_refresh_listeners**. ● Wildcard: Use the wildcard * to match multiple settings. For example, **index*** matches all settings whose name starts with **index**. Constraints: ● This parameter takes effect only when **settings_sync_enable** is set to **true**. ● If both **settings_sync_patterns** and **settings_sync_exclude_patterns** are configured, only the settings that are in **settings_sync_patterns** but not in **settings_sync_exclude_patterns** will be synchronized to the follower cluster. ● If the indexes you plan to synchronize include

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
				cold indexes, make sure the follower cluster contains cold data nodes. Otherwise, data synchronization will fail. The following configuration is required to synchronize cold indexes to the data nodes in the follower cluster (in such case, these indexes cease to be cold indexes in the follower cluster): `"settings_sync_exclude_patterns": ["index.routing.allocation.require.box_type"]`
alias_sync_enable	No	Boolean	false	Whether to periodically synchronize index aliases from the leader cluster. The value can be: • **true**: Periodically synchronize index aliases from the leader cluster. • **false**: Not to periodically synchronize index aliases from the leader cluster.
state_sync_enable	No	Boolean	false	Whether to periodically synchronize index status from the leader cluster. The value can be: • **true**: Periodically synchronize index status from the leader cluster. • **false**: Not to periodically synchronize index status from the leader cluster.

– **Synchronizing matched indexes**

Run the following command in the follower cluster to create a pattern-matching index synchronization task:

```
PUT auto_sync/pattern/${pattern_name}
{
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  "remote_cluster": "{leader_name}",
  "remote_index_patterns": "{index_name}",
  "local_index_pattern": "{{remote_index}}-sync",
  "apply_exist_index": true,
  "settings": {
    "number_of_replicas": 4
  },
  "settings_sync_enable": true,
  "settings_sync_patterns": [
    "*"
  ],
  "settings_sync_exclude_patterns": [
    "index.routing.allocation.*"
  ],
  "alias_sync_enable": true,
  "state_sync_enable": true
}
```

Table 3-131 Parameters for creating a pattern-matching index synchronization task

Parameter	Mandatory	Type	Default Value	Description
pattern_name	Yes	String	N/A	The name of the index name matching pattern.
remote_cluster	Yes	String	N/A	The name of the leader cluster configuration task. The value must be consistent with the value of **leader_name** set in **Connecting the Leader and Follower Clusters**, for example, **leader1**.
remote_index_patterns	Yes	String	N/A	The leader-cluster index matching pattern. • Single index: Enter the index name, for example, **my_index**. • Wildcard: Use the wildcard (*) to match multiple indexes. For example, **myindex*** indicates all indexes whose name starts with **myindex**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
local_index_pattern	Yes	String	N/A	Index naming pattern in the follower cluster after synchronization from the leader cluster. Template replacement is supported. For example, if this parameter is set to **{{remote_index}}-sync**, the index name **log1** changes to **log1-sync** after synchronization. **CAUTION** If leader/follower switchover is likely to happen, set this parameter to **{{remote_index}}** to ensure that the leader and follower clusters use identical index names.
apply_exist_index	Yes	Boolean	true	Whether to synchronize existing indexes from the leader cluster. The value can be: • **true**: Synchronize both existing and new indexes, as long as they match the pattern you define. • **false**: Synchronize new indexes only. Skipping the synchronization of existing indexes may lead to data inconsistency between the leader and follower clusters. Determine how to set this parameter based on your service needs.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
settings	No	Map	N/A	Index settings to be modified and their target values after synchronization to the follower cluster. This parameter allows you to flexibly configure index settings (such as the number of replicas and refresh interval) to match the hardware resources or service requirements in the follower cluster. If not configured, the index settings of the leader cluster will be used by default. Key and value: • **key**: the name of the setting, for example, number_of_replicas. • **value**: its new value, for example, 4. The following settings cannot be modified: number_of_shards, version.created, uuid, creation_date, and soft_deletes.enabled.
settings_sync_enable	No	Boolean	false	Whether to periodically synchronize index settings from the leader cluster. The value can be: • **true**: Synchronize index settings from the leader cluster. • **false**: Not to synchronize index settings from the leader cluster when they are updated. Instead, keep the initial settings.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
settings_sync_patterns	No	String	* (matches all settings)	Leader-cluster index settings to be synchronized to the follower cluster. Specify the names of the settings to be synchronized. ● A single setting: Enter the setting name (for example, **index.max_regex_length**). ● Multiple settings: Enter multiple setting names and separate them with a comma (,), for example, **index.max_regex_length,index.max_refresh_listeners**. ● Wildcard: Use the wildcard * to match multiple settings. For example, **index*** matches all settings whose name starts with **index**. Constraints: ● This parameter takes effect only when **settings_sync_enable** is set to **true**. ● If the same setting is configured both in **settings** and **settings_sync_patterns**, the setting in **settings_sync_patterns** takes effect. ● The following settings cannot be synchronized to the follower cluster: index.number_of_shards, index.version.created, index.uuid, index.creation_date, index.history.uuid, index.number_of_replicas

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
				, index.auto_expand_replicas, and index.version.upgraded.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
settings_sync_exclude_patterns	No	String	Null (excludes none)	Leader-cluster index settings not to be synchronized to the follower cluster. Specify the names of the settings not to be synchronized. • A single setting: Enter the setting name (for example, **index.max_regex_length**). • Multiple settings: Enter multiple setting names and separate them with a comma (,), for example, **index.max_regex_length,index.max_refresh_listeners**. • Wildcard: Use the wildcard * to match multiple settings. For example, **index*** matches all settings whose name starts with **index**. Constraints: • This parameter takes effect only when **settings_sync_enable** is set to **true**. • If both **settings_sync_patterns** and **settings_sync_exclude_patterns** are configured, only the settings that are in **settings_sync_patterns** but not in **settings_sync_exclude_patterns** will be synchronized to the follower cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
				• If the indexes you plan to synchronize include cold indexes, make sure the follower cluster contains cold data nodes. Otherwise, data synchronization will fail. The following configuration is required to synchronize cold indexes to the data nodes in the follower cluster (in such case, these indexes cease to be cold indexes in the follower cluster): `"settings_sync_exclude_patterns" : ["index.routing.allocation.require.box_type"]`
alias_sync_enable	No	Boolean	false	Whether to periodically synchronize index aliases from the leader cluster. The value can be: • **true**: Periodically synchronize index aliases from the leader cluster. • **false**: Not to periodically synchronize index aliases from the leader cluster.
state_sync_enable	No	Boolean	false	Whether to periodically synchronize index status from the leader cluster. The value can be: • **true**: Periodically synchronize index status from the leader cluster. • **false**: Not to periodically synchronize index status from the leader cluster.

2. Check the index synchronization status to confirm successful configuration.

   After synchronization is enabled, the index in the follower cluster becomes read-only. Synchronization is performed periodically to ensure data

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

consistency. The default synchronization interval is 30 seconds. For how to change it, see **Changing the Index Synchronization Interval**.

Run the following command in the follower cluster to obtain the synchronization status of a specified index:

```
GET {index_name}/sync_stats
```

An example output is as follows:

```
{
  "indices" : {
    "data1_follower" : {
      "shards" : {
        "0" : [
          {
            "primary" : false,            // Whether it is a primary shard
            "total_synced_times" : 27,    // Total synchronization times
            "total_empty_times" : 25,     // Total number of times when no data is synchronized between the leader and follower clusters because they have identical shards and data
            "total_synced_files" : 4,     // Number of synchronized files
            "total_synced_bytes" : 3580, // Total size of synchronized files
            "total_paused_nanos" : 0,   //Duration of synchronization pauses due to traffic throttling
            "total_paused_times" : 0,   //Number of synchronization pauses due to traffic throttling
            "current" : {
              "files_count" : 0, //Number of files that are being synchronized
              "finished_files_count" : 0, //Number of files that have been synchronized
              "bytes" : 0, //Size of files that are being synchronized
              "finished_bytes" : 0 //Size of files that have been synchronized
            }
          },
          {
            "primary" : true,             // Whether it is a primary shard
            "total_synced_times" : 28,    // Total synchronization times
            "total_empty_times" : 26,     // Total number of times when no data is synchronized between the leader and follower clusters because they have identical shards and data
            "total_synced_files": 20,    // Number of synchronized files
            "total_synced_bytes": 17547, // Total size of synchronized files
            "total_paused_nanos" : 0,     //Duration of synchronization pauses due to traffic throttling
            "total_paused_times" : 0,     //Number of synchronization pauses due to traffic throttling
            "current" : {
              "files_count" : 0, //Number of files that are being synchronized
              "finished_files_count" : 0, //Number of files that have been synchronized
              "bytes" : 0, //Size of files that are being synchronized
              "finished_bytes" : 0 //Size of files that have been synchronized
            }
          }
        ]
      }
    }
  }
}
```

Managing Pattern-matching Index Synchronization

You can query and delete existing pattern-matching index synchronization tasks.

- **Query** existing pattern-matching index synchronization tasks.
  - Run the following command in the follower cluster to query existing pattern-matching index synchronization tasks:
    ```
    GET auto_sync/pattern
    ```
  - Run the following command in the follower cluster to query details about a specified pattern-matching index synchronization task:
    ```
    GET auto_sync/pattern/{pattern_name}
    ```

  Example response:

  ```
  {
    "patterns" : [
  ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  {
    "name" : "pattern1",
    "pattern" : {
      "remote_cluster" : "leader",
      "remote_index_patterns" : [
        "log*"
      ],
      "local_index_pattern" : "{{remote_index}}-sync",
      "settings" : { }
    }
  }
 ]
}
```

- **Delete** an existing pattern-matching index synchronization task.

  CAUTION

  Deleting a pattern-matching index synchronization task only prevents new synchronization tasks from being created for matched indexes. It does not stop synchronization tasks that have already been created. To stop these tasks, manually perform **Stopping Index Synchronization**.

  a. Run the following command in the follower cluster to delete a specified pattern-matching index synchronization task:
     ```
     DELETE auto_sync/pattern/{pattern_name}
     ```
  b. Run the following command to query the list of pattern-matching index synchronization tasks to confirm successful deletion.
     ```
     GET auto_sync/pattern
     ```

Changing the Index Synchronization Interval

The default index synchronization interval between the leader and follower clusters is 30 seconds. To change it for a specified index, run the following command:

```
PUT {index_name}/_settings
{
  "index.remote_sync.sync_interval": "2s"
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-132 Parameters for modifying the index synchronization interval

Parameter	Type	Default Value	Description
index_name	String	N/A	Specifies one or more indexes. • Single index: Enter the index name, for example, **my_index**. • Multiple indexes: Enter multiple index names and use a comma (,) to separate them, for example, **my_index1,my_index2**. • Wildcard: Use the wildcard (*) to match multiple indexes. For example, **myindex*** indicates all indexes whose name starts with **myindex**.
index.remote_sync.sync_interval	String	30s (recommended)	The index synchronization interval. Value format: number + unit • Number: a natural number • Unit: nanos (nanosecond), micros (microsecond), ms (millisecond), s (second), m (minute), h (hour), or d (day) Minimum value: **1s** You may reduce this interval if the timeliness of data is prioritized, but doing so will increase the cluster's CPU load.

Stopping Index Synchronization

Run the following command in the follower cluster to stop a single-index synchronization task immediately:

```
PUT {index_name}/stop_remote_sync
```

After this command is executed:

- Any subsequent changes made to the index in the leader cluster will no longer be synchronized to the follower cluster.
- The index in the follower cluster will be removed from read-only mode, allowing new data to be written to it.

CAUTION

Index synchronization cannot be stopped when a snapshot is being created for the index. To stop synchronization, you must wait until snapshot creation is complete.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Advanced Settings

In addition to basic synchronization settings, read/write splitting allows you to adjust synchronization behavior through advanced cluster-level settings, including forced synchronization, index deletion synchronization, and synchronization rate control.

Run the following command in the follower cluster to configure advanced settings:

```
PUT _cluster/settings
{
  "persistent": {
    "remote_sync.force_synchronize": true,
    "remote_sync.auto_delete_index": true,
    "remote_sync.chunk_size": "2MB",
    "remote_sync.max_concurrent_file_chunks": 20,
    "remote_sync.max_bytes_per_sec": "100MB"
  }
}
```

Table 3-133 Advanced settings

Parameter	Type	Default Value	Description
remote_sync.force_synchronize	Boolean	false	Whether to force index synchronization. • **true**: Forces synchronization. In each synchronization cycle, the index metadata of the leader cluster is forcibly synchronized to the follower cluster, even if the number of documents has not changed. • **false**: Does not force synchronization. Index metadata is synchronized only when the number of documents in the leader cluster changes.
remote_sync.auto_delete_index	Boolean	false	Whether to synchronize index deletion to the follower cluster. • **true**: When an index is deleted from the leader cluster, the corresponding index is automatically deleted in the follower cluster. • **false**: When an index is deleted from the leader cluster, the corresponding index is retained in the follower cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
remote_sync.chunk_size	String	1MB	The size of a single chunk for index synchronization. Value format: number + unit • Number: a natural number • Unit: B, K, KB, M, MB, G, GB, T, TB, P, or PB (case-insensitive) We recommend setting this value to at least 1 MB. Setting a value smaller than 1 MB may degrade system performance.
remote_sync.max_concurrent_file_chunks	Integer	10	Maximum number of document chunks synchronized concurrently on a single node. Value range: 1 to 100. Set this parameter based on the node bandwidth and CPU resources available.
remote_sync.max_bytes_per_sec	String	40MB	Maximum size of data transmitted per second on a single node during synchronization. You can set this parameter to prevent data synchronization tasks from exhausting the network bandwidth. Value format: number + unit • Number: a natural number • Unit: B, K, KB, M, MB, G, GB, T, TB, P, or PB (case-insensitive) The value **0** indicates no limit. When your cluster's query traffic is low, you can increase the value, but doing so may increase the network load.

Case: Leader/Follower Switchover

When the leader cluster becomes faulty, you can perform a leader/follower switchover to have the follower cluster take over to ensure service continuity. To allow this to happen seamlessly and ensure data consistency, use identical index names between the leader and follower clusters when configuring index synchronization.

1. Determine the index synchronization method between the leader and follower clusters.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Run the following command in the follower cluster to check for pattern-matching index synchronization tasks:

```
GET auto_sync/pattern
```

- If there are no such tasks, synchronization is configured for specified indexes between the leader and follower clusters. In this case, go to **3**.
- If there are such tasks, index synchronization between the leader and follower clusters is based on an index matching pattern. In this case, perform the next step.

2. Run the following command in the follower cluster to delete a specified pattern-matching index synchronization task:

```
DELETE auto_sync/pattern/{pattern_name}
```

3. Stop index synchronization in the follower cluster, and switch read and write traffic to it. Make sure the follower cluster can handle requests independently.

   Run the following command in the follower cluster to stop single-index synchronization tasks for specified indexes:

```
PUT */stop_remote_sync
```

4. After the leader cluster recovers, configure information about the follower cluster in the leader cluster, and connect the leader and follower clusters again. For details, see **Connecting the Leader and Follower Clusters**.
5. Under the leader cluster, perform **Configuring Index Synchronization** to synchronize data from the follower cluster to the leader cluster, and then perform a leader/follower switchover to switch back.

3.11.5 Configuring Flow Control

Configure flow control policies for your OpenSearch cluster in both the inbound and outbound directions, ensuring cluster stability by safeguarding against abnormal traffic.

An OpenSearch cluster can become overloaded due to traffic surges, malicious requests, and internal resource competition, which can even lead to node failures. Through policies like client request throttling, backpressure, and traffic pattern analysis, flow control ensures proper resource allocation, thereby protecting clusters from overload. It covers the following scenarios:

- High-concurrency write handling: mitigates the risk of out-of-memory (OOM) exceptions under heavy write loads.
- Security defense: controls access by IP address using both blacklists and whitelists.
- Emergency response: blocks malicious or abnormal traffic in one click.
- Performance optimization: optimizes flow control thresholds and policies based on collected statistics.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

How the Feature Works

Table 3-134 Flow control policies

Policy	What It Does	Details
HTTP/HTTPS flow control	Controls client access traffic using blacklists and whitelists, an upper limit on concurrent connections, and a rate limit on new connection attempts. • Blacklists and whitelists: A whitelist takes precedence over a blacklist. If an IP address is on both lists, it is allowed. Requests sent through blacklisted connections will be ignored. • Concurrent connection limit: Limits the total number of concurrent HTTP connections to prevent overload. • New connection limit: Limits the number of new connections that can be set up per second. The warmup_period parameter protects against connection floods, allowing traffic to grow gradually and steadily. When HTTP/HTTPS flow control is enabled, requests from blacklisted IP addresses are always rejected; for IP addresses on the whitelist, no flow control rules apply; for other IP addresses, when either the concurrent connection limit or the new connection limit is reached, requests from them will be rejected.	**Configuring HTTP/HTTPS Flow Control**
Memory flow control	When the heap memory usage exceeds a pre-defined threshold (for example, 80%), the system stops receiving large requests, and garbage collection (GC) is triggered to reclaim memory. Write traffic is throttled by setting the backpressure factor (in_flight_factor) and the maximum delay for request handling (max). When memory-based flow control is enabled, large requests may be delayed for a long time when the cluster's heap memory usage exceeds the configured threshold.	**Configuring Memory-based Flow Control**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Policy	What It Does	Details
One-click traffic blocking	When triggered, the system immediately disconnects all client connections that are not whitelisted, with the exception of those used for OpenSearch Dashboards access or O&M and monitoring APIs, as an effort to restore the cluster.	**Configuring One-Click Traffic Blocking**
Request statistics sampling and analysis	Records request metrics (such as bulk writes and queries) by client IP address, and exposes them via a statistics API to evaluate the cluster load and proactively identify abnormal traffic patterns.	**Configuring Request Statistics Sampling and Analysis**
Access logging	Records the URLs and bodies of HTTP/HTTPS requests for cluster load and client request analysis. Access logs can also be saved to files (that is, persisted to disk) to facilitate troubleshooting and performance analysis. Enabling access logging incurs extra CPU and memory overhead, which may slow down request handling.	• **Configuring Access Logging** • **Configuring Access Logging in Files**

Constraints

Flow control is supported only in OpenSearch 2.19.0 and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).

Logging In to OpenSearch Dashboards

Log in to OpenSearch Dashboards and go to the command execution page. OpenSearch clusters support multiple access methods. This topic uses OpenSearch Dashboards as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Configuring HTTP/HTTPS Flow Control

Control client access traffic using blacklists and whitelists, an upper limit on concurrent connections, and a rate limit on new connection attempts, to prevent overload.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

1. Enable HTTP/HTTPS flow control.

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.http.enabled": true,
    "flowcontrol.http.allow": ["192.168.0.1/24", "192.168.2.1/24"],
    "flowcontrol.http.deny": "192.168.1.1/24",
    "flowcontrol.http.concurrent": 1000,
    "flowcontrol.http.newconnect": 1000,
    "flowcontrol.http.warmup_period": 0
  }
}
```

Table 3-135 Parameters for configuring HTTP/HTTPS flow control

Parameter	Type	Default Value	Description
flowcontrol.http.enabled	Boolean	false	Enable or disable HTTP/HTTPS flow control. When enabled, flow control will be performed based on relevant settings. The value can be: ● true: Enable HTTP/HTTPS flow control. ● false: Disable HTTP/HTTPS flow control.
flowcontrol.http.allow	List<String>	Null (no whitelist)	A whitelist of client IP addresses or CIDR blocks that are allowed to access the cluster, supporting: ● Individual IP addresses, for example, **192.18.0.1**. ● CIDR blocks, for example, **192.168.0.0/24**. ● Multiple IP addresses or CIDR blocks separated by commas (,), for example, **192.168.0.1/24, 192.168.2.1/24**. Setting this parameter to **null** restores the default value.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
flowcontrol.http.deny	List<String>	Null (no blacklist)	A blacklist of client IP addresses or CIDR blocks that are not allowed to access the cluster. A whitelist takes precedence over a blacklist. A blacklist supports the following: • Individual IP addresses, for example, **192.18.0.1**. • CIDR blocks, for example, **192.168.0.0/24**. • Multiple IP addresses or CIDR blocks separated by commas (,), for example, **192.168.0.1/24, 192.168.2.1/24**. Setting this parameter to **null** restores the default value.
flowcontrol.http.concurrent	Integer	Node vCPUs x 600	Maximum number of concurrent HTTP/HTTPS connections that can be handled by a node. Minimum value: **10** Setting this parameter to **null** restores the default value.
flowcontrol.http.newconnect	Integer	Node vCPUs x 200	Maximum number of new HTTP/HTTPS connections that can be created per second per node. Minimum value: **10** Setting this parameter to **null** restores the default value.
flowcontrol.http.warmup_period	Integer	0 (no grace period before reaching the full capacity)	A grace period during which a system gradually ramps up from accepting zero HTTP/HTTPS requests to its full, maximum capacity. Value range: 0–10000 Unit: ms For example, if **flowcontrol.http.newconnect** is set to **100** and **flowcontrol.http.warmup_period** is set to **5000ms**, it takes 5 seconds for the system to reach 100 new connections per second. Setting this parameter to **null** restores the default value.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. Disable HTTP/HTTPS flow control.

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.http.enabled": false
  }
}
```

Configuring Memory-based Flow Control

Enable write throttling to mitigate the risk of OOM exceptions when the heap memory usage of a node exceeds a predefined threshold.

1. Enable memory-based flow control.

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.memory.enabled": true,
    "flowcontrol.memory.heap_limit": "80%"
  }
}
```

Table 3-136 Memory-based flow control parameters

Parameter	Type	Default Value	Description
flowcontrol.memory.enabled	Boolean	true	Whether to enable memory-based flow control. When enabled, node heap memory usage is monitored and a threshold is set, and writes are throttled when this threshold is reached. The value can be: • true: Enable memory-based flow control. • false: Disable memory-based flow control.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
flowcontrol.memory.heap_limit	String	90% (conservative threshold)	Node heap memory usage threshold. When this threshold is exceeded, a backpressure mechanism is triggered. Value range: 10%–100% • When the heap memory usage exceeds this threshold, the system stops processing client requests that are larger than 64 KB. Processing resumes only when the heap memory usage drops below this threshold. • When the heap memory usage is five percentage points below this threshold, the system continues processing requests. However, it restricts the total data read to 5% of the total heap memory capacity per cycle. This limit, which is configured using the **flowcontrol.memory.once_free_max** parameter, creates a memory buffer to prevent immediate exhaustion when processing resumes. • While the heap memory usage stays above this threshold, the system cannot accept new client requests. If the **flowcontrol.memory.nudges_gc** parameter is set to **true**, the system will actively trigger garbage collection (GC) and repeatedly attempt to reclaim memory until usage drops below this threshold. This helps prevent system breakdown caused by potential memory leaks. In practice, you are advised to set this parameter to 80% or lower to reserve heap memory for non-read tasks, for example, segment merge. Setting this parameter to **null** restores the default value.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
flowcontrol.holding.in_flight_factor	Float	1.0 (recommended)	Backpressure factor, which controls the sensitivity of memory-based backpressure. A larger value indicates more powerful write throttling. Value range: ≥ 0.5 This parameter estimates the potential heap memory impact of an incoming large request. The calculation is as follows: **in_flight_factor x Request body size**. The resulting estimate is then used to apply memory-based backpressure and throttling. Setting this parameter to **null** restores the default value.
flowcontrol.holding.max	TimeValue	60s	Maximum request handling delay allowed before requests are handled according to the policy defined by **flowcontrol.holding.max_strategy**. Value range: ≥ 15s Unit: second Generally, you should configure this parameter based on the **flowcontrol.holding.max_strategy** setting. ● When **flowcontrol.holding.max_strategy** is set to **soft**, keep the value of this parameter lower than the client request timeout. Additionally, reserve some request execution time. ● When **flowcontrol.holding.max_strategy** is set to **hard**, keep the value of this parameter higher than the client request timeout. ● When **flowcontrol.holding.max_strategy** is set to **keep**, this parameter is invalid. Setting this parameter to **null** restores the default value.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
flowcontrol.holding.max_strategy	String	keep	Handling policy or action taken for requests delayed longer than **flowcontrol.holding.max**. The value can be: • **keep**: Maintain the backpressure state and wait for the heap memory usage to drop. The server determines whether to release requests based on real-time memory usage. In this mode, requests will be delayed until the memory usage drops to a level that allows the processing to resume. This may cause request timeout. • **soft**: Forcibly execute the requests, but the **inFlight** circuit breaker gets to decide whether to reject them. **inFlight** is a native OpenSearch circuit breaker designed to prevent system overload. For details, see **Circuit breaker settings**. This mode allows requests that have been delayed longer than **flowcontrol.holding.max** to proceed. However, it may still cause memory usage to spike and eventually lead to memory overflow. • **hard**: Reject the requests immediately and disconnect client connections. This will drop some requests. Setting this parameter to **null** restores the default value.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
flowcontrol.memory.once_free_max	String	5%	Maximum amount of memory (in the form of a percentage of node memory) that flow control is allowed to free in a single reclamation cycle. This parameter prevents overly aggressive memory reclamation that could lead to sudden request surges after memory pressure subsides. Value range: 1%–50% Setting this parameter to **null** restores the default value.
flowcontrol.memory.nudges_gc	Boolean	true (recommended)	Whether to trigger garbage collection (GC) to reclaim memory when the write pressure is too high. (The backpressure connection pool is checked every second. The write pressure is considered high if all existing connections are blocked and new write requests cannot be accepted.) The value can be: • true: Trigger GC. • false: Not to trigger GC. Setting this parameter to **null** restores the default value.

2. Disable memory-based flow control.

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.memory.enabled": false
  }
}
```

Configuring One-Click Traffic Blocking

In case of emergencies, the system immediately disconnects all client connections (excluding those used for OpenSearch Dashboards access or O&M and monitoring APIs) to restore clusters.

1. Enable one-click traffic blocking.

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.break.enabled": true
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-137 Parameters for configuring one-click traffic blocking

Parameter	Type	Default Value	Description
flowcontrol.break.enabled	Boolean	false	Whether to enable one-click traffic blocking (similar to a circuit breaker). When enabled, the system immediately disconnects all client connections, but not those used for OpenSearch Dashboards access or O&M and monitoring APIs. The value can be: • true: Enable one-click disconnection. • false: Disable one-click blocking.

2. Disable one-click disconnection:

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.break.enabled": false
  }
}
```

Configuring Request Statistics Sampling and Analysis

Collect request metrics by client IP address to help identify abnormal traffic patterns.

1. Enable request statistics sampling.

```
PUT _cluster/settings
{
  "transient": {
    "flowcontrol.log.access.enabled": true
  }
}
```

Table 3-138 Parameters for request statistics sampling

Parameter	Type	Default Value	Description
flowcontrol.log.access.enabled	Boolean	false	Whether to enable request statistics sampling, that is, whether to collect request metrics (such as bulk writes and search/msearch requests) by client IP address. The value can be: • true: Enable request statistics sampling. • false (default): Disable request statistics sampling.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
flowcontrol.log.access.count	Integer	10	Maximum number of client IP addresses sampled. Value range: 0–100 Setting this parameter to **null** restores the default value.

2. Check the sampled statistics to analyze the traffic pattern and flow control status by client IP address.
   - Check the flow control status of all nodes.
     ```
     GET /_nodes/stats/filter/v2
     ```
   - Check the flow control details of all nodes.
     ```
     GET /_nodes/stats/filter/v2?detail
     ```
   - Check the flow control status of a specified node.
     ```
     GET /_nodes/{node_id}/stats/filter/v2
     ```

     Table 3-139 Parameter description

Parameter	Type	Default Value	Description
node_id	String	N/A	Specifies one or more cluster nodes. • Single node: Enter the node ID. • Multiple nodes: Enter multiple node IDs and use a comma (,) to separate them. You can run the following command to obtain node IDs: `GET _cat/nodes?s=n&h=n,id&v=true&full_id=true`

Example response:

```
{
  "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  },
  "cluster_name" : "css-xxxx",
  "nodes" : {
    "d3qnVIpPTtSoadkV0LQEkA" : {
      "name" : "css-xxxx-ess-esn-1-1",
      "host" : "192.168.x.x",
      "timestamp" : 1672236425112,
      "flow_control" : {
        "http" : {
          "current_connect" : 52,
          "rejected_concurrent" : 0,
          "rejected_rate" : 0,
          "rejected_black" : 0,
          "rejected_breaker" : 0
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
      },
      "access_items" : [
        {
          "remote_address" : "10.0.0.x",
          "search_count" : 0,
          "bulk_count" : 0,
          "other_count" : 4
        }
      ],
      "holding_requests" : 0
    }
  }
 }
}
```

Table 3-140 Response parameters

Parameter	Description
current_connect	Number of HTTP connections to a node, which is recorded regardless of whether flow control is enabled. This value is equivalent to the **current_open** value of **GET /_nodes/stats/http** API. It shows the current client connections of each node.
rejected_concurrent	Number of concurrent connections rejected during flow control. This metric is available only when **flowcontrol.http.enabled** is set to **true**. The count will not be cleared when flow control is disabled.
rejected_rate	Number of new connections rejected during flow control. This metric is available only when **flowcontrol.http.enabled** is set to **true**. The count will not be cleared when flow control is disabled.
rejected_black	Number of new connections rejected by a preconfigured blacklist during flow control. This metric is available only when **flowcontrol.http.enabled** is set to **true**. The count will not be cleared when flow control is disabled.
rejected_breaker	Number of new connections rejected during one-click traffic blocking. This metric is available only when **flowcontrol.break.enabled** is set to **true**. The count will not be cleared when one-click traffic blocking is disabled.
access_items	IP addresses of clients that recently accessed the cluster. The number of client IP addresses sampled is determined by **flowcontrol.log.access.count**.
remote_address	IP addresses and the number of requests.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
search_count	Number of times a client accessed a database using **_search** and **_msearch**.
bulk_count	Number of times a client accessed a database using **_bulk**.
other_count	Number of times a client accessed a database using other request methods.
holding_requests	Number of connections to the current node where writes are halted due to flow control.

3. Disable request statistics sampling.

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.log.access.enabled": false
  }
}
```

Configuring Access Logging

When access logging is enabled, the system records the URLs and bodies of HTTP/HTTPS requests for cluster load and request analysis. Then, you can use the result to optimize cluster performance.

1. Enable access logging.
   - Enable access logging for all nodes in a cluster.
     ```
     PUT /_access_log?duration_limit=30s&capacity_limit=1mb
     ```
   - Enable access logging for a specified node in a cluster.
     ```
     PUT /_access_log/{node_id}?duration_limit=30s&capacity_limit=1mb
     ```

Table 3-141 Parameters for enabling access logging

Parameter	Type	Default Value	Description
duration_limit	String	30	Maximum duration for access logging. When this limit is reached, access logging stops. Value range: 10 to 120 Unit: s Setting this parameter to **null** restores the default value. Access logging stops when either **duration_limit** or **capacity_limit** is reached.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
capacity_limit	String	1	Maximum access log size. When the size of an access log reaches this limit, access logging stops. Value range: 1 to 5 Unit: MB Setting this parameter to **null** restores the default value. Access logging stops when either **duration_limit** or **capacity_limit** is reached.

2. Check access logs.
   - Check the access logs of all nodes in a cluster.
     ```
     GET /_access_log
     ```
   - Check the access logs of a specified cluster node.
     ```
     GET /_access_log/{node_id}
     ```

   Example response:

```
{
  "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  },
  "cluster_name" : "css-flowcontroller",
  "nodes" : {
    "8x-ZHu-wTemBQwpcGivFKg" : {
      "name" : "css-flowcontroller-ess-esn-1-1",
      "host" : "10.0.0.98",
      "count" : 2,
      "access" : [
        {
          "time" : "2021-02-23 02:09:50",
          "remote_address" : "/10.0.0.98:28191",
          "url" : "/_access/security/log?pretty",
          "method" : "GET",
          "content" : ""
        },
        {
          "time" : "2021-02-23 02:09:52",
          "remote_address" : "/10.0.0.98:28193",
          "url" : "/_access/security/log?pretty",
          "method" : "GET",
          "content" : ""
        }
      ]
    }
  }
}
```

Table 3-142 Response parameters

Parameter	Description
name	Node name

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
host	Node IP address
count	Number of node access requests in a statistical period
access	Details about node access requests in a statistical period
time	Request time
remote_address	Source IP address and port number in the request
url	Original URL of the request
method	Request method
content	Request content. If the value is an empty string (""), there is no request body.

3. Delete access logs. Logs are stored in the memory. After viewing the logs, you should delete them promptly to reclaim memory resources, and by doing so avoid impacting system performance.
   a. Delete access logs for all nodes.
      ```
      DELETE /_access_log
      ```
   b. Check the access logs again to confirm successful deletion.
      ```
      GET /_access_log
      ```

Configuring Access Logging in Files

Access logs can be persisted to disk for troubleshooting and analysis. Use this function sparingly, as it can impact cluster performance. Remember to disable it immediately after resolving the issue.

1. Enable access logging in files.
   ```
   PUT /_cluster/settings
   {
     "persistent": {
       "flowcontrol.log.file.enabled": true
     }
   }
   ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-143 Enable access logging in files

Parameter	Type	Default Value	Description
flowcontrol.log.file.enabled	Boolean	false	Whether to record access logs in files. When enabled, the log of each access request is recorded in files. The log file name is **Cluster name_access_log.log**. You can check this file only through the log backup function. For details about how to back up logs, see **Backing Up Logs to OBS**. The value can be: • true: Record access logs in files. • false: Not to record access logs in files.

2. Disable access logging in files.

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.log.file.enabled": false
  }
}
```

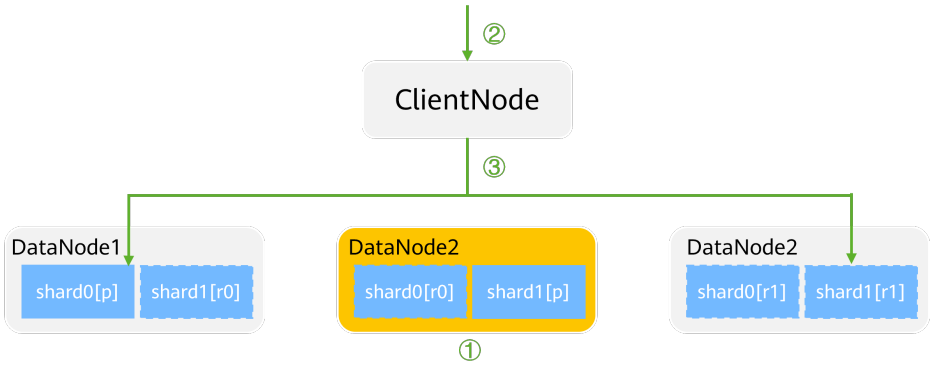
3.11.6 Configuring Query Traffic Isolation

In distributed search and data analytics scenarios, faulty cluster nodes—with memory faults or sustained high CPU usage—may lead to increased query response times or service availability issues. To address this, we need a quick and effective way to route query traffic away from faulty nodes and redirect it to other healthy nodes, thus ensuring service continuity and stability. CSS OpenSearch clusters provide a query traffic isolation feature for this purpose. With it, you can configure specific nodes as low-priority nodes or isolated nodes and define an index whitelist for exceptions. For shard query requests to indexes that are not on this whitelist, the system preferentially or completely bypass these nodes, ensuring that faulty nodes won't affect services.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

How the Feature Works

Figure 3-34 How query traffic isolation works



Here is how query traffic isolation works:

1. When a node in your cluster experiences issues, you configure it as either a low-priority node or an isolated node.
2. Query requests are broken down into per-shard queries.
3. Per-shard queries are preferentially allocated to healthy nodes, bypassing isolated nodes completely, and bypassing low-priority nodes when possible.
   - A per-shard query is allocated to a low-priority node only if this shard is not available elsewhere.
   - If the target shard of a query exists only on an isolated node, the query fails.

This way, CSS keeps query traffic away from problematic cluster nodes, ensuring query performance and service availability.

Constraints

- Query traffic isolation is supported only in OpenSearch 2.19.0 (image version 2.19.0_25.9.0_xxx or later) and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).
- By default, indexes prefixed with **.kibana** and **.opendistro** are exempted from query traffic isolation settings.

Logging In to OpenSearch Dashboards

Log in to OpenSearch Dashboards and go to the command execution page. OpenSearch clusters support multiple access methods. This topic uses OpenSearch Dashboards as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuring Query Traffic Isolation

Run the following command to configure low-priority nodes, isolated nodes, and the index whitelist (or allow-list).

```
PUT _cluster/settings
{
  "persistent": {
    "cluster.routing.css.preference.query.low_priority_nodes": "node1,node2",
    "cluster.routing.css.preference.query.isolated_nodes": "node3,node4",
    "cluster.routing.css.preference.query.allow_list": "index1,index2,test-index.*"
  }
}
```

Table 3-144 Parameter description

Parameter	Type	Default Value	Description
cluster.routing.css.preference.query.low_priority_nodes	String	Empty (no low-priority nodes)	Configure low-priority nodes. For per-shard query requests to indexes that are not on the whitelist, the system preferentially bypasses low-priority nodes. A per-shard query is allocated to a low-priority node only if this shard is not available elsewhere. When configuring multiple nodes, use a comma (**,**) to separate them.
cluster.routing.css.preference.query.isolated_nodes	String	Empty (no isolated nodes)	Configure isolated nodes. With the exception of whitelisted indexes, per-shard query requests bypass isolated nodes completely. If the target shard exists only on an isolated node, the query fails. When configuring multiple nodes, use a comma (**,**) to separate them.
cluster.routing.css.preference.query.allow_list	String	Empty	Configure indexes that are exempted from the above settings. Use a comma (**,**) to separate different index names. Regular expression matching is supported. By default, indexes prefixed with **.kibana** and **.opendistro** are exempted from query traffic isolation settings. This means the whitelist (or allow-list) will include the indexes you configure here, as well as these default ones.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3.11.7 Configuring Large Query Isolation

Large query isolation can be configured to manage queries that have high memory usage or take too long to complete. This helps improve the stability of OpenSearch clusters and prevent out-of-memory (OOM) exceptions.

As business grows, your OpenSearch clusters may face mounting query pressure. Some complex queries may occupy excessive node memory, triggering frequent garbage collection or even OOM exceptions, which may compromise cluster performance and stability. Large query isolation enables effective management of memory-intensive, time-consuming query requests, ensuring cluster stability. Large query isolation includes the following:

- Isolating large queries: Manages memory-intensive/time-consuming queries separately to avoid impacting other queries.
- Query cancellation based on a heap memory usage threshold: Cancels a large query in the isolation pool when the node heap memory usage reaches a predefined threshold.
- Global query timeout: Automatically cancels tasks when they last longer than a predefined timeout. This timeout applies globally.

How the Feature Works

- Defining large queries:
  - The system checks the memory usage of all ongoing queries and flags queries that exceed a predefined memory usage threshold as large queries.
  - The system periodically checks the execution duration of all ongoing queries and flags queries that exceed a predefined duration threshold as large queries.
- Query cancellation policies:
  - fair: Determines which query to cancel by considering both memory usage and execution duration.
  - mem-first: Cancels the query that has the highest memory usage.
  - time-first: Cancels the query that has lasted the longest.
- Native cancel API: OpenSearch's native cancel API can be used to cancel tasks, ensuring compatibility.

Constraints

Large query isolation is supported only for OpenSearch 2.19.0 or 3.4.0 (image version 3.4.0_26.6.0_xxx or later). The feature is **enabled by default** for these versions. The global timeout is disabled by default for large query isolation. You can enable and configure it via an API when necessary. Any change takes effect immediately.

Logging In to OpenSearch Dashboards

Log in to OpenSearch Dashboards and go to the command execution page. OpenSearch clusters support multiple access methods. This topic uses OpenSearch Dashboards as an example to describe the operation procedures.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Configuring Large Query Isolation

Large query isolation places large queries in an isolation pool, where they may be cancelled based on preset memory or duration thresholds. Large query isolation is **enabled by default**. You can modify this setting whenever necessary. Any change takes effect immediately.

1. Run the following command to enable or disable large query isolation:

```
PUT _cluster/settings
{
  "persistent": {
    "search.isolator.enabled": true
  }
}
```

Table 3-145 Setting large query isolation

Parameter	Type	Default Value	Description
search.isolator.enabled	Boolean	true	Whether to enable large query isolation. When enabled, large queries are managed separately from other normal queries. The value can be: • **true**: Enable large query isolation. • **false**: Disable large query isolation.

2. Run the following commands to configure thresholds for defining a large query task:

```
PUT _cluster/settings
{
  "persistent": {
    "search.isolator.memory.task.limit": "50MB",
    "search.isolator.time.management": "10s"
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-146 Parameters for configuring large query isolation thresholds

Parameter	Type	Default Value	Description
search.isolator.memory.task.limit	String	50MB	Large query memory threshold: When a query requests more memory than specified by this threshold, it is placed into an isolation pool. Value format: number + unit • Number: a natural number • Unit: B, K, KB, M, MB, G, GB, T, TB, P, or PB (case-insensitive) Minimum value: 0 (all queries are placed into the isolation pool) Maximum value: maximum node heap memory Lowering this value will cause more queries to be placed into the isolation pool, which will increase its memory usage. If you do lower this value, you should also increase the values of **search.isolator.memory.pool.limit** and **search.isolator.count.limit**, so that the isolation pool can hold more queries. This helps avoid triggering the circuit breaker mechanism due to resource exhaustion (for example, frequent query cancellation).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
search.isolator.time.management	String	10s	Large query execution duration threshold: When a query has lasted longer than specified by this threshold, it is placed into an isolation pool. Value format: number + unit • Number: a natural number • Unit: nanos (nanosecond), micros (microsecond), ms (millisecond), s (second), m (minute), h (hour), or d (day) Minimum value: 0 (all queries are placed into the isolation pool) Lowering this value will cause more queries to be placed into the isolation pool, which will increase its memory usage. If you do lower this value, you should also increase the values of **search.isolator.memory.pool.limit** and **search.isolator.count.limit**, so that the isolation pool can hold more queries. This helps avoid triggering the circuit breaker mechanism due to resource exhaustion (for example, frequent query cancellation).

3. Configure the resource usage thresholds for triggering query cancelation.

```
PUT _cluster/settings
{
  "persistent": {
    "search.isolator.memory.pool.limit": "50%",
    "search.isolator.count.limit": 1000,
    "search.isolator.memory.heap.limit": "90%"
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-147 Parameters for configuring query cancellation thresholds

Parameter	Type	Default Value	Description
search.isolator.memory.pool.limit	String	50%	Maximum memory usage of the isolation pool as a percentage of the maximum node heap memory. When the total memory usage of large queries in the isolation pool exceeds this limit, the system cancels one of the large queries in the isolation pool based on a predefined policy to free resources and prevent memory overflow. Value range: 0.0–100.0% If your cluster primarily handles large queries (high memory usage or long execution time), increase this value. Meanwhile, set **search.isolator.memory.task.limit** and **search.isolator.time.management** accordingly to control the number of queries placed into the isolation pool.
search.isolator.count.limit	Integer	1000	Maximum number of large queries allowed in the isolation pool. When this limit is reached, no more queries can be added to the isolation pool, preventing resource exhaustion. Value range: **10–50000** If your cluster primarily handles large queries (high memory usage or long execution time), increase this value. Meanwhile, set **search.isolator.memory.task.limit** and **search.isolator.time.management** accordingly to control the number of queries placed into the isolation pool.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
search.isolator.memory.heap.limit	String	90%	Node heap memory usage that triggers large query cancellation in the isolation pool. When this threshold is reached, the system cancels one of the large queries in the isolation pool based on a predefined policy to free resources and prevent memory overflow. Value range: 0.0–100.0% When **indices.breaker.total.use_real_memory** is enabled, this value must be lower than **indices.breaker.total.limit**. Otherwise, the native OpenSearch circuit breaker will always be triggered first. For details, see **Circuit breaker settings**. If you anticipate traffic peaks or surges, you can lower this value to have the isolation pool's circuit breaker triggered earlier, thus preventing heap memory overload.

4. Run the following command to set the query cancellation policy:

```
PUT _cluster/settings
{
  "persistent": {
    "search.isolator.strategy": "fair",
    "search.isolator.strategy.ratio": "0.5%"
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-148 Parameters for configuring a query cancellation policy

Parameter	Type	Default Value	Description
search.isolator.strategy	String	fair	Policy for selecting which query to cancel when query isolation is triggered. • fair (default): Determine which query to cancel by considering both memory usage and execution duration. If the difference between the memory usage of two candidate queries ≤ **maximum OpenSearch heap memory x fair policy threshold**, the query that has a longer execution duration will be cancelled; on the contrary, if the difference is greater than that, the more memory-intensive query will be cancelled instead. Maximum OpenSearch heap memory = **min(31, total node memory/2)** (GB). • mem-first: Cancels the query that has the highest memory usage. • time-first: Cancels the query that has lasted the longest. The large query isolation pool is checked every second until the heap memory is within the safe range.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
search.isolator.strategy.ratio	String	1%	Fair policy threshold, which is the ratio of the memory difference between two queries in the isolation pool to the maximum node heap memory. • When the memory usage difference between large queries in the isolation pool is small, the system preferentially cancels the query with the longest execution duration. • Otherwise, it cancels the query with the highest memory usage. This parameter is valid only when **search.isolator.strategy** is set to **fair**. Value range: 0.0–100.0% You are advised to use the default value. Adjust only if necessary and with caution.

5. Run the following command to set the maximum number of cancelled query records retained in the large query isolation log:

```
PUT _cluster/settings
{
  "persistent": {
    "search.isolator.log.count": "100"
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-149 Parameter description

Parameter	Type	Default Value	Description
search.isolator.log.count	Integer	100	The maximum number of cancelled query records retained in the large query isolation log. The large query isolation log records cancelled large queries for query performance analysis and optimization. Once this limit is exceeded, the system automatically deletes the oldest records to control the log's memory footprint. Value range: 0–5000 Setting this value to **0** disables the large query isolation log.

You can use the following APIs to query log information about cancelled queries:

- Query statistics about cancelled queries on all nodes:
  ```
  GET /_isolator_metrics
  ```
- Query statistics about cancelled queries on a specified node:
  ```
  GET /_isolator_metrics/{nodeId}
  ```
- Query details about cancelled queries on all nodes:
  ```
  GET /_isolator_metrics?detailed
  ```
- Query details about cancelled queries on a specified node:
  ```
  GET /_isolator_metrics/{nodeId}?detailed
  ```

Table 3-150 Parameter description

Parameter	Type	Default Value	Description
node_id	String	N/A	Specifies one or more cluster nodes. • Single node: Enter the node ID. • Multiple nodes: Enter multiple node IDs and use a comma (,) to separate them. You can run the following command to obtain node IDs: `GET _cat/nodes?s=n&h=n,id&v=true&full_id=true`

Example response:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
{
  "_nodes": {
    "total": 1,
    "successful": 1,
    "failed": 0
  },
  "cluster_name": "test",
  "nodes": {
    "CTqrZFXWTzmLonSZyNMKkQ": {
      "name": "test-ess-esn-1-1",
      "host": "172.16.101.116",
      "total_cancel": 0, //Total number of cancelled queries
      "isolator_cancel": 0,    //Number of queries cancelled because isolation pool thresholds were exceeded
      "out_of_time_cancel": 0    //Number of queries cancelled due to timeout
    }
  }
}
```

Configuring Global Query Timeout

When a global query timeout is configured, queries that exceed the specified duration are automatically cancelled, and the message "cancel cause by global time limit" is returned. This prevents long-running queries from consuming excessive resources. Global query timeout is **disabled by default**. You can modify this setting when necessary. Any change takes effect immediately.

Run the following command to enable and configure a global query timeout:

```
PUT _cluster/settings
{
  "persistent": {
    "search.isolator.time.enabled": true,
    "search.isolator.time.limit": "110s"
  }
}
```

Table 3-151 Parameters for setting the global query timeout

Parameter	Type	Default Value	Description
search.isolator.time.enabled	Boolean	false	Whether to enable a global query timeout. When enabled, queries are automatically cancelled when they last longer than a predefined timeout. The value can be: • true: Enable global query timeout. • false: Disable global query timeout.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
search.isolator.time.limit	String	120s	The value of the global query timeout. Value format: number + unit • Number: a natural number • Unit: nanos (nanosecond), micros (microsecond), ms (millisecond), s (second), m (minute), h (hour), or d (day) Minimum value: 0 (to cancel all queries)

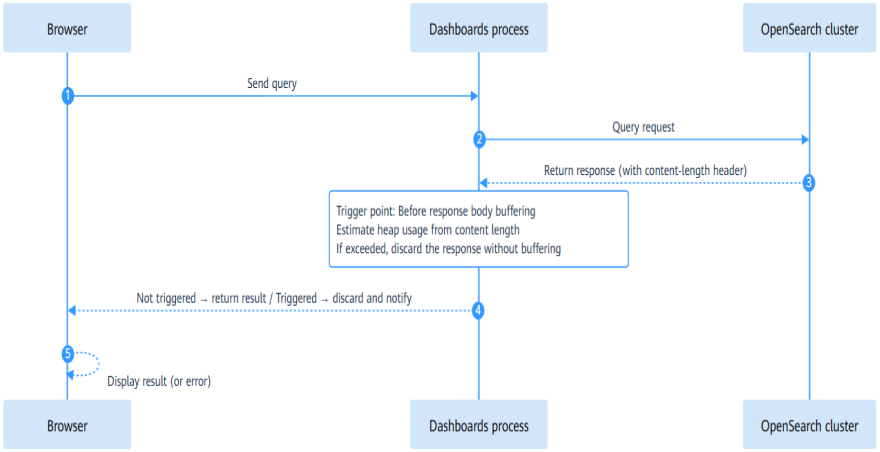
3.11.8 Configuring the OpenSearch Dashboards Query Response Circuit Breaker

During query operations, OpenSearch may return extremely large response bodies, such as full index exports or high-cardinality aggregations. When the OpenSearch Dashboards backend process buffers and parses these large responses, heap memory can be quickly exhausted, causing the process to crash due to OOM (Out of Memory), which can affect all users. CSS provides the query response circuit breaker feature. When the OpenSearch Dashboards backend process receives a response, before buffering the response body, the circuit breaker estimates the heap usage based on the **content-length** field in the response header. If the estimated heap usage exceeds the configured threshold, the response is discarded and an error is returned, preventing the process from crashing.

How the Feature Works

When the OpenSearch Dashboards backend process receives a query response, the circuit breaker determines whether to allow or block the response before buffering the response body.

Figure 3-35 Query process



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

The circuit breaker is triggered when either of the following conditions is met. After it is triggered, the response body is discarded directly:

- **Absolute response size limit**: The circuit breaker is triggered when **maxContentLength > 0** and **the response body's contentLength ≥ maxContentLength**. This indicates that the response body itself is too large and is rejected regardless of the current heap usage.
- **Heap usage limit**: The circuit breaker is triggered when **contentLength × heapMultiplier + heapUsed > heapSizeLimit × maxPercentage**. This indicates that buffering the response is estimated to exceed the configured heap usage limit.

Key parameters are as follows:

- **maxContentLength**: Absolute response body size limit. The circuit breaker is triggered when the response body reaches this size.
- **contentLength**: Size of the current response body, which is the value of the **content-length** field in the HTTP response header.
- **heapSizeLimit**: Maximum heap size of the process.
- **heapUsed**: Current heap usage of the process.
- **heapMultiplier**: Heap usage estimation multiplier. The default value is 4. The value of **contentLength** is multiplied by this value to estimate the heap usage after loading the response into memory. The default value of 4 is used because JSON data parsed into JavaScript objects typically consumes approximately 3 to 5 times more memory than the original text. A multiplier of 4 provides a balance between protection and resource utilization.
- **maxPercentage**: Maximum allowed heap usage in percentage.

NOTE

The query response circuit breaker applies to the OpenSearch Dashboards backend process and complements the OpenSearch cluster feature **Configuring Large Query Isolation**. The former protects the OpenSearch Dashboards process from oversized responses, while the latter manages large queries on the OpenSearch engine side.

Constraints

- The OpenSearch Dashboards query response circuit breaker is supported only for OpenSearch 2.19.0 or 3.4.0 clusters (image version x.x.x_26.6.0_xxx or later). It is **enabled by default** for clusters that meet these version requirements.
- For clusters with the security mode enabled, OpenSearch Dashboards query response circuit breaker parameters are visible and configurable only in the Global tenant.
- Changes made in **Advanced Settings** do not take effect immediately. They are synchronized periodically in the background and take effect within 1 minute.

Logging In to the Advanced Settings Page of OpenSearch Dashboards

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

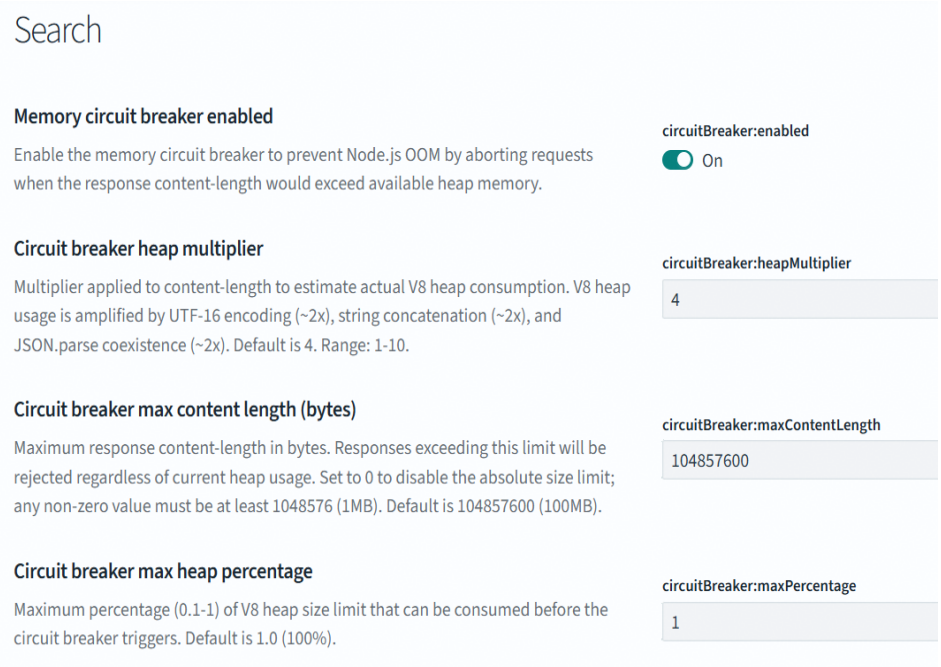
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. (Optional) For a security-mode cluster, switch to the Global tenant before configuring OpenSearch Dashboards query response circuit breaker parameters.

   On the OpenSearch Dashboards page, click the username in the upper right corner and choose **Switch tenants**. On the **Select your tenant** page, select **Global**, and click **Confirm**.
5. In the left navigation pane, choose **Management > Dashboards Management**, and select **Advanced settings**.

Configuring OpenSearch Dashboards Query Response Circuit Breaker Parameters

1. On the **Settings** page, locate the **Search** module and configure the following OpenSearch Dashboards query response circuit breaker parameters.

   Figure 3-36 Configuring OpenSearch Dashboards query response circuit breaker parameters



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

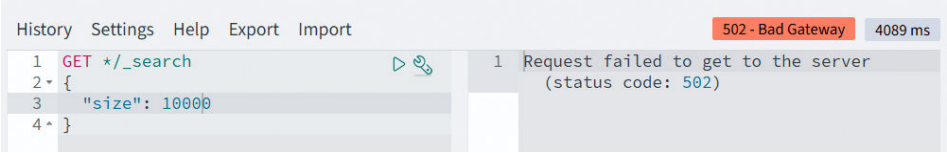
Table 3-152 Configuring OpenSearch Dashboards query response circuit breaker parameters

Parameter	Type	Default Value	Description
circuitBreaker:enabled	Boolean	On	Whether to enable the OpenSearch Dashboards query response circuit breaker. • **On**: Enables the circuit breaker. • **Off**: Disables the circuit breaker. After the circuit breaker is disabled, the OpenSearch Dashboards backend process no longer blocks oversized responses. Large query responses may cause the process to crash due to OOM, affecting all users. Disable this feature only when there is no risk of large queries in your workloads.
circuitBreaker:heapMultiplier	Integer	4	Heap usage estimation multiplier used to estimate heap consumption after a response is loaded into the heap. Value range: 1–10
circuitBreaker:maxContentLength	Integer	104857600	Absolute response body size limit. Value range: • 0 (no limit) • 1048576 (1 MB) to 1073741824 (1 GB) Unit: bytes
circuitBreaker:maxPercentage	Float	1.0	Maximum allowed heap usage in percentage. Value range: 0.1–1.0 1.0 indicates that the entire available heap can be used.

2. Click **Save changes** to save the changes.
3. Wait for about 1 minute. Then run a large query to verify that the circuit breaker takes effect.
   a. On Kibana, expand the menu in the upper-left corner, and choose **Dev Tools**.
   b. Run a large query. The request fails with HTTP status code 502.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-37 Large query



c. In the browser, press **F12** to launch Developer Tools, switch to the **Network** tab, and check **Response**. If **Memory circuit breaker triggered** is displayed, the query response has been blocked by the circuit breaker.

Figure 3-38 Query response



3.11.9 Configuring Automatic Retry for Failed Shard Allocation

As an OpenSearch cluster runs, shard allocation may fail due to factors such as high node load or a triggered circuit breaker. After a shard allocation failure, the system automatically retries the allocation. However, once the number of retries reaches the configured limit, the shard remains in the unassigned state and is no longer considered for reallocation. This may prevent read and write requests for the affected index from being processed properly. Normally, after this occurs, you can only retry the allocation manually by calling the Reroute API (**/_cluster/reroute?retry_failed=true**). This not only increases the operational burden, but also means that hours or even days may pass before manual intervention, preventing temporary failures from recovering automatically. CSS provides automatic retry for failed shard allocation. Once a shard reaches its maximum allocation failure limit, the system periodically retries allocation using an exponential backoff strategy, enabling self-healing without manual intervention.

How the Feature Works

After automatic retry for failed shard allocation is enabled, the system automatically retries shards whose number of allocation failures has reached the retry limit. The limit is controlled by **index.allocation.max_retries** and defaults to 5. The retry process is as follows:

1. Automatic retries start after the base delay.
2. If a retry succeeds, the shard is no longer tracked. If the retry fails, the backoff delay is doubled before the next retry.
3. Once the backoff delay reaches the configured maximum delay, it no longer increases. Subsequent retries are performed at the maximum delay.
4. The system periodically checks for failed shards that need to be retried at the configured polling interval. If any are found, they are retried.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-153 Exponential backoff example (base delay: 1m, maximum delay: 120m)

Retry Count	Backoff Delay	Description
0	1m	Shard is detected for the first time. The retry is allowed to proceed after the base delay.
1	2m	The first retry fails. The backoff delay is doubled.
2	4m	The second retry fails.
3	8m	The third retry fails.
4	16m	The fourth retry fails.
...	...	The delay continues to double until it reaches the maximum delay.
N	120m	The maximum delay is reached. Subsequent retries are performed at the maximum delay.

Constraints

Automatic retry for failed shard allocation is supported only for OpenSearch 2.19.0 or 3.4.0 (image version x.x.x_26.6.0_xxx or later). The feature is **enabled by default** for clusters that meet these version requirements.

Logging In to OpenSearch Dashboards

Log in to OpenSearch Dashboards and go to the command execution page. OpenSearch clusters support multiple access methods. This topic uses OpenSearch Dashboards as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Configuring Automatic Retry Parameters for Failed Shard Allocation

You can dynamically configure the parameters for automatic retry of failed shard allocation. The changes take effect immediately after configuration.

1. Enable or disable automatic retry for failed shard allocation.

```
PUT _cluster/settings
{
  "persistent": {
    "cluster.routing.allocation.retry_failed.auto_enabled": true
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  }
}
```

Table 3-154 Parameter for enabling or disabling automatic retry for failed shard allocation

Parameter	Type	Default Value	Description
cluster.routing.allocation.retry_failed.auto_enabled	Boolean	true	Enables or disables automatic retry for failed shard allocation. The value can be: • **true**: Enables automatic retry. • **false**: Disables automatic retry. After automatic retry is disabled, shards whose number of allocation failures reaches the retry limit remain in the unassigned state and must be retried manually using the Reroute API, which may result in prolonged service interruption. It is recommended that you disable this feature only when automatic recovery is not required for your cluster.

2. Configure the exponential backoff parameters that control the retry interval and frequency.

```
PUT _cluster/settings
{
  "persistent": {
    "cluster.routing.allocation.retry_failed.auto_base_delay": "1m",
    "cluster.routing.allocation.retry_failed.auto_max_delay": "120m",
    "cluster.routing.allocation.retry_failed.auto_poll_interval": "5m"
  }
}
```

Table 3-155 Exponential backoff parameters

Parameter	Type	Default Value	Description
cluster.routing.allocation.retry_failed.auto_base_delay	TimeValue	1m	Base delay for the backoff strategy. This specifies the initial wait time before the first automatic retry. After each failed retry, the delay doubles. Value range: 1m–1440m Format: number + unit. Supported units: ms (milliseconds), s (seconds), m (minutes), h (hours), and d (days).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
cluster.routing.allocation.retry_failed.auto_max_delay	TimeValue	120m	Maximum delay for the backoff strategy. Once the retry interval reaches this value, it no longer increases, preventing the backoff delay from growing indefinitely. Value range: 1m–1440m Format: number + unit. Supported units: ms (milliseconds), s (seconds), m (minutes), h (hours), and d (days).
cluster.routing.allocation.retry_failed.auto_poll_interval	TimeValue	5m	Polling interval. The system periodically checks for shards that have exceeded the retry limit, ensuring that shards are not missed when the cluster state remains stable. Value range: 1m–1440m Format: number + unit. Supported units: ms (milliseconds), s (seconds), m (minutes), h (hours), and d (days). It is recommended that you keep the default value. A smaller value increases the polling frequency and may have a slight impact on cluster performance. A larger value may delay the detection of shards that need to be retried.

3. (Optional) Verify the configuration. Query the current automatic retry configuration for failed shard allocation:

```
GET _cluster/settings?filter_path=**.retry_failed.auto_*
```

If the returned result contains the configured parameters and their values match the configured values, the configuration has taken effect.

FAQ

- **Q1: How do I manually retry failed shard allocation after disabling automatic retry?**

  After automatic retry is disabled, you can manually retry failed shard allocation by calling the Reroute API:

  ```
  POST /_cluster/reroute?retry_failed=true
  ```

- **Q2: Does automatic retry for failed shard allocation affect ongoing shard allocation?**

  No. Automatic retry only allocates shards that have reached the **index.allocation.max_retries** limit and are now in the unassigned state. It

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

does not trigger **Move** or **Balance** operations and therefore does not affect ongoing shard allocation.

3.11.10 Enhancing Data Ingestion Performance

In large-scale data ingestion scenarios, such as log analytics and real-time monitoring, the traditional OpenSearch routing mechanism can generate excessive internal forwarding requests as the number of index shards and the write throughput increase. This can lead to write rejections and increased long-tail latency. To address this issue, CSS provides several ways to enhance ingestion performance: bulk routing, bulk aggregation, text indexing acceleration, and merge optimization. They can help reduce performance overhead and significantly improve write stability and throughput in heavy-load scenarios.

How the Feature Works

To balance performance and functionality, CSS provides four ingestion performance enhancement solutions. The core idea is to eliminate or mitigate performance bottlenecks by reducing cross-node communication and accelerating index building.

Table 3-156 Ways to enhance data ingestion performance

Solution	Description	When to Use	Details
Bulk routing	Bulk routing reduces cross-node communication and request forwarding by the coordinator node, helping to prevent write rejections. How it works: Rather than routing each document individually based on its ID (the default behavior), this optimization forces multiple documents within the same bulk request to be routed together to the same shard or local node.	Enable this feature for large-scale clusters with a high number of shards. It is particularly effective when bulk request throughput is so intense that it exhausts the coordinator node's network I/O or CPU resources.	**Configuring Bulk Routing**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Solution	Description	When to Use	Details
Bulk aggregation	Bulk aggregation reduces system overhead and improves single-node capacity by minimizing resource lock contention during write operations. How it works: Instead of processing documents individually within the same bulk request, the system processes them as a unified batch, reducing memory allocations and lock contention.	Enable this feature for ultra-high concurrency write scenarios where throughput is critical. We recommend enabling it when monitoring indicates high system CPU usage or significant memory allocation pressure.	**Configuring Bulk Aggregation**
Text indexing acceleration	Text indexing acceleration significantly speeds up the indexing of text and keyword fields. How it works: Through an optimized indexing process, improved memory utilization, and accelerated tokenization, this feature improves the overall efficiency of text processing.	Enable this feature when you need high-speed index building with relatively simple tokenization logic. Examples of such scenarios include log auditing and full-text search, where large-volume text and numerous keyword fields are involved.	**Configuring Text Indexing Acceleration**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Solution	Description	When to Use	Details
Merge optimization	Merge optimization ensures write stability by preventing write throttling caused by slow segment merging. How it works: By dynamically increasing merging threads for index shards, this feature accelerates the merging of small segments and reclaims write cache space in a timely manner.	Enable this feature in high-frequency write scenarios, particularly when the write throughput increases drastically after enabling the three optimizations above and writes become unstable due to merge throttling.	**Configuring Merge Optimization**

Constraints

Ingestion performance enhancement is supported only in OpenSearch 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).

Logging In to OpenSearch Dashboards

Log in to OpenSearch Dashboards and go to the command execution page. OpenSearch clusters support multiple access methods. This topic uses OpenSearch Dashboards as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Configuring Bulk Routing

CAUTION

After bulk routing is enabled (that is, **index.bulk_routing** is set to **pack** or **local_pack**), documents are no longer routed by ID. This may affect routing-related functionality. For example, ID-based GET requests may fail because the shards cannot be located.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Enable bulk routing to alleviate request forwarding pressure for the node that receives requests.

```
PUT /{index_name}/_settings
{
  "index.bulk_routing": "local_pack"
}
```

Table 3-157 Parameters for bulk routing

Parameter	Type	Default Value	Description
index_name	String	N/A	Specifies one or more indexes. • Single index: Enter the index name, for example, **my_index**. • Multiple indexes: Enter multiple index names and use a comma (,) to separate them, for example, **my_index1,my_index2**. • Wildcard: Use the wildcard (*) to match multiple indexes. For example, **myindex*** indicates all indexes whose name starts with **myindex**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
index.bulk_routing	String	default	Specifies the internal routing policy for processing documents within bulk write requests. • **default**: Uses the default routing behavior. Each document within the same bulk request is hashed and routed separately based on its **_id**. This ensures even data distribution across all shards but can generate significant cross-node forwarding overhead during large-scale bulk writes. • **pack**: Routes all documents within the same bulk request to a single, randomly selected shard. This eliminates per-document hash calculations and cross-node forwarding, reducing request forwarding pressure on the coordinator node. • **local_pack** (recommended): First attempts to route all documents within the same bulk request to a local shard on the data node that received the bulk request. If that node does not host a shard of the target index, it falls back to the **pack** mode. This option is recommended because it achieves zero cross-node forwarding.

Configuring Bulk Aggregation

Enable bulk aggregation to enable the system to process documents within each bulk request as a unified batch, instead of processing each document individually. This significantly reduces CPU overhead and lock contention during ultra-high concurrency write operations.

```
PUT /{index_name}/_settings
{
  "index.aggr_perf_batch_size": "128"
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-158 Parameters for bulk aggregation

Parameter	Type	Default Value	Description
index.aggr_perf_batch_size	Integer	1	Specifies the maximum number of documents to process as a single batch. Minimum value: **1** (disables bulk aggregation so that documents will be processed individually) Recommended value: **128** Any value greater than 1 enables bulk aggregation. The system will internally reaggregate documents from bulk requests into batches up to this size. Larger batch sizes increase memory consumption per batch. Adjust this parameter based on your workload requirements and available system resources. The actual number of documents per batch is calculated as: MIN(Number of documents within the bulk request, aggr_perf_batch_size)

Configuring Text Indexing Acceleration

CAUTION

Text indexing acceleration cannot be enabled for indexes whose mapping contains nested fields. The system will not be able to start acceleration due to compatibility issues.

Enable text indexing acceleration to speed up the indexing of text and keyword fields.

```
PUT /{index_name}/_settings
{
  "index.native_speed_up": true,
  "index.native_analyzer": true
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-159 Parameters for text indexing acceleration

Parameter	Type	Default Value	Description
index.native_speed_up	Boolean	false	Whether to enable text indexing acceleration. The value can be: • **true**: Enables text indexing acceleration. This will significantly accelerate the indexing of text and keyword fields, optimize memory usage during indexing, and reduces the frequency of garbage collection (GC) caused by high-frequency writes. • **false**: Disables text indexing acceleration.
index.native_analyzer	Boolean	false	Whether to enable tokenization acceleration. This parameter is available only when **index.native_speed_up** is set to **true**, and it applies to text fields only. The value can be: • **true**: Enables tokenization acceleration. This will significantly reduce CPU time for tokenization. • **false**: Disables tokenization acceleration.

Configuring Merge Optimization

Increase the number of threads used for segment merging for index shards. This helps prevent write throttling caused by slow segment merging.

```
PUT /{index_name}/_settings
{
  "index.merge.scheduler.max_thread_count": 8
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-160 Parameters for segment merging

Parameter	Type	Default Value	Description
index.merge.scheduler.max_thread_count	Integer	4	Specifies the maximum number of threads used for segment merging in a single shard. Value range: 1 to (Number of CPU cores on the node/2) The recommended value is 8. It allows you to accelerate segment merging by leveraging the multi-core architecture of CPUs, eliminating write throttling caused by slow segment merging. However, you must ensure that your cluster nodes have sufficient CPU cores to run these threads.

3.11.11 Configuring Enhanced Aggregation

Low aggregation query performance often slows down analysis when large datasets are involved. CSS OpenSearch clusters leverage vectorization and optimized clustering to accelerate aggregation queries, enabling faster analytics and decision-making.

How the Feature Works

Enhanced aggregation works by pre-sorting and physically clustering data using carefully selected sorting and clustering keys upon ingestion, thereby minimizing data scanning and computational overheads during aggregation.

- Sorting key: A field (such as timestamps) used to physically order documents on disk, ensuring that documents with identical or similar sorting key values are stored contiguously.
- Clustering key: A subset of fields from the sorting key that groups related documents into contiguous physical blocks. This way, aggregations can process contiguous data blocks, instead of scattered documents.

For example, for an index that contains the **region**, **host**, and **value** fields, documents can be sorted by these three fields upon ingestion, and then clustered and stored based on the **region** and **host** fields. In this case, documents with the same **region** or **host** are stored together, and are then sorted by the **value** field within each cluster.

When an aggregation query is performed, data is queried by clustering key (**region** and **host**), or by the **value** field within each cluster. This improves query efficiency.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-39 An example of enhanced aggregation

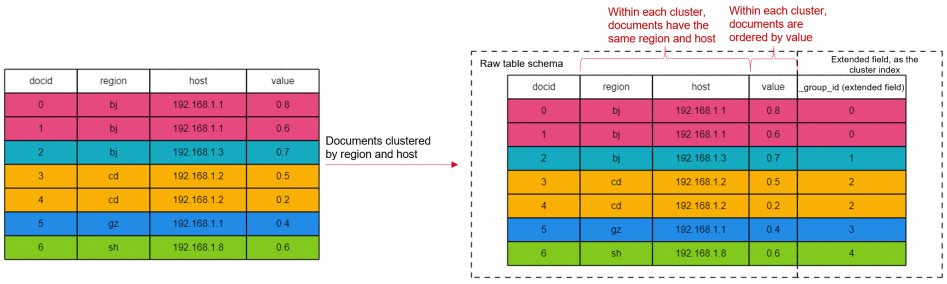


Table 3-161 Common scenarios for enhanced aggregation

Scenario	Description
Aggregation of low-cardinality fields	Aggregates fields that have a small number of unique values. In this case, grouping is used. One example is counting the number of orders per city. A clustering operation can easily achieve this purpose.
Aggregation of high-cardinality fields	Aggregates fields that have a large number of unique values. Histogram aggregation is typically used. One example is counting the hourly visits. A sorting key can be used to accelerate data aggregation by a specified scope or range.
Hybrid aggregation of low- and high-cardinality fields	First groups and aggregates low-cardinality fields (for example, orders per city) using a clustering key, and then creates a histogram for high-cardinality fields (for example, timestamps) using a sorting key. Multi-level clustering improves the efficiency of hybrid queries.

Constraints

Enhanced aggregation is supported only in OpenSearch 2.19.0 and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).

Logging In to OpenSearch Dashboards

Log in to OpenSearch Dashboards and go to the command execution page. OpenSearch clusters support multiple access methods. This topic uses OpenSearch Dashboards as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Aggregation of Low-Cardinality Fields

Generally, low-cardinality fields use grouping as a way of aggregation. With appropriate sorting keys, grouping prepares the data for batch processing.

For example, to aggregate two low-cardinality fields **city** and **product**, perform the following steps:

1. Run the following command to set the index **testindex**:

```
PUT testindex
{
  "mappings": {
    "properties": {
      "date": {
        "type": "date",
        "format": "yyyy-MM-dd"
      },
      "city": {
        "type": "keyword"
      },
      "product": {
        "type": "keyword"
      },
      "trade": {
        "type": "double"
      }
    }
  },
  "settings": {
    "index": {
      "search": {
        "turbo": {
          "enabled": "true"
        }
      },
      "sort": {
        "field": [
          "city",
          "product"
        ]
      },
      "cluster": {
        "field": [
          "city",
          "product"
        ]
      }
    }
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-162 Parameters for low-cardinality field aggregation

Parameter	Type	Default Value	Description
index.search.turbo.enabled	Boolean	false	Whether to enable enhanced aggregation. Normally, enhanced aggregation must be enabled where aggregations are used. The value can be: • **false**: Disable enhanced aggregation. • **true**: Enable enhanced aggregation.
index.sort.field	Array of strings	N/A	Specify sorting keys. Sorting keys are fields used to sequence or rank documents. You can specify one or multiple fields as sorting keys. When multiple fields are specified, they will apply in the sequence in which they are specified. Documents are first ranked by the first field, then the initial result set is ranked by the second field, and so on. Value range: The value must be fields contained in the index.
index.cluster.field	Array of strings	N/A	Specify clustering keys. Clustering keys determine which documents are collected into the same clusters. During an aggregation operation, documents in the same cluster can be processed in batches, significantly enhancing aggregation performance. Constraint: The clustering keys must be a subset of the sorting keys. Value range: The value must be fields contained in the index.

2. Run the following command to import sample data.

```
PUT /_bulk
{ "index": { "_index": "testindex", "_id": "1" } }
{ "date": "2025-01-01", "city": "cityA", "product": "books", "trade": 3000.0}
{ "index": { "_index": "testindex", "_id": "2" } }
{ "date": "2025-01-02", "city": "cityA", "product": "books", "trade": 1000.0}
{ "index": { "_index": "testindex", "_id": "3" } }
{ "date": "2025-01-01", "city": "cityA", "product": "bottles", "trade": 100.0}
{ "index": { "_index": "testindex", "_id": "4" } }
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
{ "date": "2025-01-02", "city": "cityA", "product": "bottles", "trade": 300.0}
{ "index": { "_index": "testindex", "_id": "5" } }
{ "date": "2025-01-01", "city": "cityB", "product": "books", "trade": 7000.0}
{ "index": { "_index": "testindex", "_id": "6" } }
{ "date": "2025-01-02", "city": "cityB", "product": "books", "trade": 1000.0}
```

3. Run the following query command to aggregate low-cardinality fields.

   For example, query the average **trade** value for each product in different cities.

```
POST testindex/_search
{
  "size": 0,
  "aggs": {
    "groupby_city": {
      "terms": {
        "field": "city"
      },
      "aggs": {
        "groupby_product": {
          "terms": {
            "field": "product"
          },
          "aggs": {
            "avg_trade": {
              "avg": {
                "field": "trade"
              }
            }
          }
        }
      }
    }
  }
}
```

Aggregation of High-Cardinality Fields

High-cardinality fields commonly use sorting keys for histogram aggregation, which facilitates data processing per range or scope.

For example, to aggregate the typical high-cardinality field **date**, perform the following steps:

1. Run the following command to set the index **testindex**:

```
PUT testindex
{
  "mappings": {
    "properties": {
      "date": {
        "type": "date",
        "format": "yyyy-MM-dd"
      },
      "score": {
        "type": "double"
      }
    }
  },
  "settings": {
    "index": {
      "search": {
        "turbo": {
          "enabled": "true"
        }
      },
      "sort": {
        "field": [
          "date"
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
      ]
    }
   }
  }
}
```

Table 3-163 Parameters for high-cardinality field aggregation

Parameter	Type	Default Value	Description
index.search.turbo.enabled	Boolean	false	Whether to enable enhanced aggregation. Normally, enhanced aggregation must be enabled where aggregations are used. The value can be: • **false**: Disable enhanced aggregation. • **true**: Enable enhanced aggregation.
index.sort.field	Array of strings	N/A	Specify sorting keys. Sorting keys are fields used to sequence or rank documents. You can specify one or multiple fields as sorting keys. When multiple fields are specified, they will apply in the sequence in which they are specified. Documents are first ranked by the first field, then the initial result set is ranked by the second field, and so on. Value range: The value must be fields contained in the index.

2. Run the following command to import sample data.

```
PUT /_bulk
{ "index": { "_index": "testindex", "_id": "1" } }
{ "date": "2025-01-01", "score": "12.0"}
{ "index": { "_index": "testindex", "_id": "2" } }
{ "date": "2025-01-02","score": "24.0"}
{ "index": { "_index": "testindex", "_id": "3" } }
{ "date": "2025-01-01","score": "53.0"}
{ "index": { "_index": "testindex", "_id": "4" } }
{ "date": "2025-01-02", "score": "22.0"}
{ "index": { "_index": "testindex", "_id": "5" } }
{ "date": "2025-01-01", "score": "99.0"}
{ "index": { "_index": "testindex", "_id": "6" } }
{ "date": "2025-01-02","score": "26.0"}
```

3. Run the following query command to aggregate high-cardinality fields.

   This query groups the **date** field using a histogram and then calculates the average score.

```
POST testindex/_search?pretty
{
  "size": 0,
  "aggs": {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  "groupbytime": {
    "date_histogram": {
      "field": "date",
      "calendar_interval": "day"
    },
    "aggs": {
      "avg_score": {
        "avg": {
          "field": "score"
        }
      }
    }
  }
}
```

Hybrid Aggregation of Low- and High-Cardinality Fields

Where low-cardinality and high-cardinality fields are mixed, first group and aggregate low-cardinality fields using a clustering key, then create a histogram for high-cardinality fields using a sorting key.

For example, to first group the low-cardinality field **city**, then group the low-cardinality field **product**, and then group the high-cardinality field **date** into a histogram, perform the following steps:

1. Run the following command to set the index **testindex**:

```
PUT testindex
{
  "mappings": {
    "properties": {
      "date": {
        "type": "date",
        "format": "yyyy-MM-dd"
      },
      "city": {
        "type": "keyword"
      },
      "product": {
        "type": "keyword"
      },
      "trade": {
        "type": "double"
      }
    }
  },
  "settings": {
    "index": {
      "search": {
        "turbo": {
          "enabled": "true"
        }
      },
      "sort": {
        "field": [
          "city",
          "product",
          "date"
        ]
      },
      "cluster": {
        "field": [
          "city",
          "product"
        ]
      }
    }
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  }
}
```

Table 3-164 Parameters for hybrid aggregation of low- and high-cardinality fields

Parameter	Type	Default Value	Description
index.search.turbo.enabled	Boolean	false	Whether to enable enhanced aggregation. Normally, enhanced aggregation must be enabled where aggregations are used. The value can be: • **false**: Disable enhanced aggregation. • **true**: Enable enhanced aggregation.
index.sort.field	Array of strings	N/A	Specify sorting keys. Sorting keys are fields used to sequence or rank documents. You can specify one or multiple fields as sorting keys. When multiple fields are specified, they will apply in the sequence in which they are specified. Documents are first ranked by the first field, then the initial result set is ranked by the second field, and so on. Constraint: High-cardinality fields must be among the sorting keys, and must follow the last low-cardinality field. Value range: The value must be fields contained in the index.
index.cluster.field	Array of strings	N/A	Specify clustering keys. Clustering keys determine which documents are collected into the same clusters. During an aggregation operation, documents in the same cluster can be processed in batches, significantly enhancing aggregation performance. Constraint: The clustering keys must be a subset of the sorting keys. Value range: The value must be fields contained in the index.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. Run the following command to import sample data.

```
PUT /_bulk
{ "index": { "_index": "testindex", "_id": "1" } }
{ "date": "2025-01-01", "city": "cityA", "product": "books", "trade": 3000.0}
{ "index": { "_index": "testindex", "_id": "2" } }
{ "date": "2025-01-02", "city": "cityA", "product": "books", "trade": 1000.0}
{ "index": { "_index": "testindex", "_id": "3" } }
{ "date": "2025-01-01", "city": "cityA", "product": "bottles", "trade": 100.0}
{ "index": { "_index": "testindex", "_id": "4" } }
{ "date": "2025-01-02", "city": "cityA", "product": "bottles", "trade": 300.0}
{ "index": { "_index": "testindex", "_id": "5" } }
{ "date": "2025-01-01", "city": "cityB", "product": "books", "trade": 7000.0}
{ "index": { "_index": "testindex", "_id": "6" } }
{ "date": "2025-01-02", "city": "cityB", "product": "books", "trade": 1000.0}
```

3. Run the following query command to perform a hybrid aggregation of low- and high-cardinality fields.

   For example, query the average **trade** value of each product in different cities on each day specified by the **date** field.

```
POST testindex/_search
{
  "size": 0,
  "aggs": {
    "groupby_region": {
      "terms": {
        "field": "city"
      },
      "aggs": {
        "groupby_host": {
          "terms": {
            "field": "product"
          },
          "aggs": {
            "groupby_timestamp": {
              "date_histogram": {
                "field": "date",
                "interval": "day"
              },
              "aggs": {
                "avg_score": {
                  "avg": {
                    "field": "trade"
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

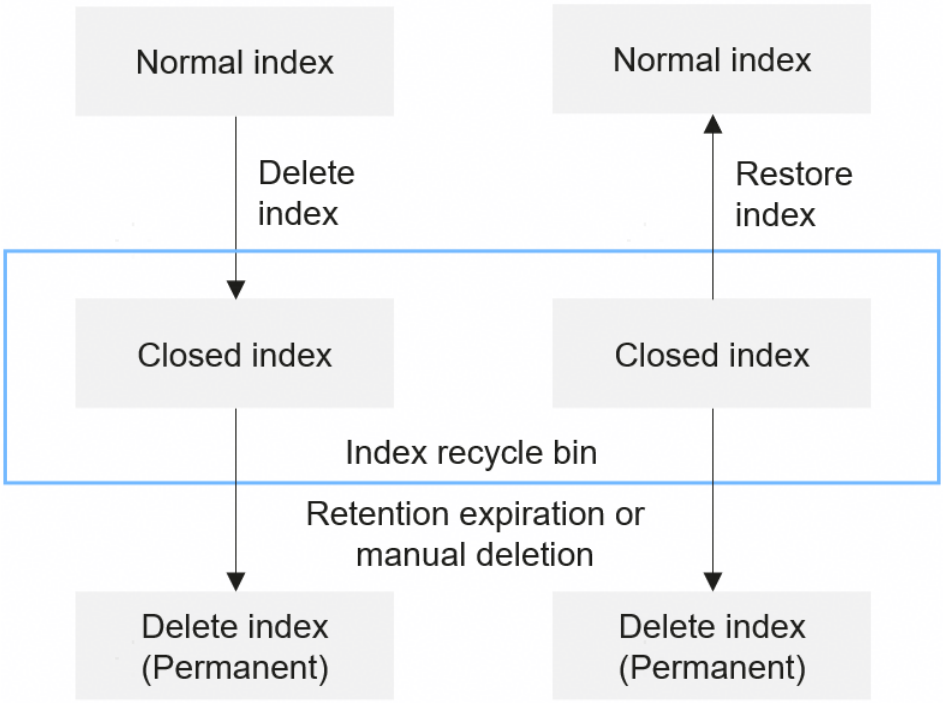
3.11.12 Enabling the Index Recycle Bin

By default, when indexes are deleted in an OpenSearch cluster, the index data is deleted permanently. To prevent data loss caused by accidental deletion, CSS provides an index recycle bin. When enabled, deleted indexes are temporarily stored in the recycle bin, allowing for recovery before they are permanently removed. This feature improves data reliability and operational security.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

How the Feature Works

Figure 3-40 How the index recycle bin works



The index recycle bin prevents permanent data loss caused by accidental index deletion. As shown in **Figure 3-40**, when the index recycle bin is enabled:

- When you delete an index, the index status changes to Closed, and the closed index is moved to the recycle bin. This may cause the cluster status to briefly turn Red. This status indicates there are unallocated shards in the cluster, but it does not affect service availability.
- Deleted indexes are retained in the recycle bin for a predefined period. When this retention period expires, the deleted index and its data are removed from the recycle bin and are thus permanently deleted. You can also manually delete them from the index recycle bin at any time, which is also permanent.
- To restore a deleted index, select it in the recycle bin, and click Restore. The index becomes Open again, and its shards are reinitialized. This process can also cause the cluster status to become Red temporarily.

Constraints

- The index recycle bin is supported only in OpenSearch 2.19.0 and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).
- Indexes in the recycle bin remain part of the cluster metadata. This is why you cannot create indexes with the same name as those in the recycle bin.

Logging In to OpenSearch Dashboards

Log in to OpenSearch Dashboards and go to the command execution page. OpenSearch clusters support multiple access methods. This topic uses OpenSearch Dashboards as an example to describe the operation procedures.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Enabling the Index Recycle Bin

1. Run the following command to enable the index recycle bin:

```
PUT _cluster/settings
{
  "persistent": {
    "index.trash.enabled": true
  }
}
```

Table 3-165 Parameters for enabling the index recycle bin

Parameter	Type	Default Value	Description
index.trash.enabled	Boolean	false	Whether to enable the index recycle bin. When the index recycle bin is enabled, deleting indexes will first move them to the recycle bin. The indexes will be permanently deleted only when they are cleared from the recycle bin. The value can be: • **true**: Enable the index recycle bin. • **false**: Disable the index recycle bin.
indices.trash.keep.time	String	1d	Index retention period in the recycle bin. Expired indexes are permanently removed via a scheduled deletion task. Value format: number + unit • Number: a natural number • Unit: nanos (nanosecond), micros (microsecond), ms (millisecond), s (second), m (minute), h (hour), or d (day) Minimum value: **1d**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
indices.trash.clean.interval	String	5m	Recycle bin cleanup interval. The system runs a periodic cleanup job to permanently remove expired indexes in the recycle bin. Value format: number + unit • Number: a natural number • Unit: nanos (nanosecond), micros (microsecond), ms (millisecond), s (second), m (minute), h (hour), or d (day) Minimum value: 5m

2. To manually delete specified indexes from the index recycle bin right away, instead of waiting for them to be removed periodically, run the following command:

```
DELETE {index_name}
```

Table 3-166 Parameter description

Parameter	Type	Default Value	Description
index_name	String	N/A	Specifies one or more indexes. • Single index: Enter the index name, for example, **my_index**. • Multiple indexes: Enter multiple index names and use a comma (,) to separate them, for example, **my_index1,my_index2**. • Wildcard: Use the wildcard (*) to match multiple indexes. For example, **myindex*** indicates all indexes whose name starts with **myindex**.

Viewing Indexes in the Recycle Bin

Run the following command to view indexes in the index recycle bin to confirm index status:

```
GET _cat/trash?v=true&s=index
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-167 Request parameters

Parameter	Type	Default Value	Description
v	Boolean	false	Whether the returned table will have a header. The value can be: • **true**: display the table header. • **false**: not to display the table header.
s	String	N/A	Index sorting field. The value can be: • **index**: Sorts indexes by index name. • **uuid**: Sorts indexes by index UUID. • **pri**: Sorts indexes by their number of primary shards. • **rep**: Sorts indexes by their number of replica shards. • **trash.ts**: Sorts indexes by the timestamp of when they were moved to the recycle bin. • **delete.time**: Sorts indexes by their remaining retention period in the recycle bin.
format	String	None specified (table format)	The format of the returned result. Value range: json (text), yaml (text), cbor (binary), or smile (binary) If none of these four formats is specified, the result will be returned in a table.
h	String	None specified (all)	Column names to be displayed in the returned table. If you wish to display only some of the columns, specify their names and separate them with a comma (,), for example, **h=index,uuid,delete.time**.

The following is an example of the output.

```
index    uuid                   pri rep      trash.ts delete.time
index1   CMD3FCLzTOyTg4RUekWNNA   1   1 1714465116615       23.6h
index1   6ATijuu6SfqamVI-WMyOKg   1   1 1714466233898       23.9h
```

NOTE

When **delete.time** becomes 0, an index will be permanently deleted.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Restoring an Index from the Recycle Bin

1. Run the following command to restore an index from the recycle bin:
   ```
   POST /trash/recover/{index_name}
   ```
2. Run the following command to query the index list and check whether the index has been successfully restored:
   ```
   GET _cat/indices?v=true
   ```

Emptying the Recycle Bin

WARNING

Emptying the recycle bin permanently deletes its data. Before emptying the recycle bin, make sure important data has been backed up.

1. Run the following command to empty the recycle bin:
   ```
   POST trash/empty
   ```
2. Run the following command to confirm that the recycle bin has been emptied:
   ```
   GET _cat/trash?v=true&s=index
   ```

3.11.13 Tracking Query Resource Consumption

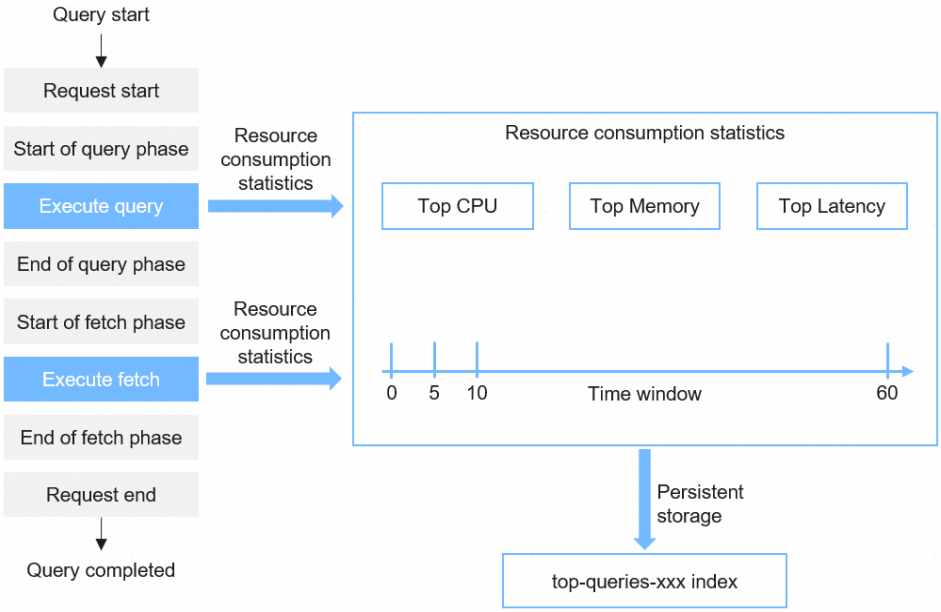
During routine maintenance, O&M engineers may need to identify and analyze top (resource-consuming) queries that are causing high resource consumption or performance issues in OpenSearch clusters. Typically, this requires calling OpenSearch APIs to retrieve a list of ongoing tasks and examining hot threads to determine which queries are causing excessive resource usage, such as high CPU consumption. The process can be complex and time-consuming. To improve O&M efficiency, CSS provides a query resource tracker. With this feature, O&M personnel can call an API to obtain top queries with the highest latency, CPU usage, or memory consumption, filter the queries by time range, and quickly identify problematic queries. This can significantly improve troubleshooting efficiency and accuracy.

How the Feature Works

The query resource tracker helps identify and optimize top resource-consuming queries, improving system performance and resource utilization. It is a useful tool in big data analytics and log processing scenarios.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-41 How the query resource tracker works



During query execution, the system records the resource consumption of each sub-phase (such as Query, Fetch, and Scroll), tracking metrics like CPU time and memory usage. The data is aggregated by a fixed time window (for example, 5 minutes). Queries with the highest resource consumption are recorded in a dedicated top queries index for analysis. A new top queries index is created daily to store these query resource statistics. The naming format is top-queries-xxx (where xxx indicates the date).

The top queries are ranked by the metric you specify, which can be CPU usage, memory usage, or latency. By default, the system ranks queries by latency.

Constraints

- The query resource tracker adds memory cache fields, which may impact cluster performance.
- The query resource tracker is supported only in OpenSearch 2.19.0 (image version 2.19.0_25.9.0_xxx or later) or 3.4.0 (image version 3.4.0_26.6.0_xxx or later). It is **enabled by default** for clusters that meet these version requirements.

Modifying Top Queries Monitoring Settings

OpenSearch allows you to modify top queries monitoring settings through OpenSearch Dashboards or APIs.

- **Method 1: Modify top queries monitoring settings through OpenSearch Dashboards.**
  a. Log in to the **CSS management console**.
  b. In the navigation pane on the left, choose **Clusters > OpenSearch**.
  c. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

d. In the left navigation pane, choose **OpenSearch Plugins > Query Insights**. The **Query insights** page is displayed.

e. Click the **Configuration** tab and modify relevant settings.

Table 3-168 Configuration parameters

Parameter	Description
Metric Type	Specify the metric type to set settings for. • Latency • CPU • Memory
Enabled	Enable or disable top N query monitoring by the selected metric type.
Value of N (count)	Specify the value of N. N is the number of queries to be monitored within the time window. Value range: 1 to 100 Default value: **10**
Window size	The size of the observation window. Monitoring data is aggregated and computed by a fixed window (for example, 5 minutes). Value range: 1m (1 minute), 5m (5 minutes), 10m (10 minutes), 30m (30 minutes), or *x*h (*x* hours, where *x* ranges from 1 to 24) Default value: **5m**
Group By	Specifies the group by type. The value can be: • **None** (default): No grouping. • **Similarity**: Group queries by feature similarity. Within each time window, only the first query is displayed for each group. The default matching mode is the complete query structure (structure + field + type). For more information, see **An Introduction to Query Grouping**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Exporter	Whether to persist data to a top queries index. • **Local Index** (default): Persist query resource monitoring data to the top-queries-xxx index. • **None**: Not to store data to the top-queries-xxx index. Instead, only the data from the most recent two windows is kept in the memory, while older data is discarded.
Delete After (days)	Retention period of the top-queries-*xxx* index, in days. Value range: 1 to 180 Default value: **7**

f. Click **Save** to save the changes.

- **Method 2: Modify top queries monitoring settings through an API.**

  Run the following command to modify the top queries monitoring settings as needed:

```
PUT _cluster/settings
{
  "persistent": {
    "search.insights.top_queries.cpu.enabled": true,
    "search.insights.top_queries.cpu.window_size": "10m",
    "search.insights.top_queries.cpu.top_n_size": 20,
    "search.insights.top_queries.exporter.delete_after_days": 8,
    "search.insights.top_queries.group_by": "none"
  }
}
```

Table 3-169 Parameter description

Parameter	Type	Default Value	Description
search.insights.top_queries.<metric>.enabled	Boolean	true	Whether to enable top query monitoring by the specified metric. Supported metrics: **latency**, **cpu** (CPU usage), or **memory** (memory usage). The value can be: • **true**: Enable top query monitoring. • **false**: Disable top query monitoring.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
search.insights.top_queries.<metric>.window_size	String	5m	The size of the observation window. Monitoring data is aggregated and computed by a fixed window (for example, 5 minutes). Supported metrics: **latency**, **cpu** (CPU usage), or **memory** (memory usage). Value range: 1m (1 minute), 5m (5 minutes), 10m (10 minutes), 30m (30 minutes), or *x*h (*x* hours, where *x* ranges from 1 to 24)
search.insights.top_queries.<metric>.top_n_size	Integer	10	Number of top N queries monitored in each time window. For example, if this parameter is set to 20, only the top 20 queries are monitored in each time window. Supported metrics: **latency**, **cpu** (CPU usage), or **memory** (memory usage). Value range: 1 to 100
search.insights.top_queries.exporter.delete_after_days	Integer	7	Retention period of the top-queries-*xxx* index. For example, if this parameter is set to 8, the index is retained for eight days. Value range: 1 to 180 Unit: days
search.insights.top_queries.group_by	String	none	Whether to enable top query grouping. Only OpenSearch 2.19.0 clusters (image version 2.19.0_25.9.0_xxx or later) support this parameter. The value can be: • **none**: No grouping. • **similarity**: Group queries by feature similarity. Within each time window, only the first query is displayed for each group. For more information, see **An Introduction to Query Grouping**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
search.insights.top_queries.grouping.group_by	String	none	Whether to enable top query grouping. Only OpenSearch 3.4.0 clusters (image version 3.4.0_26.6.0_xxx or later) support this parameter. The value can be: • **none**: No grouping. • **similarity**: Group queries by feature similarity. Within each time window, only the first query is displayed for each group. For more information, see **An Introduction to Query Grouping**.
search.insights.top_queries.grouping.attributes.field_name	Boolean	true	Whether to use query field names for query grouping. This item takes effect only when **search.insights.top_queries.group_by** is set to **similarity**. The value can be: • **true**: Group queries by field name. • **false**: Ignore query field names.
search.insights.top_queries.grouping.attributes.field_type	Boolean	true	Whether to use query field types for query grouping. This item takes effect only when **search.insights.top_queries.group_by** is set to **similarity**. The value can be: • **true**: Group queries by field type. • **false**: Ignore query field types.

Obtaining Top Queries

OpenSearch allows you to obtain top queries through OpenSearch Dashboards or an API.

- **Method 1: Obtain top queries through OpenSearch Dashboards.**
  a. Log in to the **CSS management console**.
  b. In the navigation pane on the left, choose **Clusters > OpenSearch**.
  c. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

d. In the left navigation pane, choose **OpenSearch Plugins > Query Insights**. The **Query insights** page is displayed.

e. On the **Top N queries** tab, check the resource usage of top N queries to evaluate query performance and resource utilization efficiency. Click **Id** to go to the query details page. You can view the query statement, execution time, and resource usage.

- **Method 2: Obtain top queries through an API.**

  The following is an example of the command that you can run to obtain top queries by a specified metric and time range:

```
GET _insights/top_queries?type=cpu&from=2025-12-02T00:00:00.000Z&to=2025-12-02T17:00:00.000Z
```

Table 3-170 Request parameters

Parameter	Type	Default Value	Description
type	String	latency	The metric by which top queries are identified. Value range: **latency**, **cpu** (CPU usage), or **memory** (memory usage).
from	String	Null (obtain the top N queries in the last two windows)	Start time of the query time range. **from** and **to** must both be configured. Value format: YYYY-MM-DDTHH:mm:ss.SSSZ (timestamp in ISO 8601 format)
to	String	Null (obtain the top N queries in the last two windows)	End time of the query time range. **from** and **to** must both be configured. Value format: YYYY-MM-DDTHH:mm:ss.SSSZ (timestamp in ISO 8601 format)

Example response:

```
{
  "top_queries": [
    {
      "timestamp": 1764662136273,                          //Timestamp of the query.
      "date": "2025-12-02 07:55:36Z",                      //Time when the query was executed.
      "id": "4a5b4b1e-b502-4621-a5c1-09b8d9a1b81c",        //Unique ID of the query.
      "task_resource_usages": [
        {
          "action": "indices:data/read/search[phase/query]", // CPU and memory consumption of the query phase on node FB2ixw4IQCuXzCR83GT5Yg
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        "taskId": 1877927,
        "parentTaskId": 111295,
        "nodeId": "FB2ixw4IQCuXzCR83GT5Yg",
        "taskResourceUsage": {
          "cpu_time_in_nanos": 3017078,
          "memory_in_bytes": 102360
        }
      },
      {
        "action": "indices:data/read/search[phase/query]", // CPU and memory consumption of the
query phase on node FB2ixw4IQCuXzCR83GT5Yg
        "taskId": 1877926,
        "parentTaskId": 111295,
        "nodeId": "FB2ixw4IQCuXzCR83GT5Yg",
        "taskResourceUsage": {
          "cpu_time_in_nanos": 5618940,
          "memory_in_bytes": 271680
        }
      },
      {
        "action": "indices:data/read/search[phase/query]", // CPU and memory consumption of the
query phase on node 2ICOHICoSS26YeQu5PIrlg
        "taskId": 107710,
        "parentTaskId": 111295,
        "nodeId": "2ICOHICoSS26YeQu5PIrlg",
        "taskResourceUsage": {
          "cpu_time_in_nanos": 8703914,
          "memory_in_bytes": 501560
        }
      },
      {
        "action": "indices:data/read/search[phase/fetch/id]", // CPU and memory consumption of the
fetch phase on node FB2ixw4IQCuXzCR83GT5Yg
        "taskId": 1877928,
        "parentTaskId": 111295,
        "nodeId": "FB2ixw4IQCuXzCR83GT5Yg",
        "taskResourceUsage": {
          "cpu_time_in_nanos": 424055,
          "memory_in_bytes": 59000
        }
      },
      {
        "action": "indices:data/read/search[phase/fetch/id]", // CPU and memory consumption of the
fetch phase on node 2ICOHICoSS26YeQu5PIrlg
        "taskId": 107711,
        "parentTaskId": 111295,
        "nodeId": "2ICOHICoSS26YeQu5PIrlg",
        "taskResourceUsage": {
          "cpu_time_in_nanos": 1279677,
          "memory_in_bytes": 337504
        }
      },
      {
        "action": "indices:data/read/search",              // CPU and memory consumption on the access
node gzsjh_47SjCe6QFs9pKjEg during the query start phase
        "taskId": 111295,
        "parentTaskId": -1,
        "nodeId": "gzsjh_47SjCe6QFs9pKjEg",
        "taskResourceUsage": {
          "cpu_time_in_nanos": 297268,
          "memory_in_bytes": 8632
        }
      }
    ],
    "source": {                                    //Specific query statement
      "query": {
        "match": {
          "message": {
            "query": "http",
```

```
            "operator": "OR",
            "prefix_length": 0,
            "max_expansions": 50,
            "fuzzy_transpositions": true,
            "lenient": false,
            "zero_terms_query": "NONE",
            "auto_generate_synonyms_phrase_query": true,
            "boost": 1
          }
        }
      }
    },
    "indices": [                                  //Queried index
      "log1"
    ],
    "total_shards": 3,                            //Total number of shards queried
    "phase_latency_map": {                        //Time consumed during each phase
      "expand": 0,                                //Time consumed in the expand phase
      "query": 16,                                //Time consumed in the query phase
      "fetch": 3                                  //Time consumed in the fetch phase
    },
    "labels": {},
    "group_by": "NONE",                           //Query grouping type
    "node_id": "gzsjh_47SjCe6QFs9pKjEg",          //ID of the node that received the request
    "search_type": "query_then_fetch",            //Query type
    "measurements": {                             //Metrics used
      "memory": {                                 //Memory consumption
        "number": 1280736,
        "count": 1,
        "aggregationType": "NONE"
      },
      "latency": {                                //Latency
        "number": 20,
        "count": 1,
        "aggregationType": "NONE"
      },
      "cpu": {                                    // CPU consumption
        "number": 19340932,
        "count": 1,
        "aggregationType": "NONE"
      }
    }
  }
 ]
}
```

Obtaining the Resource Consumption of the Query Resource Tracker

Run the following command to obtain the resource consumption of the query resource tracker:

```
GET _insights/health_stats
```

Example response:

```
{
  "QDyhJ8Q6Td2acc3KGQ43bQ" : {
    "ThreadPoolInfo" : {
      "query_insights_executor" : {  //Dedicated queue for the plugin, used to execute top query analysis tasks
        "type" : "scaling",
        "core" : 1,
        "max" : 1,
        "keep_alive" : "5m",
        "queue_size" : -1
      }
    },
    "QueryRecordsQueueSize" : 0,  //Number of unprocessed tasks in the query_insights_executor queue
    "TopQueriesHealthStats" : {
      "latency" : {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
      "TopQueriesHeapSize" : 0,  //Memory occupied by top queries statistics
      "QueryGroupCount_Total" : 0,  //Number of groups kept in the memory when grouping is enabled
      "QueryGroupCount_MaxHeap" : 0  //Memory used to store groups when grouping is enabled
    },
    "cpu" : {
      "TopQueriesHeapSize" : 0,
      "QueryGroupCount_Total" : 0,
      "QueryGroupCount_MaxHeap" : 0
    },
    "memory" : {
      "TopQueriesHeapSize" : 0,
      "QueryGroupCount_Total" : 0,
      "QueryGroupCount_MaxHeap" : 0
    }
  },
  "FieldTypeCacheStats" : { //Cache statistics. When query grouping is enabled, the field mapping is cached
to avoid repeated mapping lookups.
    "size_in_bytes" : 0,
    "entry_count" : 0,
    "evictions" : 0,
    "hit_count" : 0,
    "miss_count" : 0
  }
 }
}
```

An Introduction to Query Grouping

When a single query continuously consumes excessive resources, it can monopolize the TopN statistics, obscuring other resource-intensive queries. Query grouping addresses this by aggregating similar queries through pattern matching, ensuring that only the first query from each group appears in the TopN results.

Table 3-171 Grouping modes

Grouping Mode	Description
Complete query structure (structure + field + type)	Precisely matches the field type.
Query structure only	Considers the query structure only, while ignoring field names and types.
Query structure + field only	Considers the query structure and field names only, while ignoring field types.
Query structure + type only	Considers the query structure and field types only, while ignoring field names.

NOTE

Choose field type matching if the field types in your database remain relatively constant. The system will cache field types to improve grouping efficiency.

Example

The mapping of an index is as follows:

```
"mappings": {
 "properties": {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
    "field1": {
      "type": "keyword"
    },
    "field2": {
      "type": "text"
    },
    "field3": {
      "type": "text"
    },
    "field4": {
      "type": "long"
    }
  }
}
```

Perform the following query on the index:

```
{
  "query": {
    "bool": {
      "must": [
        {
          "term": {
            "field1": "example_value"
          }
        }
      ],
      "filter": [
        {
          "match": {
            "field2": "search_text"
          }
        },
        {
          "range": {
            "field4": {
              "gte": 1,
              "lte": 100
            }
          }
        }
      ],
      "should": [
        {
          "regexp": {
            "field3": ".*"
          }
        }
      ]
    }
  }
}
```

Table 3-172 provides examples that help you understand how each grouping mode behaves.

Table 3-172 Behavior of different grouping modes

Grouping Mode	Pattern
Complete query structure (structure + field + type)	bool [] must: term [field1, keyword] filter: match [field2, text] range [field4, long] should: regexp [field3, text]

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Grouping Mode	Pattern
Query structure only	bool must: term filter: match range should: regexp
Query structure + field only	bool [] must: term [field1] filter: match [field2] range [field4] should: regexp [field3]
Query structure + type only	bool [] must: term [keyword] filter: match [text] range [long] should: regexp [text]

3.12 Monitoring

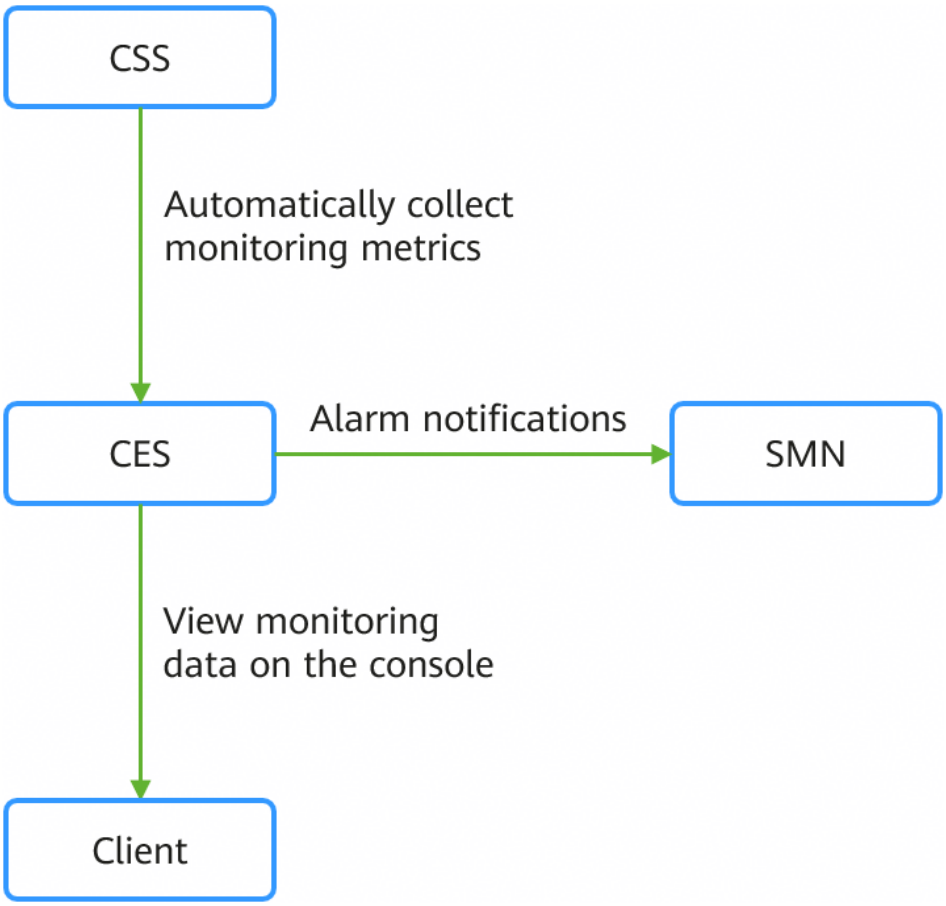
3.12.1 Using Cloud Eye to Monitor Clusters

In production environments, OpenSearch clusters run continuously, and issues such as full disks, JVM out-of-memory errors, and query queue backlogs often emerge before causing failures. However, relying only on cluster logs makes it difficult to detect these issues early and analyze metric trends visually. To address this, CSS integrates with Cloud Eye to collect core OpenSearch cluster metrics, such as disk usage, CPU usage, JVM heap usage, and read/write latency, in real time. Cloud Eye provides visual monitoring dashboards to display metric trends and sends alarm notifications through Simple Message Notification (SMN) when metrics exceed configured thresholds. This helps operations personnel shift from post-failure troubleshooting to proactive issue prevention.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

How the Feature Works

Figure 3-42 Monitoring data flow



The monitoring data flow is as follows:

1. Clusters automatically report monitoring data to Cloud Eye.
2. You can set alarm rules for CSS on Cloud Eye. You can configure alarm notifications via SMN.
3. You can then view monitoring metrics on the console.

Cloud Eye provides the following monitoring dimensions for OpenSearch clusters.

Table 3-173 Monitoring dimensions

Dimension	Description	Applicable Scenario
Cluster	Aggregates metrics at the OpenSearch cluster level, such as cluster health status and cluster-level disk usage.	Overall health checks and cluster-level alerting

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Dimension	Description	Applicable Scenario
Node	Collects resource metrics at the node level, such as the CPU usage and JVM heap usage of individual nodes.	Identifying hot nodes and troubleshooting node-level performance issues

Recommended Monitoring Metrics

Table 3-174 Recommended metrics

Dimension	Recommended Metrics	Description
Disk and cluster health (routine monitoring)	Disk usage, cluster health	Full disks are a primary cause of cluster unavailability. You are advised to set an alarm threshold of 85% for disk usage.
CPU and JVM	Average JVM heap usage, maximum JVM heap usage, average CPU usage, and maximum CPU usage	Continuously high JVM heap usage (> 75%) may cause frequent GC and affect query performance.
Read/Write latency and throughput	Average indexing latency, average indexing rate, average query latency, and average query rate	Continuously increasing indexing/query latency usually indicates excessive cluster load or the presence of slow queries.
Write/Search queues and rejected requests	Total queued tasks in the Write queue, total queued tasks in the Search queue, total rejected tasks in the Write queue, and total rejected tasks in the Search queue	Rejected requests indicate that the cluster is overloaded and requests are being rejected. You are advised to configure an alarm rule that triggers when the number of rejected requests exceeds 0.

Constraints

- Runtime duration: The cluster must run for more than 10 minutes before Cloud Eye can collect valid monitoring data.
- Operation permissions: The account used for operations must have the required Cloud Eye permissions. Otherwise, you cannot view monitoring information or configure alarm rules.

Prerequisites

The target OpenSearch cluster is available.

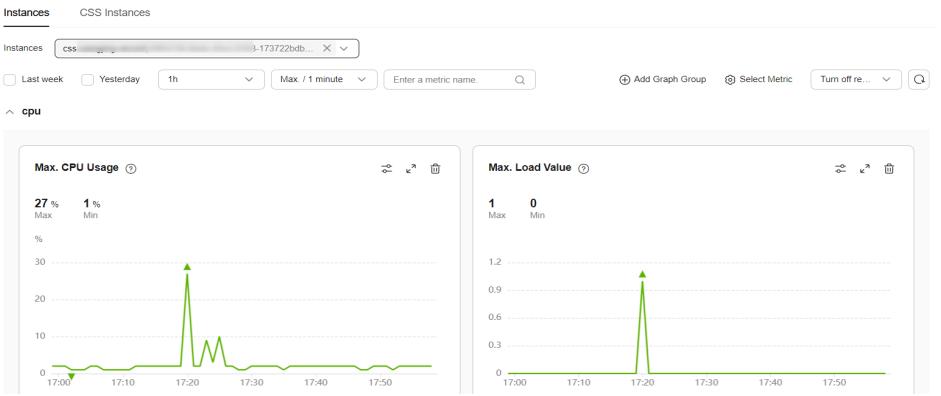
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Viewing Monitoring Metrics

You can view the real-time monitoring metrics of clusters and nodes on the built-in monitoring page of the CSS console. You can compare data across multiple clusters and perform period-over-period analysis.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Monitoring Metrics** in the **Operation** column.
4. View cluster-level or node-level monitoring data as needed.

Figure 3-43 Viewing monitoring metrics



- **Viewing cluster-level monitoring metrics**

  On the monitoring metrics page, select the **Instances** tab.

  - **Instances**: Select the cluster whose monitoring data you want to view. Data for the current cluster is displayed by default. You can select multiple clusters for horizontal comparison.
  - You can set the monitoring time range and view period-over-period comparisons for the same cluster to identify periodic patterns or abnormal deviations.
  - **Create Graph Group**: Add custom groups on top of the default groups to group related metrics for focused analysis. When you move the pointer over a group name on the left, the icons for changing the group name and deleting the group are displayed on the right. You can modify the group.
  - **Select Metric**: In the **Select Metric** dialog box, you can select both **All Metrics** (single line charts) and **Top N Metrics** (comparison of the top N nodes by metric value).

- **Viewing node-level monitoring metrics**

  On the monitoring metrics page, select the **CSS Instances** tab.

  - **CSS Instances**: Select the cluster nodes whose monitoring data you want to view. Data for the first node in the current cluster is displayed by default. You can select multiple nodes for horizontal comparison.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- You can also set monitoring periods so that you can compare metrics across different periods for the same cluster.
- **Create Graph Group**: Add custom groups on top of the default groups to group related metrics for focused analysis. When you move the pointer over a group name on the left, the icons for changing the group name and deleting the group are displayed on the right. You can modify the group.
- Select Metric: In the displayed **Select Metric** dialog box, set **All Metrics** (per-metric charts).

5. (Optional) To view monitoring dashboards, see **Method 1: Monitoring Trends on the Monitoring Page (Monitoring Dashboards)**.

   Monitoring dashboards display time-series trend charts for multiple metrics in one place, allowing you to quickly understand the overall cluster status during routine checks.

Configuring Alarm Rules

You can configure alarm thresholds and notification methods for monitoring metrics. When metrics exceed configured thresholds, notifications are automatically sent to specified destinations, enabling operations personnel to shift from reactive troubleshooting to proactive monitoring.

1. Log in to the **Cloud Eye console**.
2. In the navigation pane on the left, choose **Alarm Management** > **Rule Settings**.
3. In the **Resource Type** column, search for alarm rules whose **Resource Type** is **Cloud Search Service**, and check whether these rules meet your requirements.

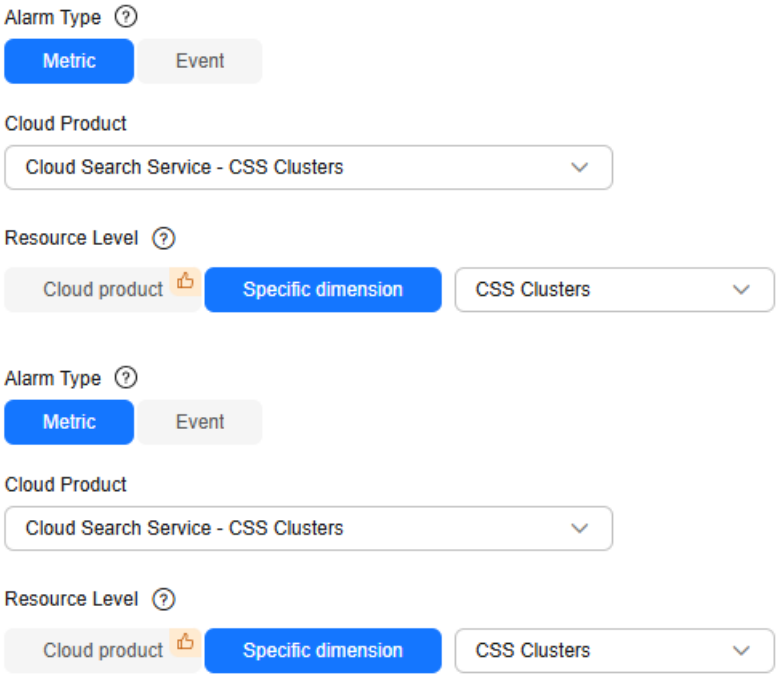
   To create new alarm rules, see **Creating an Alarm Rule and Notifications**. **Table 3-175** describes key parameters. Configure other parameters based on your requirements.

Table 3-175 Key parameters for configuring an alarm rule

Parameter	Description
Alarm Type	Select **Metric**.
Cloud Product	Select **Cloud Search Service - CSS Clusters**.
Resource Level	Select **Specific dimension**. In the drop-down list box on the right, select a specific dimension. • **CSS Clusters**: Creates an alarm rule at the cluster level. • **CSS Clusters - CSS Instances**: Creates an alarm rule at the individual node level.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-44 Configuring a cluster/node-level alarm rule



4. After an alarm rule is created, it is displayed in the alarm rule list. The rule **Status** is **Enabled**.

3.12.2 Configuring Index-Level Monitoring

When maintaining large-scale OpenSearch clusters, operations teams frequently encounter a critical visibility gap: while cluster-wide metrics (such as CPU and memory) indicate a healthy state, specific workloads or services experience long read/write latency. Traditional cluster-level monitoring often relies on aggregated averages and fails to identify the specific indexes causing the problem. As a result, root-cause analysis is often time-consuming and labor-intensive. CSS supports index-level monitoring. It automatically collects real-time read/write traffic, latency, and storage changes for each index, and visualizes the data using a built-in OpenSearch Dashboards dashboard. This helps you quickly identify abnormal indexes without writing any complex script. You can then make necessary adjustments to optimize performance and ensure service stability.

How the Feature Works

Index monitoring tracks the status and performance of each index in an OpenSearch cluster in real time, helping operations teams promptly detect and resolve performance issues. The feature works as follows:

1. Collection: A background task periodically (every 10 seconds by default, configurable) collects statistics about each target index, including the number of documents, storage size, and shard status.
2. Storage: The collected data is written to a dedicated monitoring index named **monitoring-eye-css-[yyyy-mm-dd]**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3. Visualization: OpenSearch Dashboards reads data from the monitoring index and visualizes it through preset dashboards or custom visualizations, enabling time series analysis, trend comparison, and fault detection.

Constraints

- Index-level monitoring is supported only in OpenSearch 2.19.0 and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).
- Do not create regular indexes with their names starting with **monitoring-eye-css-***. Doing so may interfere with the collection of index monitoring data.
- Do not delete the **monitoring-eye-css-[yyyy-mm-dd]** index or its associated index pattern. Otherwise, index monitoring data will be unavailable.

Logging In to OpenSearch Dashboards

Log in to OpenSearch Dashboards and go to the command execution page. OpenSearch clusters support multiple access methods. This topic uses OpenSearch Dashboards as an example to describe the operation procedures.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters** > **OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Enabling Index Monitoring

- Run the following command to enable monitoring for all indexes in the current cluster:

```
PUT _cluster/settings
{
  "persistent": {
    "css.monitoring.index.enabled": "true"
  }
}
```

- Run the following command to enable monitoring for specified high-priority indexes:

```
PUT _cluster/settings
{
  "persistent": {
    "css.monitoring.index.enabled": "true",
    "css.monitoring.index.interval": "30s",
    "css.monitoring.index.indices": ["index_name"],
    "css.monitoring.history.duration": "3d"
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-176 Parameters for enabling index monitoring

Parameter	Type	Default Value	Description
css.monitoring.index.enabled	Boolean	false	To enable or disable index monitoring. The value can be: • **true**: To enable index monitoring. • **false**: To disable index monitoring.
css.monitoring.index.indices	String	* (indicating that all indexes will be monitored)	The names of the indexes you want to monitor. • Single index: Enter the index name, for example, **my_index**. • Multiple indexes: Enter multiple index names and use a comma (,) to separate them, for example, **my_index1,my_index2**. • Wildcard: Use the wildcard (*) to match multiple indexes. For example, **myindex*** matches all indexes whose name starts with **myindex**. By default, all indexes are monitored. If you have a large number of indexes, consider monitoring only high-priority ones.
css.monitoring.index.interval	Time	10s	Index monitoring data collection interval. Value format: number + unit • Number: a natural number • Unit: nanos (nanosecond), micros (microsecond), ms (millisecond), s (second), m (minute), h (hour), or d (day) Minimum value: **1s** Set an appropriate data collection interval based on your monitoring needs and cluster load. Collecting monitoring data too frequently can impact performance.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
css.monitoring.history.duration	Time	7d	Retention duration of index monitoring data, that is, the data in the **monitoring-eye-css-[yyyy-mm-dd]** index. Upon expiration, data will be deleted automatically and permanently. Value format: number + unit • Number: a natural number • Unit: nanos (nanosecond), micros (microsecond), ms (millisecond), s (second), m (minute), h (hour), or d (day) Minimum value: **1d** Set an appropriate data retention duration to balance your monitoring needs with storage costs. A longer retention duration increases storage costs.

Checking the Read and Write Traffic of Indexes

When index monitoring is enabled for a cluster, you can use an API to query the real-time read and write traffic of indexes.

CAUTION

System indexes (those starting with **.**) contain internal management and maintenance information. To ensure system security, their read and write traffic cannot be queried.

- Run the following command to query the real-time read and write traffic information of all indexes in the cluster:
  ```
  GET /_cat/monitoring
  ```
- Run the following command to query the real-time read and write traffic of specified indexes in the cluster:
  ```
  GET /_cat/monitoring/{index_name}
  ```
- Run the following command to query the read/write traffic of indexes in a specified period:
  ```
  GET _cat/monitoring?begin=1650099461000
  GET _cat/monitoring?begin=2022-04-16T08:57:41
  GET _cat/monitoring?begin=2022-04-16T08:57:41&end=2022-04-17T08:57:41
  ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-177 Parameters for querying the read and write traffic of indexes

Parameter	Type	Default Value	Description
index_name	String	N/A	Specifies one or more indexes. • Single index: Enter the index name, for example, **my_index**. • Multiple indexes: Enter multiple index names and use a comma (,) to separate them, for example, **my_index1,my_index2**. • Wildcard: Use the wildcard (*) to match multiple indexes. For example, **myindex*** matches all indexes whose name starts with **myindex**.
begin	String	Last 5 minutes	Start time of the monitoring period. Use the UTC time. Supported formats: • Standard date and time format: yyyy-MM-dd or yyyy-MM-ddTHH:mm:ss • A timestamp in milliseconds
end	String	Current time	End time of the monitoring period. Use the UTC time. Supported formats: • Standard date and time format: yyyy-MM-dd or yyyy-MM-ddTHH:mm:ss • A timestamp in milliseconds

Example response:

```
index   begin                end                  status pri rep init unassign docs.count docs.deleted store.size
pri.store.size delete.rate indexing.rate search.rate
test 2022-03-25T09:46:53.765Z 2022-03-25T09:51:43.767Z yellow  1   1 0    1     9        0      5.9kb
5.9kb          0/s           0/s          0/s
```

Table 3-178 Response parameters

Parameter	Description
index	Index name.
begin	Start time of the monitoring data you queried.
end	End time of the monitoring data you queried.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
status	Index status within the queried period.
pri	The number of primary shards within the queried period.
rep	The number of replica shards within the queried period.
init	The number of initialized indexes within the queried period.
unassign	The number of unallocated indexes within the queried period.
docs.count	The number of documents within the queried period.
docs.deleted	The number of deleted documents within the queried period.
store.size	Index storage size within the queried period.
pri.store.size	Size of a primary index shard within the queried period.
delete.rate	Number of indexes deleted per second within the queried period.
indexing.rate	Number of indexes written per second within the queried period.
search.rate	Number of indexes queried per second within the queried period.

Visualizing and Analyzing Index Monitoring Data in OpenSearch Dashboards

OpenSearch Dashboards enables efficient analysis of index monitoring data, including time series analysis, trend comparison, and fault detection.

CSS provides a pre-built dashboard that allows you to quickly view index monitoring information. You can also create custom visualizations in OpenSearch Dashboards for more flexible monitoring.

- Check index monitoring results in the pre-built dashboard.
  a. In the OpenSearch Dashboards console, click the menu button in the upper-left corner, and click **Dashboard**.
  b. Find and click **[Monitoring] Index monitoring Dashboard** to view the index monitoring results.

     The pre-built dashboard displays the number of read and write operations per second in the cluster, as well as the top 10 indexes by read and write requests.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-45 Preconfigured dashboards

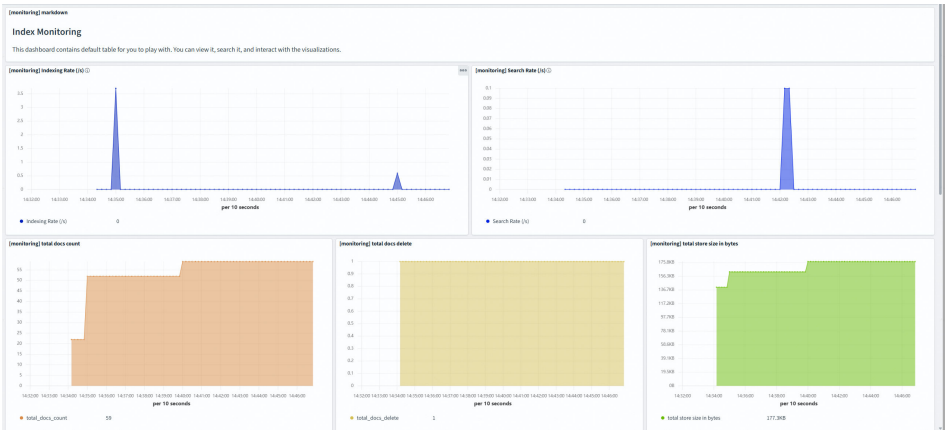


Table 3-179 Preconfigured charts

Chart Name	Description
[monitoring] markdown	Markdown chart, which briefly describes the dashboard content.
[monitoring] Indexing Rate (/s)	Number of documents written to a cluster per second.
[monitoring] Search Rate (/s)	Average number of queries per second in a cluster.
[monitoring] indexing rate of index for top10	Top 10 indexes with the most documents written per second.
[monitoring] search rate of index for top10	Top 10 indexes with the most queries per second.
[monitoring] total docs count	Total number of documents in a cluster.
[monitoring] total docs delete	Total number of deleted documents in a cluster.
[monitoring] total store size in bytes	Total storage space occupied by documents in a cluster.
[monitoring] indices store_size for top10	Top 10 indexes that occupy the largest storage space.
[monitoring] indices docs_count for top10	Top 10 indexes that store the largest number of documents.
[monitoring] indexing time in millis of index for top10(ms)	Top 10 indexes with the longest document write latency in a unit time (ms).
[monitoring] search query time in millis of index for top10(ms)	Top 10 indexes with the longest index query time in a unit time (ms).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Chart Name	Description
[monitoring] segment count of index for top10	Top 10 indexes with the largest number of index segments.
[monitoring] segment memory in bytes of index for top10	Top 10 indexes with the largest heap memory usage of index segments.

- Check index monitoring results in custom visualizations.

  The following procedure describes how to create a custom visualization to monitor the changes in the document counts of indexes.

  a. Click **Visualize** in the upper-left menu of the OpenSearch Dashboards console.

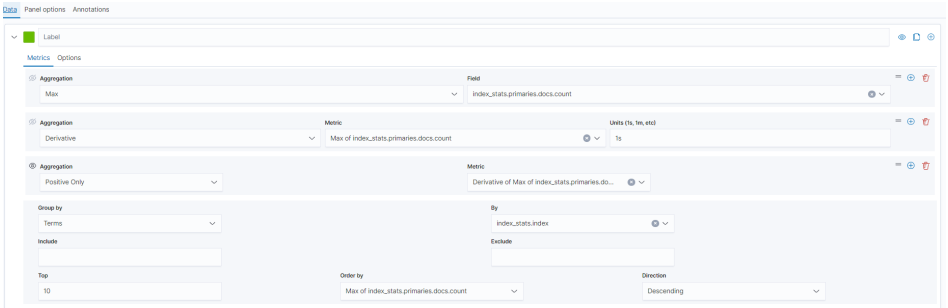
  b. Click **Create visualization** and select **TSVB**.

  c. Set the chart parameters.

  On the **Data** tab of the page shown in **Figure 3-46**, set the parameters as needed.

  - Select **Max** for **Aggregation**, and select **index_stats.primaries.docs.count** in **Field**, indicating the number of documents in a primary shard.
  - Select **Derivative** from **Aggregation** to indicate differences between aggregation buckets. Set **Units** to **1s** to visualize network rates as "per second".
  - Set **Aggregation** to **Positive Only** to prevent negative numbers after resetting.
  - To show statistics by index, set **Group by** to **Terms** and **By** to **index_stats.index**. Statistics will be grouped by index name.

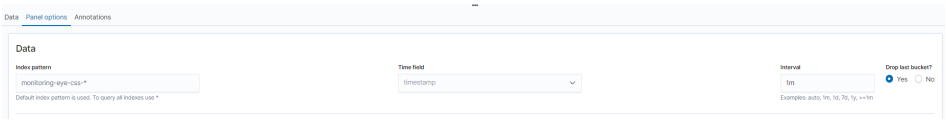
  Figure 3-46 TSVB page

  To view data in different time segments, set the aggregation interval, or the displayed data will be incomplete. On the page shown in **Figure 3-47**, select the **Panel options** tab, and set **Interval** to **1m** or **30m** to adjust the granularity of **timestamp**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-47 Setting the interval



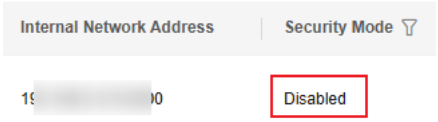
FAQ: What Do I Do If Index Monitoring Charts Are Not Displayed?

Symptom: The cluster's preset dashboards and visualizations are not displayed.

Solution:

1. In the cluster list, check whether the security mode is enabled for the cluster.

   Figure 3-48 Checking whether the security mode is enabled

   - If **Security Mode** is **Enabled**, switch to the Global space.

     On OpenSearch Dashboards, click the username in the upper right corner and choose **Switch tenants**. On the **Select your tenant** page, switch the space and click **Confirm**. Then, check whether the charts are now displayed properly. If they are still not, go to the next step.
   - If **Security Mode** is **Disabled**, go to the next step.
2. In the cluster list, check whether the image version of the cluster is 2.19.0_25.9.0_xxx or later.
   - If the cluster image version is 2.19.0_25.9.0_xxx or later, go to the next step to restart index monitoring.
   - If the cluster image version is earlier than 2.19.0_25.9.0_xxx, restart all cluster nodes and then restart index monitoring. For details, see **Restarting a Cluster**.
3. Restart index monitoring.

   Set **css.monitoring.index.enabled** to **false** and then to **true**. Wait for approximately 5 minutes. The index monitoring charts will be automatically displayed.

   ```
   PUT _cluster/settings
   {
     "persistent": {
       "css.monitoring.index.enabled": "true"
     }
   }
   ```

   If the problem persists, contact technical support.

3.12.3 Supported Metrics

You can use Cloud Eye to monitor CSS cluster metrics in real time and quickly handle exceptions.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Overview

This section describes CSS metrics that can be monitored by Cloud Eye as well as their namespaces and dimensions. You can query metrics and alerts generated by CSS on the **Cloud Eye console** or using an **API**.

NOTE

Cloud Eye can monitor dimensions (or objects) nested to a maximum depth of four levels (levels 0 to 3). 3 is the deepest level. For example, if the monitored dimension of a metric is **cluster_id,instance_id**, **cluster_id** indicates level 0 and **instance_id** indicates level 1.

Namespaces

SYS.ES

Cluster Monitoring Metrics

NOTE

Accumulated value: The value is accumulated from when a node is started. After the node is restarted, the value is reset to zero and starts accumulating again.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-180 OpenSearch cluster monitoring metrics

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
status	Cluster Health Status	Health status of the monitored object • **0**: The cluster is 100% available. • **1**: All primary shards are allocated, but some replicas are missing. There is an availability risk. Pay close attention to the cluster status. • **2**: Data is missing and the cluster cannot work properly. • **3**: Failed to obtain the cluster status.	0, 1, 2, 3	N/A	N/A	cluster_id	1 minute
disk_util	Max. Disk Usage	Maximum disk usage of nodes in a CSS cluster	0~100	%	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
max_jvm_heap_usage	Max. JVM Heap Usage	Maximum JVM heap usage of nodes in a CSS cluster.	0~100	%	N/A	cluster_id	1 minute
max_jvm_young_gc_time	Max. JVM Young GC Duration	Maximum accumulated JVM Young GC duration of nodes in a CSS cluster.	≥ 0	ms	N/A	cluster_id	1 minute
max_jvm_young_gc_count	Max. JVM Young GC Count	Maximum accumulated JVM Young GC count of nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_jvm_old_gc_time	Max. JVM Old GC Duration	Maximum accumulated JVM Old GC duration of nodes in a CSS cluster.	≥ 0	ms	N/A	cluster_id	1 minute
max_jvm_old_gc_count	Max. JVM Old GC Count	Maximum accumulated JVM Old GC count of nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
total_fs_size	Total Size of File Systems	Total size of file systems in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute
free_fs_size	Available Size of File Systems	Available size of file systems in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute
max_cpu_usage	Max. CPU Usage	Maximum node CPU usage in a CSS cluster.	0~100	%	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
max_cpu_time_of_jvm_process	Max. CPU Time of JVM Process	Maximum accumulated CPU usage duration of node JVM processes in a CSS cluster.	≥ 0	ms	N/A	cluster_id	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.	≥ 0	byte	1024	cluster_id	1 minute
max_current_opened_http_count	Current Max. Opened HTTP Connections	Maximum number of HTTP connections that are currently open for nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_total_opened_http_count	Total Max. Opened HTTP Connections	Maximum number of HTTP connections that were open for nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
indices_count	Indexes	Number of indexes in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute
total_shards_count	Shards	Number of shards in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute
primary_shards_count	Primary Shards	Number of primary shards in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
docs_count	Documents	Number of documents in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute
docs_deleted_count	Deleted Documents	Number of documents deleted in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute
nodes_count	Nodes	Number of nodes in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute
data_nodes_count	Data Nodes	The number of nodes that function as data nodes in a CSS cluster. This includes cold data nodes.	≥ 0	Count	N/A	cluster_id	1 minute
master_nodes_count	Master Nodes	The number of nodes that function as master nodes in a CSS cluster. Without dedicated master nodes, it indicates the total number of non-client nodes. With dedicated master nodes, it indicates the number of dedicated master nodes.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
ingest_nodes_count	Ingest Nodes	The number of nodes that function as ingest nodes in a CSS cluster. (Ingest nodes preprocess, clean, and transform data before indexing.)	≥ 0	Count	N/A	cluster_id	1 minute
max_load_average	Max. Node Load	Maximum number of average queued tasks per minute on nodes in a cluster	≥ 0	Count	N/A	cluster_id	1 minute
avg_cpu_usage	Avg. CPU Usage	Average node CPU usage in a CSS cluster.	0~100	%	N/A	cluster_id	1 minute
avg_load_average	Avg. Node Load	Average number of queued tasks per minute on nodes in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute
avg_jvm_heap_usage	Avg. JVM Heap Usage	Average node JVM heap usage in a CSS cluster	0~100	%	N/A	cluster_id	1 minute
max_open_file_descriptors	Max. Open File Descriptors	Maximum number of node file descriptors that are currently open in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_open_file_descriptors	Avg. Open File Descriptors	Average number of node file descriptors that are currently open in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_max_file_descriptors	Max. Allowed File Descriptors	Maximum number of allowed node file descriptors in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_open_file_descriptors	Open File Descriptors	Number of node file descriptors that are currently open in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_write_queue	Tasks in Write Queue	Number of tasks in queue for the write thread pool.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_search_queue	Tasks in Search Queue	Total number of tasks in queue for the search thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_force_merge_queue	Tasks in Force Merge Queue	Total number of tasks in queue for the force merge thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_write_rejected	Rejected Tasks in Write Queue	Total number of tasks rejected by the write thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_search_rejected	Rejected Tasks in Search Queue	Total number of tasks rejected by the search thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_force_merge_rejected	Rejected Tasks in Force Merge Queue	Total number of tasks rejected by the force merge thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_write_queue	Max. Tasks in Write Queue	Maximum number of tasks in queue for the write thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_search_queue	Max. Tasks in Search Queue	Maximum number of tasks in queue for the search thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
max_thread_pool_force_merge_queue	Max. Tasks in Force Merge Queue	Maximum number of tasks in queue for the force merge thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_write_threads	Size of Write Thread Pool	Total size of the write thread pools on nodes in the CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_search_threads	Size of Search Thread Pool	Total size of the search thread pools on nodes in the CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_force_merge_threads	Size of Force Merge Thread Pool	Total size of the force merge thread pools on nodes in the CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_write_queue	Avg. Tasks in Write Queue	Average number of tasks in queue for the write thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_thread_pool_search_queue	Avg. Tasks in Search Queue	Average number of tasks in queue for the search thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_force_merge_queue	Avg. Tasks in Force Merge Queue	Average number of tasks in queue for the force merge thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_search_threads	Avg. Size of Search Thread Pool	Average size of the search thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_write_threads	Avg. Size of Write Thread Pool	Average size of the write thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_force_merge_threads	Avg. Size of Force Merge Thread Pool	Average size of the force merge thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_write_rejected	Avg. Rejected Tasks in Write Queue	Average number of tasks rejected by the write thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_flush_queue	Tasks in Flush Queue	Total number of tasks in queue for the flush thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_flush_rejected	Rejected Tasks in Flush Queue	Total number of tasks rejected by the flush thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_flush_queue	Max. Tasks in Flush Queue	Maximum number of tasks in queue for the flush thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_flush_threads	Size of Flush Thread Pool	Total size of the flush thread pools on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_flush_queue	Avg. Tasks in Flush Queue	Average number of tasks in queue for the flush thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_flush_threads	Avg. Size of Flush Thread Pool	Average size of the flush thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_generic_queue	Tasks in Generic Queue	Total number of tasks in queue for the generic thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_generic_rejected	Rejected Tasks in Generic Queue	Total number of tasks rejected by the generic thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_generic_queue	Max. Tasks in Generic Queue	Maximum number of tasks in queue for the generic thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_generic_threads	Size of Generic Thread Pool	Total size of the generic thread pools on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_generic_queue	Avg. Tasks in Generic Queue	Average number of tasks in queue for the generic thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_generic_threads	Avg. Size of Generic Thread Pool	Average size of the generic thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_management_queue	Tasks in Management Queue	Total number of tasks in queue for the management thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_management_rejected	Rejected Tasks in Management Queue	Total number of tasks rejected by the management thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_management_queue	Max. Tasks in Management Queue	Maximum number of tasks in queue for the management thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_management_threads	Size of Management Thread Pool	Total size of the management thread pools on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_management_queue	Avg. Tasks in Management Queue	Average number of tasks in queue for the management thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_thread_pool_management_threads	Avg. Size of Management Thread Pool	Average size of the management thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_refresh_queue	Tasks in Refresh Queue	Total number of tasks in queue for the refresh thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_refresh_rejected	Rejected Tasks in Refresh Queue	Total number of tasks rejected by the refresh thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_refresh_queue	Max. Tasks in Refresh Queue	Maximum number of tasks in queue for the refresh thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_refresh_threads	Size of Refresh Thread Pool	Total size of the refresh thread pools on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_refresh_queue	Avg. Tasks in Refresh Queue	Average number of tasks in queue for the refresh thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_thread_pool_refresh_threads	Avg. Size of Refresh Thread Pool	Average size of the refresh thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_searcher_queue	Tasks in OBS Searcher Queue	Total number of tasks in queue for the OBS Searcher thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_searcher_rejected	Rejected Tasks in OBS Searcher Queue	Total number of tasks rejected by the OBS Searcher thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_obs_searcher_queue	Max. Tasks in OBS Searcher Queue	Maximum number of tasks in queue for the OBS Searcher thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_searcher_threads	Size of OBS Searcher Thread Pool	Total size of the OBS Searcher thread pools on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_thread_pool_obs_searcher_queue	Avg. Tasks in OBS Searcher Queue	Average number of tasks in queue for the OBS Searcher thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_obs_searcher_threads	Avg. Size of OBS Searcher Thread Pool	Average size of the OBS Searcher thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_queue	Tasks in OBS Queue	Total number of tasks in queue for the OBS thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_rejected	Rejected Tasks in OBS Queue	Total number of tasks rejected by the OBS thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_obs_queue	Max. Tasks in OBS Queue	Maximum number of tasks in queue for the OBS thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_threads	Size of OBS Thread Pool	Total size of the OBS thread pools on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_thread_pool_obs_queue	Avg. Tasks in OBS Queue	Average number of tasks in queue for the OBS thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_obs_threads	Avg. Size of OBS Thread Pool	Average size of the OBS thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_upload_queue	Tasks in OBS Upload Queue	Total number of tasks in queue for the OBS Upload thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_upload_rejected	Rejected Tasks in OBS Upload Queue	Total number of tasks rejected by the OBS Upload thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_obs_upload_queue	Max. Tasks in OBS Upload Queue	Maximum number of tasks in queue for the OBS Upload thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_obs_upload_threads	Size of OBS Upload Thread Pool	Total size of the OBS Upload thread pools on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_obs_upload_queue	Avg. Tasks in OBS Upload Queue	Average number of tasks in queue for the OBS Upload thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_obs_upload_threads	Avg. Size of OBS Upload Thread Pool	Average size of the OBS Upload thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_download_queue	Tasks in OBS Download Queue	Total number of tasks in queue for the OBS Download thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_download_rejected	Rejected Tasks in OBS Download Queue	Total number of tasks rejected by the OBS Download thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
max_thread_pool_obs_download_queue	Max. Tasks in OBS Download Queue	Maximum number of tasks in queue for the OBS Download thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_download_threads	Size of OBS Download Thread Pool	Total size of the OBS Download thread pools on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_obs_download_queue	Avg. Tasks in OBS Download Queue	Average number of tasks in queue for the OBS Download thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_obs_download_threads	Avg. Size of OBS Download Thread Pool	Average size of the OBS Download thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
min_free_fs_size	Min. Available Storage Space	Minimum available storage space of nodes in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_jvm_old_gc_count	Avg. GCs of Old-Generation JVM	Average number of old-generation garbage collections of nodes in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute
avg_jvm_old_gc_time	Avg. GC Duration of Old-Generation JVM	Average old-generation garbage collection duration of nodes in a CSS cluster	≥ 0	ms	N/A	cluster_id	1 minute
avg_jvm_young_gc_count	Avg. GCs of Young-Generation JVM	Average number of young-generation garbage collections of nodes in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute
avg_jvm_young_gc_time	Avg. GC Duration of Young-Generation JVM	Average young-generation garbage collection duration of nodes in a CSS cluster	≥ 0	ms	N/A	cluster_id	1 minute
avg_max_file_descriptors	Avg. Maximum Allowed File Descriptors	Average value of the maximum number of allowed file descriptors on each node in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_mem_free_in_bytes	Avg. Available Memory	Average unused memory capacity of nodes in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute
avg_mem_free_percent	Avg. Available Memory Percentage	Average percentage of unused memory of nodes in a CSS cluster.	0~100	%	N/A	cluster_id	1 minute
avg_mem_used_in_bytes	Avg. Used Memory	Average used memory of nodes in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute
avg_mem_used_percent	Avg. Used Memory Percentage	Average percentage of used memory of nodes in a CSS cluster.	0~100	%	N/A	cluster_id	1 minute
max_mem_free_in_bytes	Max. Available Memory	Maximum unused memory of nodes in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute
max_mem_free_percent	Max. Available Memory Percentage	Maximum percentage of unused memory of nodes in a CSS cluster.	0~100	%	N/A	cluster_id	1 minute
max_mem_used_in_bytes	Max. Used Memory	Maximum used memory of nodes in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
max_mem_used_percent	Max. Used Memory Percentage	Maximum percentage of used memory of nodes in a CSS cluster.	0~100	%	N/A	cluster_id	1 minute
sum_jvm_old_gc_count	Total GCs of Old-Generation JVM	Number of old-generation garbage collections of nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_jvm_old_gc_time	Total GC Duration of Old-Generation JVM	Total old-generation garbage collection duration of nodes in the CSS cluster.	≥ 0	ms	N/A	cluster_id	1 minute
sum_jvm_young_gc_count	Total GCs of Young-Generation JVM	Number of young-generation garbage collections of nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_jvm_young_gc_time	Total GC Duration of Young-Generation JVM	Total young-generation garbage collection duration of nodes in the CSS cluster.	≥ 0	ms	N/A	cluster_id	1 minute
sum_current_opened_http_count	Currently Open HTTP Connections	Number of HTTP connections that are open on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_total_opened_http_count	Historical Open HTTP Connections	Number of HTTP connections that were open on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
IndexingLatency	Average Index Latency	Average time required for a shard to complete an indexing operation.	≥ 0	ms	N/A	cluster_id	1 minute
IndexingRate	Average Index Rate	Average number of index operations per second in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
SearchLatency	Average Search Latency	Average time required for a shard to complete a search operation.	≥ 0	ms	N/A	cluster_id	1 minute
SearchRate	Average QPS	Average queries per second (QPS) in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
task_max_running_time	Max. Task Runtime	Duration of the most time-consuming search or write task in the cluster.	≥ 0	ms	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
number_of_pending_tasks	Number of Pending Tasks in Queue	Number of pending tasks to be processed by the master node in the CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_disk_read_requests_rate	Total Disk Read IOPS	Number of disk read requests to all nodes in the CSS cluster per second.	≥ 0	request/s	N/A	cluster_id	1 minute
sum_disk_write_requests_rate	Total Disk Write IOPS	Number of disk write requests to all nodes in the CSS cluster per second.	≥ 0	request/s	N/A	cluster_id	1 minute
sum_disk_read_bytes_rate	Total Disk Read Rate	Number of bytes of data read from the disks of all nodes in the CSS cluster per second.	≥ 0	Byte/s	1024	cluster_id	1 minute
sum_disk_write_bytes_rate	Total Disk Write Rate	Number of bytes of data written into the disks of all nodes in the CSS cluster per second.	≥ 0	Byte/s	1024	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_vector_index_off_heap_used_in_bytes	Average Off-Heap Memory Usage of Vector Indexes	Average off-heap memory (bytes) used for querying vector indexes on each node in a CSS cluster. This metric is supported only in OpenSearch 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	≥ 0	byte	1024	cluster_id	1 minute
avg_vector_index_off_heap_usage	Average Off-Heap Memory Utilization (%) of Vector Indexes	Average off-heap memory (in %) used for querying vector indexes on each node in a CSS cluster. This metric is calculated only when vector index queries occur. This metric is supported only in OpenSearch 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	0~100	%	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
max_vector_index_off_heap_used_in_bytes	Maximum Off-Heap Memory Usage of Vector Indexes	Maximum off-heap memory (bytes) used for querying vector indexes on each node in a CSS cluster. This metric is supported only in OpenSearch 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	≥ 0	byte	1024	cluster_id	1 minute
max_vector_index_off_heap_usage	Maximum Off-Heap Memory Utilization (%) of Vector Indexes	Maximum off-heap memory (in %) used for querying vector indexes on each node in a CSS cluster. This metric is calculated only when vector index queries occur. This metric is supported only in OpenSearch 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	0~100	%	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
vector_index_circuit_breaker_status	Vector Index Circuit Breaker Status	Circuit breaker status for vector indexes. ● 0: Not enabled. ● 1: Enabled. Check whether the cluster's off-heap memory usage exceeds the threshold. This metric is supported only in OpenSearch 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	0, 1	N/A	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
cold_data_storage	The size of cold data stored	The total size of cold data stored in a CSS cluster, which is the total size of the data of frozen indexes stored in an OBS bucket. This metric is measured hourly. Users cannot check this OBS bucket on the OBS console.	≥ 0	byte	1024	cluster_id	1 minute
number_of_index_creation_failures	Number of index creation failures	Number of index creation failures in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
shard_doc_exceed_threshold_count	Number of shards where the number of documents exceeds the threshold	Number of shards where the number of documents exceeds the threshold in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
write_fail_alias_error	Number of Write Failures Due to Alias Exceptions	Number of write failures caused by abnormal index aliases in a CSS cluster. This metric is supported only in OpenSearch 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	≥ 0	Count	N/A	cluster_id	1 minute
write_fail_documents_count_exceed	Number of Write Failures Due to Excessive Documents	Number of write failures caused by too many index documents in a CSS cluster. This metric is supported only in OpenSearch 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
write_fail_fields_count_exceed	Number of Write Failures Due to Excessive Fields	Number of write failures caused by too many index fields in a CSS cluster. This metric is supported only in OpenSearch 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	≥ 0	Count	N/A	cluster_id	1 minute
write_fail_shards_count_exceed	Number of Write Failures Due to Excessive Shards	Number of write failures caused by too many index shards in a CSS cluster. This metric is supported only in OpenSearch 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
write_fail_index_blocks_write	Number of Write Failures Due to Index Write Prohibition	Number of write failures caused by the setting of write prohibition in index configuration in a CSS cluster. This metric is supported only in OpenSearch 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	≥ 0	Count	N/A	cluster_id	1 minute
write_fail_mapping_malformed	Number of Write Failures Due to Incorrect Field Types	Number of write failures caused by incorrect index field types in a CSS cluster. This metric is supported only in OpenSearch 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
remote_sync_task_count	Number of Primary/ Secondary Synchronization Jobs	Number of primary/ secondary synchronization jobs for a CSS cluster. This metric is supported only in OpenSearch 2.19.0 and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	≥ 0	Count	N/A	cluster_id	1 minute
remote_sync_failed_task_count	Number of Failed Primary/ Secondary Synchronization Jobs	Number of failed primary/ secondary synchronization jobs for a CSS cluster. This metric is supported only in OpenSearch 2.19.0 and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
remote_sync_diff_bytes_of_primary_shards	Size of Data to Be Synchronized via Primary/ Secondary Synchronization Jobs for Primary Shards	Total size of data to be synchronized via primary/ secondary synchronization jobs for all primary shards in a CSS cluster. This metric is supported only in OpenSearch 2.19.0 and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	≥ 0	byte	1024	cluster_id	1 minute
remote_sync_finished_bytes_of_primary_shards	Size of Data Synchronized via Primary/ Secondary Synchronization Jobs for Primary Shards	Total size of data already synchronized via primary/ secondary synchronization jobs for all primary shards in a CSS cluster. This metric is supported only in OpenSearch 2.19.0 and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	≥ 0	byte	1024	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
remote_sync_diff_docs_of_primary_shards	Number of Documents to Be Synchronized via Primary/ Secondary Synchronization Jobs for Primary Shards	Total number of documents to be synchronized via primary/ secondary synchronization jobs for all primary shards in a CSS cluster. This metric is supported only in OpenSearch 2.19.0 and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	≥ 0	Count	N/A	cluster_id	1 minute
remote_sync_diff_bytes	Size of Data to Be Synchronized via Primary/ Secondary Synchronization Jobs	Total size of data to be synchronized via primary/ secondary synchronization jobs for all shards in a CSS cluster. This metric is supported only in OpenSearch 2.19.0 and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	≥ 0	byte	1024	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
remote_sync_finished_bytes	Size of Data Synchronized via Primary/Secondary Synchronization Jobs	Total size of data already synchronized via primary/secondary synchronization jobs for all shards in a CSS cluster. This metric is supported only in OpenSearch 2.19.0 and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	≥ 0	byte	1024	cluster_id	1 minute
remote_sync_diff_docs	Number of Documents to Be Synchronized via Primary/Secondary Synchronization Jobs	Total number of documents to be synchronized via primary/secondary synchronization jobs for all shards in a CSS cluster. This metric is supported only in OpenSearch 2.19.0 and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	≥ 0	Count	N/A	cluster_id	1 minute
index_storage	Index Data Volume	Total data size of all indexes in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
backup_failed_task_count	Number of Failed Backup Tasks	Number of failed snapshot backup tasks in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
ism_fail_count	Number of Failed Index Lifecycle Tasks	Number of failed index lifecycle tasks in a CSS cluster. This metric is supported only in OpenSearch 2.19.0 and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	≥ 0	Count	N/A	cluster_id	1 minute
disk_total	Total Disk Space	Total data disk capacity of all nodes in a CSS cluster.	≥ 0	byte	1024	cluster_id	1 minute
number_of_unassigned_shards	Number of Unallocated Shards	Total number of unallocated shards in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
shard_quota_usage	Shard Quota Usage	Ratio of allocated shards to the upper limit of the shard quantity.	0~100	%	N/A	cluster_id	1 minute
disk_iops_qos_num	Disk IOPS Overload Times	Total number of times the disk IOPS exceeds the predefined threshold.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
disk_iobw_qos_num	Disk I/O Bandwidth Overload Times	Total number of times the disk I/O bandwidth usage exceeds the predefined threshold.	≥ 0	Count	N/A	cluster_id	1 minute
max_disk_io_util	Maximum Disk I/O Usage	Maximum disk I/O usage among all disks.	0~100	%	N/A	cluster_id	1 minute
avg_disk_io_util	Average Disk I/O Usage	Average disk I/O usage among all disks.	0~100	%	N/A	cluster_id	1 minute
network_bandwidth_in	Inbound Bandwidth	Total inbound traffic per second.	≥ 0	Byte/s	1024	cluster_id	1 minute
network_bandwidth_out	Outbound Bandwidth	Total outbound traffic per second.	≥ 0	Byte/s	1024	cluster_id	1 minute
sum_thread_pool_business_worker_queue	Total Queued Tasks in business_worker Thread Pool	Total number of tasks in queue for the business_worker thread pool on nodes in a cluster. This metric is supported only in OpenSearch 2.19.0.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_business_worker_active	Total Ongoing Tasks in business_worker Thread Pool	Total number of ongoing tasks in the business_worker thread pool on nodes in a cluster. This metric is supported only in OpenSearch 2.19.0.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_business_worker_queue	Max. Queued Tasks in business_worker Thread Pool	Maximum number of tasks in queue for the business_worker thread pool on nodes in a cluster. This metric is supported only in OpenSearch 2.19.0.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_business_worker_active	Max. Ongoing Tasks in business_worker Thread Pool	Maximum number of ongoing tasks in the business_worker thread pool on nodes in a cluster. This metric is supported only in OpenSearch 2.19.0.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_thread_pool_business_worker_queue	Avg. Queued Tasks in business_worker Thread Pool	Average number of tasks in queue for the business_worker thread pool on nodes in a cluster. This metric is supported only in OpenSearch 2.19.0.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_business_worker_active	Avg. Ongoing Tasks in business_worker Thread Pool	Average number of ongoing tasks in the business_worker thread pool on nodes in a cluster. This metric is supported only in OpenSearch 2.19.0.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_transport_worker_queue	Total Queued Tasks in transport_worker Thread Pool	Total number of tasks in queue for the transport_worker thread pool on nodes in a cluster. This metric is supported only in OpenSearch 2.19.0.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
max_thread_pool_transport_worker_queue	Max. Queued Tasks in transport_worker Thread Pool	Maximum number of tasks in queue for the transport_worker thread pool on nodes in a cluster. This metric is supported only in OpenSearch 2.19.0.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_transport_worker_queue	Avg. Queued Tasks in transport_worker Thread Pool	Average number of tasks in queue for the transport_worker thread pool on nodes in a cluster. This metric is supported only in OpenSearch 2.19.0.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_flush_active	Total Ongoing Tasks in Flush Queue	Total number of ongoing tasks in the flush queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_flush_active	Max. Ongoing Tasks in Flush Queue	Maximum number of ongoing tasks in the flush queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_thread_pool_flush_active	Avg. Ongoing Tasks in Flush Queue	Average number of ongoing tasks in the flush queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_generic_active	Total Ongoing Tasks in Generic Queue	Total number of ongoing tasks in the generic queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_generic_active	Max. Ongoing Tasks in Generic Queue	Maximum number of ongoing tasks in the generic queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_generic_active	Avg. Ongoing Tasks in Generic Queue	Average number of ongoing tasks in the generic queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_management_active	Total Ongoing Tasks in Management Queue	Total number of ongoing tasks in the management queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
max_thread_pool_management_active	Max. Ongoing Tasks in Management Queue	Maximum number of ongoing tasks in the management queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_management_active	Avg. Ongoing Tasks in Management Queue	Average number of ongoing tasks in the management queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_refresh_active	Total Ongoing Tasks in Refresh Queue	Total number of ongoing tasks in the refresh queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_refresh_active	Max. Ongoing Tasks in Refresh Queue	Maximum number of ongoing tasks in the refresh queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_refresh_active	Avg. Ongoing Tasks in Refresh Queue	Average number of ongoing tasks in the refresh queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_obs_active	Total Ongoing Tasks in OBS Queue	Total number of ongoing tasks in the OBS queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_obs_active	Max. Ongoing Tasks in OBS Queue	Maximum number of ongoing tasks in the OBS queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_obs_active	Avg. Ongoing Tasks in OBS Queue	Average number of ongoing tasks in the OBS queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_obs_searcher_active	Total Ongoing Tasks in OBS Searcher Queue	Total number of ongoing tasks in the OBS Searcher queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_obs_searcher_active	Max. Ongoing Tasks in OBS Searcher Queue	Maximum number of ongoing tasks in the OBS Searcher queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_thread_pool_obs_searcher_active	Avg. Ongoing Tasks in OBS Searcher Queue	Average number of ongoing tasks in the OBS Searcher queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_write_active	Total Ongoing Tasks in Write Queue	Total number of ongoing tasks in the write queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_write_active	Max. Ongoing Tasks in Write Queue	Maximum number of ongoing tasks in the write queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_write_active	Avg. Ongoing Tasks in Write Queue	Average number of ongoing tasks in the write queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_search_active	Total Ongoing Tasks in Search Queue	Total number of ongoing tasks in the search queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_search_active	Max. Ongoing Tasks in Search Queue	Maximum number of ongoing tasks in the search queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_thread_pool_search_active	Avg. Ongoing Tasks in Search Queue	Average number of ongoing tasks in the search queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_force_merge_active	Total Ongoing Tasks in Force Merge Queue	Total number of ongoing tasks in the force merge queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_force_merge_active	Max. Ongoing Tasks in Force Merge Queue	Maximum number of ongoing tasks in the force merge queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_force_merge_active	Avg. Ongoing Tasks in Force Merge Queue	Average number of ongoing tasks in the force merge queue on nodes in a cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_get_queue	Tasks in GET Queue	Total number of tasks in queue for the GET thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_thread_pool_get_rejected	Rejected Tasks in GET Queue	Total number of tasks rejected by the GET thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_thread_pool_get_active	Total Ongoing Tasks in Get Queue	Total number of ongoing tasks in the GET thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_get_queue	Max. Tasks in GET Queue	Maximum number of tasks in queue for the GET thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_thread_pool_get_active	Max. Ongoing Tasks in GET Queue	Maximum number of ongoing tasks in the GET thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
avg_thread_pool_get_queue	Avg. Tasks in GET Queue	Average number of tasks in queue for the GET thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_thread_pool_get_active	Avg. Ongoing Tasks in GET Queue	Average number of ongoing tasks in the GET thread pool on nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
sum_left_node_count	Disconnected Nodes	The number of nodes that are disconnected in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute

Node Monitoring Metrics

Table 3-181 OpenSearch cluster node monitoring metrics

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
jvm_heap_usage	JVM Heap Usage	JVM heap memory usage of a node.	0~100	%	N/A	cluster_id,instance_id	1 minute
cpu_usage	CPU Usage	CPU usage.	0~100	%	N/A	cluster_id,instance_id	1 minute
disk_util	Max. Disk Usage	Disk usage of a node	0~100	%	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
load_average	Average Load	Average number of queued tasks per minute on a node	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
open_file_descriptors	Open File Descriptors	Number of opened file descriptors on a node	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
max_file_descriptors	Max. Allowed File Descriptors	Maximum number of allowed file descriptors	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_write_queue	Tasks in Write Queue	Number of tasks in queue for the write thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_search_queue	Tasks in Search Queue	Number of tasks in queue for the search thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_force_merge_queue	Tasks in Force Merge Queue	Number of tasks in queue for the force merge thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_write_rejected	Rejected Tasks in Write Queue	Number of tasks rejected by the write thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_search_rejected	Rejected Tasks in Search Queue	Number of tasks rejected by the search thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
thread_pool_force_merge_rejected	Rejected Tasks in Force Merge Queue	Number of tasks rejected by the force merge thread pool	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_write_threads	Size of Write Thread Pool	Size of a write thread pool	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_search_threads	Size of Search Thread Pool	Size of a search thread pool	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_force_merge_threads	Size of Force Merge Thread Pool	Size of a force merge thread pool	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_flush_queue	Tasks in Flush Queue	Number of tasks in queue for the flush thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_flush_rejected	Rejected Tasks in Flush Queue	Number of tasks rejected by the flush thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_flush_threads	Size of Flush Thread Pool	Size of the flush thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_generic_queue	Tasks in Generic Queue	Number of tasks in queue for the generic thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
thread_pool_generic_rejected	Rejected Tasks in Generic Queue	Number of tasks rejected by the generic thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_generic_threads	Size of Generic Thread Pool	Size of the generic thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_management_queue	Tasks in Management Queue	Number of tasks in queue for the management thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_management_rejected	Rejected Tasks in Management Queue	Number of tasks rejected by the management thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_management_threads	Size of Management Thread Pool	Size of the management thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_refresh_queue	Tasks in Refresh Queue	Number of tasks in queue for the refresh thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_refresh_rejected	Rejected Tasks in Refresh Queue	Number of tasks rejected by the refresh thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_refresh_threads	Size of Refresh Thread Pool	Size of the refresh thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
thread_pool_obs_searcher_queue	Tasks in OBS Searcher Queue	Number of tasks in queue for the OBS Searcher thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_searcher_rejected	Rejected Tasks in OBS Searcher Queue	Number of tasks rejected by the OBS Searcher thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_searcher_threads	Size of OBS Searcher Thread Pool	Size of the OBS Searcher thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_queue	Tasks in OBS Queue	Number of tasks in queue for the OBS thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_rejected	Rejected Tasks in OBS Queue	Number of tasks rejected by the OBS thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_threads	Size of OBS Thread Pool	Size of the OBS thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_upload_queue	Tasks in OBS Upload Queue	Number of tasks in queue for the OBS Upload thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_upload_rejected	Rejected Tasks in OBS Upload Queue	Number of tasks rejected by the OBS Upload thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
thread_pool_obs_upload_threads	Size of OBS Upload Thread Pool	Size of the OBS Upload thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_download_queue	Tasks in OBS Download Queue	Number of tasks in queue for the OBS Download thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_download_rejected	Rejected Tasks in OBS Download Queue	Number of tasks rejected by the OBS Download thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_download_threads	Size of OBS Download Thread Pool	Size of the OBS Download thread pool.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
free_fs_size	Available Size of File Systems	Available size of file systems in a CSS cluster.	≥ 0	byte	1024	cluster_id,instance_id	1 minute
total_fs_size	Total Size of File Systems	Total size of file systems in a CSS cluster.	≥ 0	byte	1024	cluster_id,instance_id	1 minute
jvm_old_gc_count	Total GCs of Old-Generation JVM	Number of old-generation garbage collection times	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
jvm_old_gc_time	Total GC Duration of Old-Generation JVM	Time spent on old-generation garbage collection	≥ 0	ms	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
jvm_young_gc_count	Total GCs of Young-Generation JVM	Number of young-generation garbage collection times	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
jvm_young_gc_time	GC Duration of Young-Generation JVM	Time spent on young-generation garbage collection	≥ 0	ms	N/A	cluster_id,instance_id	1 minute
mem_free_in_bytes	Available Memory	Unused memory space of a node.	≥ 0	byte	1024	cluster_id,instance_id	1 minute
mem_free_percent	Available Memory Percentage	Percentage of unused memory space on a node.	0~100	%	N/A	cluster_id,instance_id	1 minute
mem_used_in_bytes	Used Memory	Used memory space of a node.	≥ 0	byte	1024	cluster_id,instance_id	1 minute
current_opened_http_count	Currently Open HTTP Connections	Number of HTTP connections currently open on a node	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
total_opened_http_count	Total Open HTTP Connections	Total number of HTTP connections on a node	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
disk_read_requests_rate	Disk Read IOPS	Number of disk read requests to a node in the CSS cluster per second.	≥ 0	request/s	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
disk_write_requests_rate	Disk Write IOPS	Number of disk write requests to a node in the CSS cluster per second.	≥ 0	request/s	N/A	cluster_id,instance_id	1 minute
disk_read_bytes_rate	Disk Read Rate	Number of bytes of data read from the disks of a node in the CSS cluster per second.	≥ 0	Byte/s	1024	cluster_id,instance_id	1 minute
disk_write_bytes_rate	Disk Write Rate	Number of bytes of data written into the disks of a node in the CSS cluster per second.	≥ 0	Byte/s	1024	cluster_id,instance_id	1 minute
shards_count	Shards	Number of shards per node in a CSS cluster.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
shard_quota_usage	Shard Quota Usage	Ratio of allocated shards to the upper limit of the shard quantity.	0~100	%	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
vector_index_off_heap_used_in_bytes	Off-Heap Memory Usage of Vector Indexes	Off-heap memory (bytes) used for querying vector indexes on nodes in a CSS cluster. This metric is supported only in OpenSearch 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	≥ 0	byte	1024	cluster_id,instance_id	1 minute
vector_index_off_heap_usage	Off-Heap Memory Utilization (%) of Vector Indexes	Off-heap memory (in %) used for querying vector indexes on nodes in a CSS cluster. This metric is supported only in OpenSearch 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later).	0~100	%	N/A	cluster_id,instance_id	1 minute
disk_iops_qos_num	Disk IOPS Overload Times	Total number of times the disk IOPS exceeds the predefined threshold.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
disk_iobw_qos_num	Disk I/O Bandwidth Overload Times	Total number of times the disk I/O bandwidth usage exceeds the predefined threshold.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
max_disk_io_util	Maximum Disk I/O Usage	Maximum disk I/O usage among all disks.	0~100	%	N/A	cluster_id,instance_id	1 minute
avg_disk_io_util	Average Disk I/O Usage	Average disk I/O usage among all disks.	0~100	%	N/A	cluster_id,instance_id	1 minute
network_bandwidth_in	Inbound Bandwidth	Total inbound traffic per second on the VM.	≥ 0	Byte/s	1024	cluster_id,instance_id	1 minute
network_bandwidth_out	Outbound Bandwidth	Total outbound traffic per second on the VM.	≥ 0	Byte/s	1024	cluster_id,instance_id	1 minute
thread_pool_business_worker_queue	Queued Tasks in business_worker Thread Pool	Number of tasks in queue for the business_worker thread pool on the node. This metric is supported only in OpenSearch 2.19.0.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
thread_pool_business_worker_active	Ongoing Tasks in business_worker Thread Pool	Number of ongoing tasks in the business_worker thread pool on the node. This metric is supported only in OpenSearch 2.19.0.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_transport_worker_queue	Queued Tasks in transport_worker Thread Pool	Number of tasks in queue for the transport_worker thread pool on the node. This metric is supported only in OpenSearch 2.19.0.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
node_indexing_rate	Node Indexing Rate	Indexing rate on the node (average number of indexing operations per second).	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
node_search_rate	Node Query Rate	Query rate on the node (average number of queries per second).	≥ 0	Count	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
node_get_rate	Node Get Rate	Get rate on the node (average number of get requests per second).	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_flush_active	Ongoing Tasks in Flush Queue	Number of ongoing tasks in the Flush queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_generic_active	Ongoing Tasks in Generic Queue	Number of ongoing tasks in the generic queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_management_active	Ongoing Tasks in Management Queue	Number of ongoing tasks in the management queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_refresh_active	Ongoing Tasks in Refresh Queue	Number of ongoing tasks in the refresh queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_active	Ongoing Tasks in OBS Queue	Number of ongoing tasks in the OBS queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_obs_searcher_active	Ongoing Tasks in OBS Searcher Queue	Number of ongoing tasks in the OBS Searcher queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_write_active	Ongoing Tasks in Write Queue	Number of ongoing tasks in the write queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
thread_pool_search_active	Ongoing Tasks in Search Queue	Number of ongoing tasks in the search queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_force_merge_active	Ongoing Tasks in Force Merge Queue	Number of ongoing tasks in the Force Merge queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_get_rejected	Rejected Tasks in GET Queue	Number of rejected tasks in the GET queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_get_queue	Tasks in GET Queue	Number of queued tasks in the GET queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
thread_pool_get_active	Ongoing Tasks in GET Queue	Number of ongoing tasks in the GET queue.	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
rss_usage	RSS Memory Usage	The ratio of RSS memory to total physical memory. This metric is supported only in OpenSearch 2.19.0.	0~100	%	N/A	cluster_id,instance_id	1 minute

If an object is in a hierarchical system, specify the monitored dimension in hierarchical form when you use an API to query the metrics of this object.

For example, to query the CPU usage (cpu_usage) of an OpenSearch cluster node in CSS, the dimension information of this metric is **cluster_id,instance_id**, where **cluster_id** indicates level 0 and **instance_id** level 1.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- To query a single metric by calling an API, the **instance_id** dimension is used as follows:

```
dim.0=cluster_id,3d65c1ac-9a9f-4c5f-a054-35184a087bb2&dim.1=instance_id,6666cd76f96956469e7be39d750cc7d9
```

  where, **3d65c1ac-9a9f-4c5f-a054-35184a087bb2** and **6666cd76f96956469e7be39d750cc7d9** are the values of **cluster_id** and **instance_id**, respectively. For how to obtain these values, see **Dimensions**.

- To query multiple metrics by calling an API, the **instance_id** dimension is used as follows:

```
"dimensions": [
   {
      "name": "cluster_id",
      "value": "3d65c1ac-9a9f-4c5f-a054-35184a087bb2"
   },
   {
      "name": "instance_id",
      "value": "6666cd76f96956469e7be39d750cc7d9"
   }
]
```

  where, **3d65c1ac-9a9f-4c5f-a054-35184a087bb2** and **6666cd76f96956469e7be39d750cc7d9** are the values of **cluster_id** and **instance_id**, respectively. For how to obtain the values, see **Dimensions**.

Dimensions

Table 3-182 Dimension description

Key	Value
cluster_id	Cluster ID. You can obtain this value from the **clusters[].id** field in the response body of the **Querying the Cluster List** API.
cluster_id,instance_id	Cluster node ID. You can obtain this value from the **clusters[].instances[].id** field in the response body of the **Querying the Cluster List** API.

3.13 Managing Logs

3.13.1 Querying Logs

During routine O&M, when an OpenSearch cluster becomes sluggish, returns an error for a query request, or shows a yellow status, the O&M team must be able to obtain relevant information immediately. CSS provides a log query feature that aggregates underlying server logs to the console, where you can query them by node and log level in seconds. By analyzing run logs, slow query logs, and deprecation logs, you can quickly locate the root causes of faults and issues. Additionally, you can dynamically adjust log levels and enable trace logging for debugging, allowing you to obtain the most detailed log records without restarting your cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Querying Recent Logs

Query recently generated logs that are not yet archived.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Logs > Log Search**. The **Log Search** page is displayed.

   You can search records by log type, node, log level, or keyword. For a detailed description of each type of logs, see **Log Types**.

> ⚠ CAUTION
>
> When a log file reaches 128 MB or when the time reaches 00:00 UTC, the system automatically compresses and archives it. Only unarchived logs appear on the log search page, while archived logs remain accessible through the log backup function. For more information, see **Backing Up Logs to OBS**.

Adjusting Log Levels Dynamically

Log4j2 is used as the log component in OpenSearch clusters. Multiple log levels (ERROR, WARN, INFO, DEBUG, and TRACE) are supported. The default log level is INFO. To facilitate troubleshooting and debugging, you can dynamically adjust the log levels.

- INFO is the default log level. The levels, in order of increasing detail, are: ERROR, WARN, INFO, DEBUG, and TRACE. When INFO is set, you will see logs for itself and all higher-severity levels (ERROR and WARN), while more verbose levels (DEBUG and TRACE) are excluded.
- You can change the log level of a specified module in real time via an OpenSearch API.

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

      The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.
2. Run the following command to change the log level, for example, to DEBUG:

```
PUT _cluster/settings
{
  "persistent": {
    "logger": {
      "org.opensearch.action": "DEBUG"
    }
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  }
}
```

3. Refresh the log query page and check whether DEBUG logs have been generated. You should always restore the log level to the default setting once you finish the task at hand.
4. Run the following command to restore the default log level **INFO**:

```
PUT _cluster/settings
{
  "persistent": {
    "logger": {
      "org.opensearch.action": null
    }
  }
}
```

Enabling Trace Logging

To analyze communication details on the HTTP or transport layer, you can enable trace logging for the HTTP or Transport module to obtain detailed log records.

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

      The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.
2. Run the following command to enable trace logging:

```
PUT _cluster/settings
{
  "transient": {
    "logger.org.opensearch.transport.TransportService.tracer": "trace",
    "transport.tracer.include": "",
    "http.tracer.include": "",
    "logger.org.opensearch.http.HttpTracer": "trace"
  }
}
```

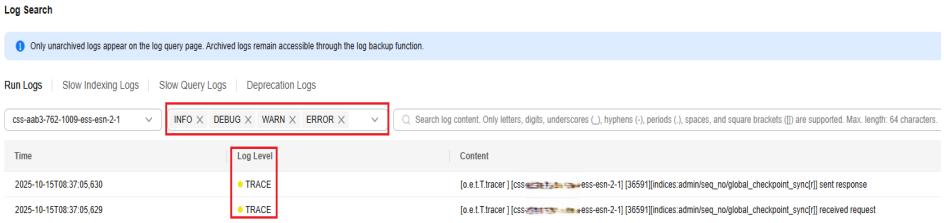
CAUTION

Enabling trace logging is a non-persistent configuration and will be disabled upon a cluster restart. It is typically used for emergency packet capture and analysis during troubleshooting.

3. Go to the log details page to view trace logs.
   a. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
   b. Choose **Logs > Log Search**. The **Log Search** page is displayed.
   c. Select all log levels (mandatory) and view trace logs.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-49 Viewing trace logs



Log Types

Table 3-183 Introducing different log types

Log Type	Description	Purpose
Run logs	Run logs, or main logs, record the cluster status and key information about write and query operations. For example, write logs record operations such as index creation, index mapping update, and write queue exhaustion; and query logs record query queue status and query exceptions.	Check the status and write and query operations of each cluster node, including inter-node connectivity, full GC, index creation or deletion, and cluster-level query errors.
Slow indexing logs	Slow indexing logs record indexing operations (such as bulk, index, update, and delete) that took a long time to complete, helping you identify performance bottlenecks.	In the case of slow write performance, you can query slow indexing logs to locate the cause.
Slow query logs	Slow query logs record search requests that took a long time to complete. They help you monitor and analyze time-consuming search requests, so you can identify performance bottlenecks, optimize SQL queries, and improve overall system performance.	In the case of slow query performance, you can query slow query logs to locate the cause.
Deprecation logs	Deprecation logs record deprecation warnings. Deprecation warnings are written to this log when you use APIs, configurations, or functions that are marked for removal in future versions.	Check for APIs or features that are about to expire in future versions.

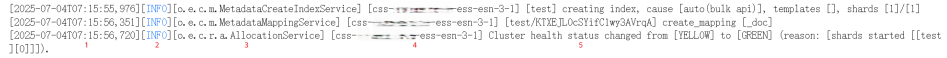
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Log Type	Description	Purpose
Access logs	Access logs record cluster access requests, such as the request path and source address. You cannot check access logs on the console. To check them, you need to back them up to an OBS bucket or transfer them to a target cluster first.	If there is a surge in service requests, you can analyze the request sources and paths by checking the access logs.

- **Run log description**

  Run logs record the cluster status and key information about write and query operations. For example, the log record below indicates that an index named **test** was created and afterwards the cluster status changed from **YELLOW** to **GREEN**.

  Figure 3-50 Run log sample

  Log content:

  - 1. Log generation time
  - 2. Log level, which can be DEBUG, INFO, WARN, or ERROR
  - 3. Log-generating module
  - 4. Name of the log-generating node
  - 5. Log content

- **Slow indexing log description**

  Slow indexing logs record indexing operations that took a long time to complete. For example, the log record below shows an indexing request that lasted longer than the preset threshold. The log contains the index name, duration, and request content.

  Figure 3-51 Slow indexing log sample

  Log content:

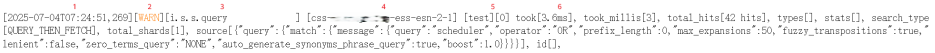
  - 1. Log generation time
  - 2. Log level, which can be DEBUG, INFO, WARN, or ERROR
  - 3. Log-generating module
  - 4. Name of the log-generating node
  - 5. Index name and ID
  - 6. Log content. In this example, the log recorded the request execution duration, index type, and index request body.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- **Slow query log description**

  Slow query logs record search requests that took a long time to complete. For example, the log record below shows a search request that lasted longer than the preset threshold. The log contains the index name, duration, and request content.

  Figure 3-52 Slow query log sample

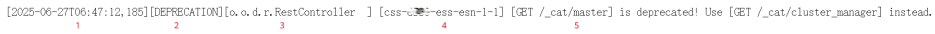
  Log content:

  - 1. Log generation time
  - 2. Log level, which can be DEBUG, INFO, WARN, or ERROR
  - 3. Log-generating module
  - 4. Name of the log-generating node
  - 5. Index name and shard ID
  - 6. Log content. In this example, the log recorded the query duration, number of hits, and query request body.

- **Deprecation log description**

  Deprecation logs record deprecation warnings. For example, the log record below indicates that **GET /_cat/master** has been deprecated and should be replaced with **GET /_cat/cluster_manager**.

  Figure 3-53 Deprecation log sample

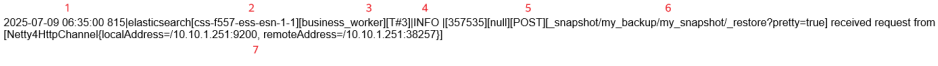
  Log content:

  - 1. Log generation time
  - 2. Log level, which can only be DEPRECATION.
  - 3. Log-generating module
  - 4. Name of the log-generating node
  - 5. Log content

- **Access log description**

  Access logs record cluster access requests and source addresses. For example, the log record below has recorded source information for the /_snapshot/my_backup/my_snapshot/_restore?pretty=true operation.

  Figure 3-54 Access log sample

  Log content:

  - 1. Log generation time
  - 2. Name of the log-generating node

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- 3. Name of the log-generating thread
- 4. Log level, which can be DEBUG, INFO, WARN, or ERROR
- 5. Log request method
- 6. Request path
- 7. Source and destination addresses of the request

3.13.2 Ingesting Logs into a Cluster

For large clusters that generate high volumes of logs, traditional search methods may struggle to extract comprehensive insights from the log data. To enable more advanced log analysis, you can leverage CSS's log ingestion feature to ingest logs generated by your cluster in real-time into a designated index in a specified cluster. Tools like OpenSearch Dashboards can then be used to visualize and analyze the logs. By converting unstructured log data into searchable indexes, you can perform centralized log analysis across multiple clusters, greatly enhancing the efficiency of troubleshooting complex issues.

Constraints

Log ingestion is supported only when the OpenSearch cluster version is 1.3.6 or 2.19.0 and the image version is no earlier than x.x.x_24.2.0_xxx.

Enabling Real-Time Log Ingestion

Log ingestion converts unstructured log data into structured documents stored in searchable OpenSearch indexes.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Logs > Log Ingestion**. The **Log Ingestion** page is displayed.

   CAUTION

   If the **Log Ingestion** tab is not displayed, the cluster does not support log ingestion.

5. Enable log ingestion. If it is already enabled, skip this step.
   a. Click **Enable Log Ingestion**. In the displayed dialog box, configure necessary settings.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-184 Log ingestion settings

Parameter	Description
Index Prefix Name	If you set a prefix for the log file indexes, the log index names will use the format **index prefix name + log ingestion date**. The unit of log ingestion is days. An index prefix name is a string of 1 to 128 characters. It can only contain digits, lowercase letters, underscores (_), and hyphens (-).
Retention Period	Set the log retention period. Ingested logs will be deleted upon expiration of this period. Value range: 1 to 3650 Unit: days
Log Storage Cluster	Select a cluster to store ingested logs. Options include **Current cluster** and **Other clusters**. • **Current cluster**: Store ingested logs in the current cluster. • **Other clusters**: Store ingested logs in another cluster. Select a cluster from the **Cluster** drop-down list to use as the log storage destination. Make sure this cluster is in the same VPC and have the same version as the current cluster. When you select another cluster, click **Check** to verify network connectivity between the current cluster and the selected other cluster. Log ingestion can start only when "The current cluster is available" is returned.

b. Click **OK** to enable log ingestion.

The configuration information will be displayed on the **Log Ingestion** tab.

6. On the **Log Ingestion** tab, if the task status changes to **Running**, log ingestion has started.

Viewing Ingested Logs

On the **Log Ingestion** tab, click **Access Dashboards**. The OpenSearch Dashboards login page for the log storage cluster is displayed. OpenSearch Dashboards's powerful visualization features simplify log search and analytics.

Figure 3-55 Viewing ingested logs



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

To view details about the log storage cluster, click the cluster name in cluster list to go to the cluster overview page.

Disabling Log Ingestion

If you no longer need the log ingestion feature, you can disable it to reclaim storage resources.

On the **Log Ingestion** page, click **Disable Log Ingestion**. In the displayed dialog box, click **OK**.

CAUTION

Disabling log ingestion does not automatically remove ingested log data. Rather, it is deleted by the system upon expiration of its retention period. You can also manually delete it from the log storage cluster before the retention period expires.

3.13.3 Backing Up Logs to OBS

During routine O&M, querying logs in real-time through the web console is a commonly used method for diagnosing issues. However, due to limited local storage capacity, the system automatically archives historical logs when they exceed 128 MB or at 00:00 UTC. Long-term log retention is essential for compliance and operational purposes, but it must also be cost-effective. To address this, CSS provides a log backup feature that incrementally synchronizes run logs, slow query logs, and deprecation logs to OBS based on predefined policies. This solution takes advantage of OBS's vast, low-cost storage for long-term retention while also allowing users to download historical log files at any time for detailed offline analysis.

Impact on Billing

Standard storage is used for storing log backups in OBS. For pricing details, see **Billing**section "Billing" in *Object Storage Service User Guide*.

Prerequisites

You have created an OBS bucket for storing log backups. The OBS bucket must meet the following requirements:

- **Storage Class**: **Standard**.
- **Region**: the same as that of your CSS cluster.

Enabling Log Backup

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Logs > Log Backup**. The **Log Backup** page is displayed.

   Check whether log backup is enabled for the cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- Yes: No further action is required.
- No: Go to the next step.

5. On the **Log Backup** page, click **Enable Backup**. In the displayed dialog box, configure the necessary settings.

Table 3-185 Log backup settings

Parameter	Description
OBS Bucket	Select an OBS bucket for storing log backups from the drop-down list box. If no OBS buckets meet your requirements, click **Create Bucket** to go to the OBS console and create one. For details, see **Creating a Bucket**. An IAM user must be granted the following OBS permissions to view available OBS buckets: obs:bucket:getBucketStoragePolicy obs:bucket:listAllMyBuckets
Backup Path	Set the log storage location in the OBS bucket. The backup path cannot: • Contain the following characters: **\:*?"<>|'{}** • Start with a slash (/). • Start or end with a period (.). • Contain more than two consecutive slashes (/) or periods (.). • Exceed 512 characters.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
IAM Agency	To back up data to an OBS bucket, you must have the write permission to it. By configuring an IAM agency, you can authorize CSS to access its OBS resources through an associated account. • If you are configuring an agency for the first time, click **Automatically Create IAM Agency** to create **css-obs-agency**. • If there is an IAM agency automatically created earlier, you can click **One-click authorization** to have the OBS Administrator permissions deleted automatically, and have the following custom policies added automatically instead to implement more refined permissions control. "obs:bucket:GetBucketLocation", "obs:object:GetObjectVersion", "obs:object:GetObject", "obs:object:DeleteObject", "obs:bucket:HeadBucket", "obs:bucket:GetBucketStoragePolicy", "obs:object:DeleteObjectVersion", "obs:bucket:ListBucketVersions", "obs:bucket:ListBucket", "obs:object:PutObject" • When OBS buckets use SSE-KMS encryption, the IAM agency must be granted KMS permissions. You can click **Automatically Create IAM Agency** and **One-click authorization** to have the following custom policies created automatically. "kms:cmk:create", "kms:dek:create", "kms:cmk:get", "kms:dek:decrypt", "kms:cmk:list" • To use **Automatically Create IAM Agency** and **One-click authorization**, the following minimum permissions are required: "iam:agencies:listAgencies", "iam:roles:listRoles", "iam:agencies:getAgency", "iam:agencies:createAgency", "iam:permissions:listRolesForAgency", "iam:permissions:grantRoleToAgency", "iam:permissions:listRolesForAgencyOnProject", "iam:permissions:revokeRoleFromAgency", "iam:roles:createRole" • To use an IAM agency, the following minimum permissions are required: "iam:agencies:listAgencies", "iam:agencies:getAgency", "iam:permissions:listRolesForAgencyOnProject", "iam:permissions:listRolesForAgency" **WARNING** The agency name can contain only letters (case-sensitive), digits, underscores (_), and hyphens (-). Otherwise, the backup will fail.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Determine whether to enable **Automatic backup** based on service needs.

- To enable automatic, periodic log backup, select **Automatic backup**, and set the backup policy. For more, see **Configuring Automatic Log Backup**.
- If manual log backup is sufficient, deselect **Automatic backup**.

6. Click **OK** to enable log backup.

   The configuration information will be displayed on the **Log Backup** tab.

Configuring Automatic Log Backup

After enabling log backup for a cluster, you can configure an auto backup policy to periodically back up logs based on a preset schedule.

1. On the **Log Backup** page, click **Modify Settings** on the right.

   Figure 3-56 Modifying Settings

2. In the **Modify Settings** dialog box, select **Automatic backup**, and configure the backup policy.

   Table 3-186 Automatic backup settings

Parameter	Description
Time Zone	Select a time zone for the backup start time.
Backup Start Time	Specify the start time of auto backup. Select a value from the drop-down list. The value range is from **00:00** to **23:00**. The backup always happens on the hour.

3. Click **OK** to enable automatic log backup.

   Automatic log backup tasks are displayed in the task list. When **Task Status** in the task list changes to **Succeeded**, the backup is successful.

4. If automatic log backup is no longer required, click **Modify Settings**. In the displayed dialog box, deselect **Automatic backup**, and click **OK**. In the displayed confirmation dialog box, click **OK**.

   Disabling log backup does not automatically delete log backups automatically generated earlier. Instead, you need to manually delete them on the OBS console.

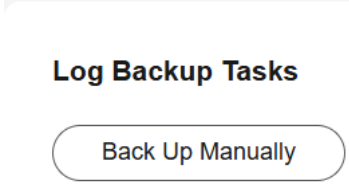
Manually Backing Up Logs

After enabling log backup for a cluster, you can manually back up its logs at any time.

1. On the **Log Backup** page, click **Back Up Manually** under **Log Backup Tasks**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-57 Manually backing up logs



2. In the displayed dialog box, confirm the log backup path and click **OK**.

   The log backup task is displayed in the task list. When **Task Status** in the task list changes to **Succeeded**, the backup is successful.

Checking Log Backup Files

Logs are backed up incrementally. After the backup is successful, you can access the target OBS bucket to obtain the full log files by clicking **Log Path**.

Figure 3-58 OBS bucket address

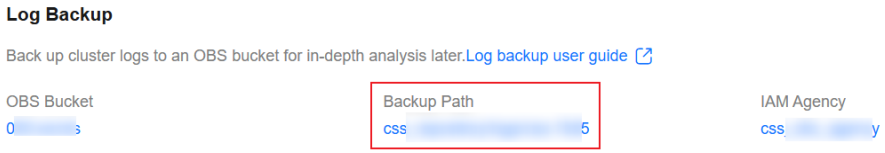


Table 3-187 lists the log types, where **clustername** indicates the cluster name.

Table 3-187 Log types

Log Name	Description
*clustername*_deprecation.log	Deprecation log file
*clustername*_index_indexing_slowlog.log	Slow indexing log file
*clustername*_index_search_slowlog.log	Slow query log file
clustername.log	Run log file
*clustername*_access.log	Access log file

Disabling Log Backup

Disable log backup if it is no longer needed.

On the **Log Backup** page, click **Disable Backup**. In the displayed dialog box, click **OK**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

> **CAUTION**
>
> Disabling log backup does not automatically delete existing backups. Instead, you need to manually delete them on the OBS console.

3.13.4 Viewing Audit Logs

To ensure data security and compliance, enterprises must routinely record and audit operations performed on their cloud service resources. With Cloud Trace Service (CTS), you can record operations associated with CSS OpenSearch clusters for later query, auditing, and backtracking.

Prerequisites

CTS has been enabled.

Key operations that can be audited using CTS

Table 3-188 Key operations that can be audited using CTS

Operation	Resource Type	Event Name
Querying product dynamics on the Overview page	cluster	listByGraph
Querying cluster and disk information on the Dashboards page	cluster	showDashboard
Querying supported engine types	cluster	listDatastores
Querying flavors supported by a specified engine type	cluster	getSingleDatabaseFlavors
Querying supported AZs	user	getAZInfo
Querying available disk types	cluster	listDiskTypes
Obtaining the node flavor list	cluster	listFlavors
Querying details about a specified flavor	cluster	showFlavorDetail
Querying feature availability	user	showFunctionSwitch
Querying available resource quotas	user	queryQuota
Creating a cluster	cluster	createCluster
Querying the cluster list	cluster	listClusters
Querying cluster details	cluster	showClusterDetail

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Operation	Resource Type	Event Name
Querying the storage capacity of a cluster	cluster	getClusterDiskInfos
Updating resource status	cluster	updateServiceStatus
Updating resource metadata	cluster	updateServiceMetadata
Changing the cluster name	cluster	updateClusterName
Modifying the cluster description	cluster	updateClusterDesc
Deleting a failed snapshot task	cluster	deleteFailedTaskByTaskId
Deleting all failed snapshot tasks	cluster	deleteAllFailedTask
Querying all failed snapshot tasks	cluster	listAllFailedTasks
Clearing the status of a specified task for a cluster	cluster	clearClusterAction
Quickly restarting a cluster	cluster	restartCluster
Performing a rolling restart on a cluster	cluster	rollingRestart
Deleting a cluster	cluster	deleteCluster
Creating a cluster (V2)	cluster	createClusterV2
Querying the cluster list	cluster	listClustersDetails
Querying the cluster list (V2)	cluster	listClustersDetailsV2
Changing the security group	cluster	changeSecurityGroup
Changing the password of a cluster	cluster	resetPassword
Downloading a security certificate	cluster	downloadCert
Changing the subnet of a cluster	cluster	changeClusterSubnet
Obtaining a cluster route	cluster	getRoutes
Updating cluster routes	cluster	updateRoute
Obtaining private network access control policies	cluster	getAccessPolicy

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Operation	Resource Type	Event Name
Modifying private network access control policies	cluster	updateAccessPolicy
Adding data nodes to a cluster	cluster	updateExtendCluster
Adding more nodes and storage capacity	cluster	updateExtendInstanceStorage
Changing the specifications of data nodes in a cluster	cluster	updateFlavor
Changing the flavor of a specified node type	cluster	updateFlavorByType
Querying available flavors	cluster	queryResizeFlavors
Removing specified nodes	cluster	updateShrinkNodes
Removing nodes of a specific type	cluster	updateShrinkCluster
Querying information about nodes to be deleted	cluster	queryNeedDeleteInstances
Changing the security mode	cluster	changeMode
Querying the system indexes of a security-mode cluster	cluster	getSecurityIndex
Adding new node types	cluster	addIndependentNode
Enabling master nodes	cluster	addIndependentMasterNode
Enabling client nodes	cluster	addIndependentClientNode
Enabling cold data nodes	cluster	addIndependentColdNode
Replacing a specified node	cluster	updateInstance
Changing AZs for cluster nodes	cluster	updateAzByInstanceType
Upgrading the cluster kernel	cluster	upgradeCore
Obtaining information about the target image to upgrade to	cluster	listImages
Obtaining upgrade details	cluster	upgradeDetail
Retrying a failed kernel upgrade or AZ change task	cluster	retryAction
Terminating a failed kernel upgrade or AZ change task	cluster	abortAction

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Operation	Resource Type	Event Name
Enabling or disabling public network access	cluster	bindOrUnbindPublicIp
Enabling public network access	cluster	createBindPublic
Disabling public network access	cluster	updateUnbindPublic
Modifying public network bandwidth	cluster	updatePublicBandWidth
Enabling the public network access control whitelist	cluster	startPublicWhitelist
Disabling the public network access control whitelist	cluster	stopPublicWhitelist
Enabling public network access to Dashboards	cluster	startKibanaPublic
Disabling public network access to Dashboards	cluster	updateCloseKibana
Modifying the public network bandwidth for Dashboards	cluster	updateAlterKibana
Modifying public network access control for Dashboards	cluster	updatePublicKibanaWhitelist
Disabling public network access control for Dashboards	cluster	stopPublicKibanaWhitelist
Modifying the public network bandwidth for the cluster or Dashboards	cluster	updateBandWidth
Modifying the public network access control whitelist for the cluster or Dashboards	cluster	updateWhitelist
Querying the public network or Dashboards public network to be unbound	cluster	queryNeedDeleteBandwidth
Enabling logging	cluster	startLogs
Disabling logging	cluster	stopLogs
Querying the log backup task list	cluster	listLogsJob
Querying basic log configurations	cluster	showGetLogSetting

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Operation	Resource Type	Event Name
Modifying basic log configurations	cluster	updateLogSetting
Enabling automatic log backup	cluster	startLogAutoBackupPolicy
Disabling automatic log backup	cluster	stopLogAutoBackupPolicy
Manually backing up logs	cluster	createLogBackup
Configuring automatic log backup	cluster	autoCreateLogBackup
Querying logs	cluster	showLogBackup
Checking the network connectivity between the current cluster and the destination cluster	cluster	detectClusterConnectivity
Automatically configuring basic cluster snapshot settings	cluster	startAutoSetting
Configuring basic cluster snapshot settings	cluster	openSnapshotFunction
Modifying the basic settings of a cluster snapshot	cluster	updateSnapshotSetting
Manually creating a snapshot	snapshot	createSnapshot
Configuring automatic snapshot creation	snapshot	autoCreateSnapshot
Restoring a snapshot	snapshot	restoreSnapshot
Deleting a specified snapshot	snapshot	deleteSnapshot
Setting an automatic snapshot creation policy	cluster	createAutoCreatePolicy
Querying cluster snapshot policies	cluster	queryClustersBackupPolicy
Querying automatic snapshot creation policies	cluster	showAutoCreatePolicy
Querying the cluster snapshot list	cluster	listSnapshots
Querying the cluster snapshot list	cluster	listClusterBackups
Disabling snapshots	cluster	stopSnapshot

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Operation	Resource Type	Event Name
Enabling automatic snapshot creation	cluster	startAutoCreateSnapshots
Disabling automatic snapshot creation	cluster	stopAutoCreateSnapshots
Querying the destination cluster for snapshot restoration	cluster	queryTargetRestoreCluster
Enabling or disabling VPC Endpoint	cluster	bindOrUnbindEIP
Enabling VPC Endpoint	cluster	startVpecp
Disabling VPC Endpoint	cluster	stopVpecp
Obtaining a VPCEP connection	cluster	showVpcepConnection
Updating a VPCEP connection	cluster	updateVpcepConnection
Modifying the VPCEP whitelist	cluster	updateVpcepWhitelist
Obtaining the parameter settings list	cluster	listYmls
Obtaining the parameter settings task list	cluster	listYmlsJob
Modifying parameter settings	cluster	updateYmls
Querying dedicated load balancers supported by a cluster	cluster	listElbs
Enabling or disabling a cluster load balancer	cluster	enableOrDisableElb
Configuring a cluster load balancer listener	cluster	createElbListener
Updating load balancer listeners for a cluster	cluster	updateEsListener
Obtaining information about a cluster's load balancers	cluster	showElbDetail
Querying certificates supported by a load balancer	cluster	listElbCerts
Querying historical reports and details for intelligent diagnostics	cluster	listAiOps

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Operation	Resource Type	Event Name
Manually creating an intelligent diagnostics task	cluster	createAiOps
Automatically creating an intelligent diagnostics task	cluster	autoCreateAiOps
Deleting an intelligent diagnostics task	cluster	deleteAiOps
Obtaining SMN topics available for intelligent O&M alarms	cluster	listSmnTopics
Updating the scheduled diagnostics configuration for intelligent O&M	cluster	updateAiOpsSetting
Disabling scheduled diagnostics for intelligent O&M	cluster	closeAiOpsSetting
Viewing the scheduled diagnostics configuration for intelligent O&M	cluster	showAiOpsSetting
Viewing intelligent O&M diagnostics items	cluster	showAiOpsDetector
Querying default plug-ins	cluster	getDefaultPlugins
Creating an agency	user	createAgency
Querying the agency list	cluster	listAgencies
Querying available OBS buckets	user	getAvailableBuckets
Querying files in a specified OBS bucket	cluster	listObjects
Querying resources by tag	cluster	getResourceByTags

Querying Real-Time Traces

After a management tracker is created on the CTS console, the system starts recording operations performed on cloud service resources. After a data tracker is created, the system starts recording operations performed on data in OBS buckets. CTS retains operation records generated in the latest seven days.

3.14 Managing Alarms

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

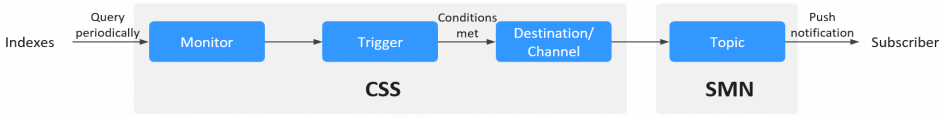
3.14.1 Configuring OpenSearch Alert Notifications via SMN

When monitoring big data platforms in real time, O&M teams need to be promptly notified of cluster health status or any abnormal changes in service metrics. Although OpenSearch provides powerful search capabilities, the native engine lacks deep integration with cloud-native notification services. CSS addresses this by offering built-in alerting and notification plugins that integrate seamlessly with the Simple Message Notification (SMN) service. This integration enables a highly automated monitoring workflow: monitors are configured to query specified indexes periodically; triggers evaluate the query results against predefined thresholds to identify anomalies and generate alerts; CSS then calls the SMN API to push alerts to subscribers.

How the Feature Works

The open-source OpenSearch alerting plugin (opensearch-alerting) and OpenSearch notifications plugin (opensearch-notifications and opensearch-notifications-core) are built into CSS OpenSearch clusters. They trigger alerts when data meets predefined conditions.

Figure 3-59 Alert notification link



1. A monitor defines the query conditions, such as which index to query and how often.
2. A trigger checks the monitor's query results to determine whether preset conditions that trigger an alert are met (for example, **count > 100**). If yes, it generates an alert and triggers a predefined action.
3. A destination/channel is a message channel that defines the SMN topic where alerts are to be sent.
4. A CSS cluster is granted permission to access SMN through an agency.
5. SMN sends the received alert to subscribers.

For more about the alerting plugin, see **Alerting - OpenSearch Documentation**. For more about the notifications plugin, see **Notifications - OpenSearch Documentation**.

Constraints

Only OpenSearch 1.3.6, 2.19.0, and 3.4.0 clusters provide a built-in alerting plugin. Only clusters of these versions support alert notification via SMN.

- The OpenSearch alerting plugin (opensearch-alerting) is built into OpenSearch 1.3.6 clusters. This plugin consists of three components: **Alerts**, **Monitors**, and **Destinations**. CSS integrates the SMN service in the **Destinations** component, allowing it to send alert notifications to the SMN service as a destination.
- Both the open-source OpenSearch alerting plugin (opensearch-alerting) and OpenSearch notifications plugin (opensearch-notifications and opensearch-

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

notifications-core) are built into OpenSearch 2.19.0 and 3.4.0 clusters (image version 3.4.0_26.6.0_xxx or later). The Destinations component of the OpenSearch alerting plugin is now an independent notification plugin that manages notification channels. CSS integrates the SMN service in its **Notifications** component, so it can use SMN as a notification channel.

Preparations

1. Prepare the needed SMN topic. Create a topic on SMN and add subscribers (mobile number or email address). For details, see **Publishing a Template Message**.
2. Grant SMN access to CSS. You can configure an IAM agency to authorize CSS to access SMN.
   a. Use a CSS administrator account to perform: Log in to the **CSS management console**.

      The CSS administrator account must have the following minimum IAM permissions:

      ```
      "iam:agencies:listAgencies",
      "iam:roles:listRoles",
      "iam:agencies:getAgency",
      "iam:agencies:createAgency",
      "iam:permissions:listRolesForAgency",
      "iam:permissions:grantRoleToAgency",
      "iam:permissions:listRolesForAgencyOnProject",
      "iam:permissions:revokeRoleFromAgency",
      "iam:roles:createRole"
      ```

   b. In the navigation pane, choose **Service Authorization**.
   c. On the **Service Authorization** page, click **Create Agency** for SMN. In the dialog box displayed, confirm that the agency is successfully created.
      - If an agency has already been created, the message "The css_smn_agency already exists. Please check the agency permission." is displayed in the upper right corner, indicating SMN access has been granted to CSS.
      - If you do not have the permission to create an agency, an error message is displayed in the upper right corner indicating "no permission", in which case, check that the administrator account has been assigned the necessary IAM permissions.

Configuring Alert Notifications via SMN (OpenSearch 2.19.0 or 3.4.0)

CAUTION

For OpenSearch 3.4.0, the image version must be 3.4.0_26.6.0_xxx or later to support alert notifications via SMN.

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

   Log in using an account with CSS permissions.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

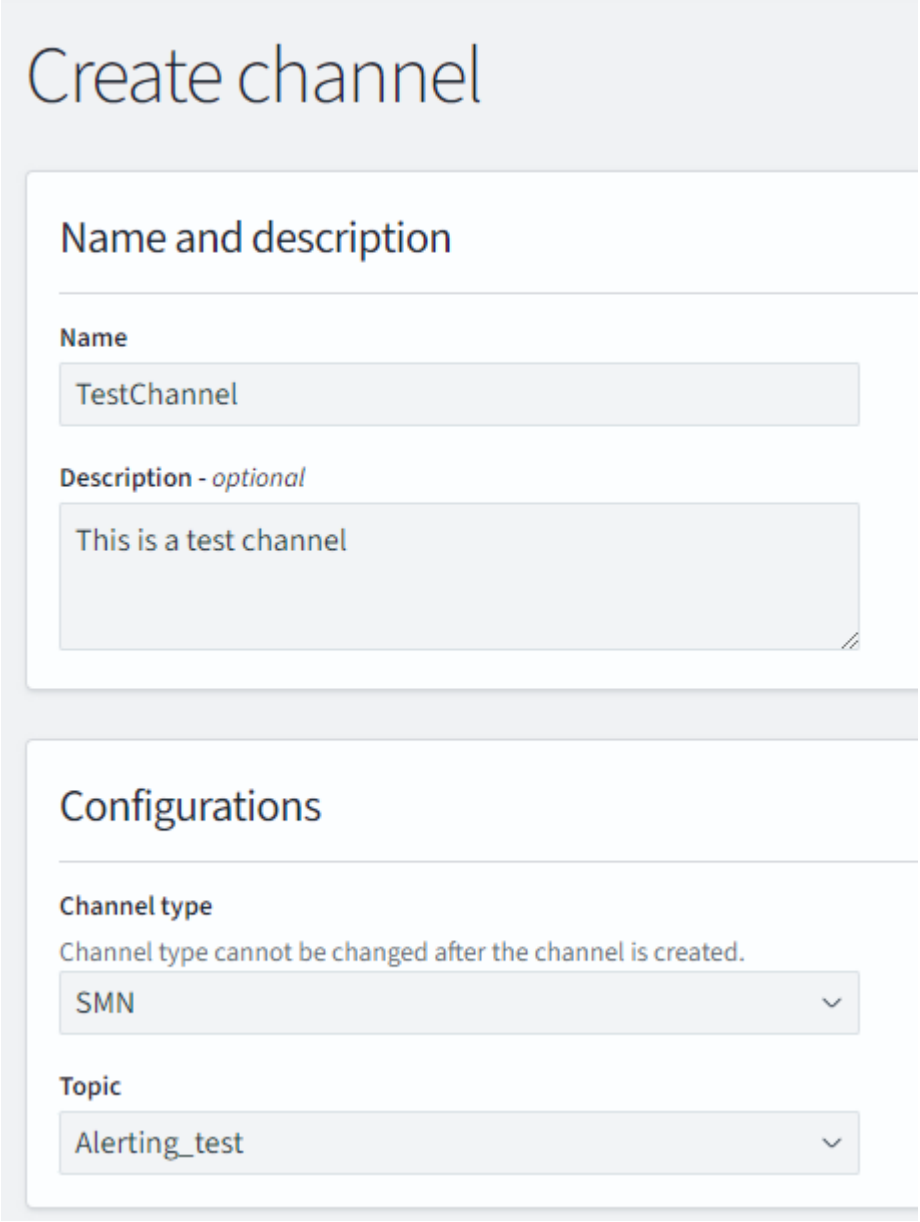
4. On the OpenSearch Dashboards console, expand the menu in the upper-left corner, and choose **Management > Notifications**.
5. Create an SMN channel to send alert messages.
   a. On the **Channels** page, click **Create channel** and configure a notification channel.

Table 3-189 Channel parameters

Parameter	Description
Name	Custom channel name.
Description	Channel description.
Channel type	Type of the notification. Select SMN.
Topic	An associated SMN topic. Select the SMN topic created in **Preparations**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

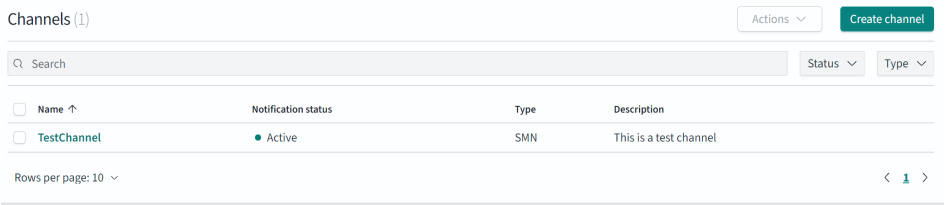
Figure 3-60 Create channel



b. Click **Create**.

c. Return to the **Channels** page. If the new channel is displayed, it has been created successfully.

Figure 3-61 Channels



6. On the OpenSearch Dashboards console, expand the menu in the upper-left corner, and choose **OpenSearch Plugins** > **Alerting**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

7. Create a monitor and trigger to define the alert triggering conditions and monitor interval.

   a. Click the **Monitors** tab on the **Alerting** page and click **Create monitor** to configure monitor information.

Table 3-190 Monitor parameters

Parameter		Description
Monitor details	Monitor name	User-defined monitor name
	Monitor type	Monitor type, which can be: • Per query monitor • Per bucket monitor • Per cluster metrics monitor • Per document monitors • Composite monitors In this example, **Per query monitor** is selected. For more information, see **Monitors** in the OpenSearch official documentation.
	Monitor defining method	Monitor defining method. **Extraction query editor** is recommended. • **Visual editor** • **Extraction query editor** • **Anomaly detector** The options of **Monitor defining method** are determined by the **Monitor type** you selected.
	Detector	If **Monitor defining method** is set to **Anomaly detector**, select an exception detection task.
	Frequency	Select the monitoring frequency and set the monitoring interval. The options include: • **By interval** • **Daily** • **Weekly** • **Monthly** • **Custom cron expression**
Select data	Index	When **Monitor defining method** is set to **Visual editor** or **Extraction query editor**, you need to specify the index to be monitored.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter		Description
	Time field	When **Monitor defining method** is set to **Visual editor**, you need to specify the time field to define counting parameters such as **count**.
Query	Metrics	When **Monitor defining method** is set to **Visual editor**, you need to set the metrics range for extracting statistics.
	Time range for the last	When **Monitor defining method** is set to **Visual editor**, you need to set the monitoring time range for plugins.
	Data filter	When **Monitor defining method** is set to **Visual editor**, you need to set filters for data search.
	Group by	When **Monitor defining method** is set to **Visual editor**, you need to specify a field so that any value of the field will always trigger an alert.
	Define extraction query	When **Monitor defining method** is set to **Extraction query editor**, you need to enter the query statement to define the monitoring.
	Request type	When **Monitor type** is set to **Per cluster metrics monitor**, you need to specify the request type to monitor cluster metrics, such as the running status and CPU usage.
	Preview query and performanc e	Preview the query result and verify query performance under the current configuration.

b. Click **Add trigger** to add triggers and specify the alert triggering conditions and actions to be triggered when an alert is triggered.

c. On the **Triggers** page, set the alert triggering sensitivity and message release on the destination end.

Table 3-191 Trigger parameters

Parameter		Description
Define trigger	Trigger name	User-defined trigger name.

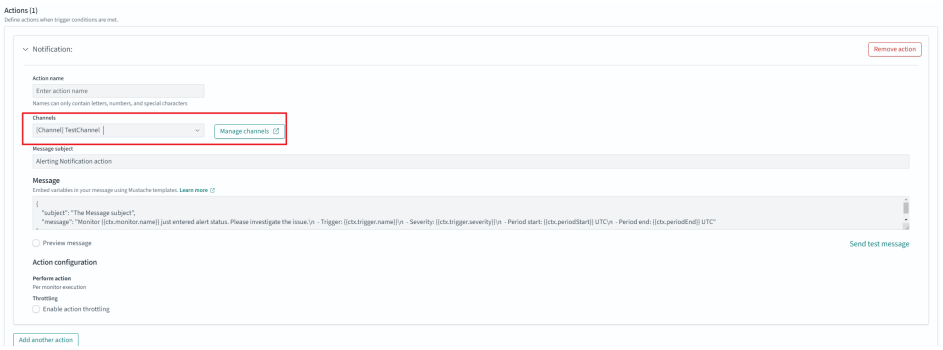
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter		Description
	Severity level	Sensitivity of a trigger, that is, the number of alerts that need to be triggered before a notification is sent. **1** indicates the highest sensitivity.
	Trigger condition	Trigger condition. An alert is triggered when the trigger condition is hit.
Configure actions	Action name	Trigger action name.
	Channels	Select the SMN channel created in **5**.
	Message subject	A description of the message.
	Message	Alert message body. By default, the subject and body are defined when the destination is an email address.
	Perform action	When **Monitor type** is set to **Per bucket monitor**, you need to specify whether to send alerts in combination. The value can be: • **Per execution**: A combination alert is sent when multiple alert triggering conditions are hit. • **Per alert**: Alerts are sent separately when multiple alert triggering conditions are hit.
	Actionable alerts	When **Monitor type** is set to **Per bucket monitor**, and **Perform action** is set to **Per alert**, you need to set the alerts that can be executed after alert triggering conditions are hit. • **De-duplicated**: Alerts that have been triggered. OpenSearch retains the existing alerts to prevent the plugin from generating duplicate alerts. • **New**: Newly created alerts. • **Completed**: Alerts that are no longer ongoing.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter		Description
	Throttling	Specify the message frequency to limit the number of notifications you receive within a given span of time. Without it, high-frequency or low-severity triggers may cause information overload or result in unexpected cloud costs. For example, if this parameter is set to 10 minutes, SMN sends only one alert notification in the next 10 minutes even if the trigger condition is hit multiple times. After 10 minutes, SMN sends another alert notification if the alert condition is met.

Figure 3-62 Setting the destination of a trigger action



d. Click **Send test message** to test the alert notification link.

e. As shown in **Figure 3-64**, if a subscriber to the SMN topic receives the message, alert notification is configured successfully.

Figure 3-63 Sending a test message

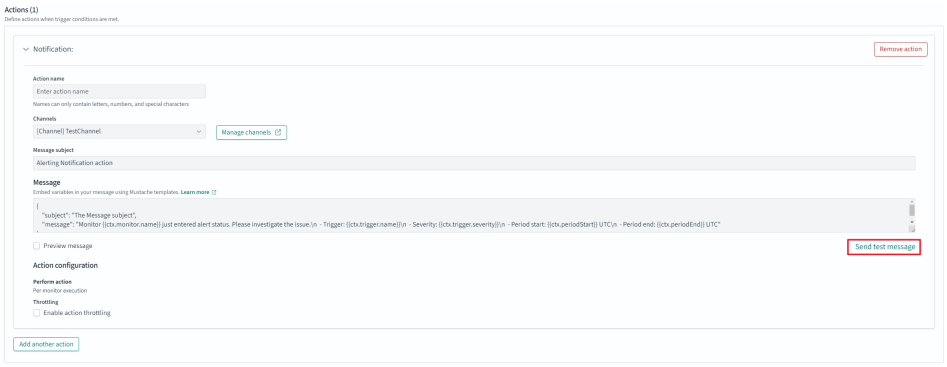
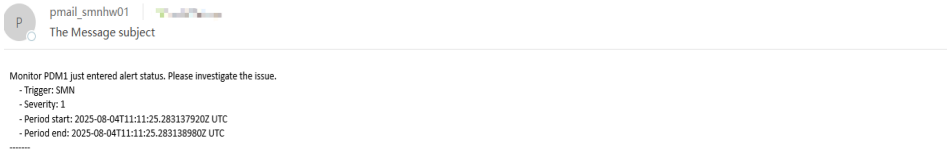


Figure 3-64 Email notification



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

f. Click **Create** to go to the monitor details page.

Configuring Alert Notifications via SMN (OpenSearch 1.3.6)

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

   Log in using an account with CSS permissions.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. On the **OpenSearch Dashboards** page, choose **OpenSearch Plugins** > **Alerting** in the navigation tree on the left.
5. Create an SMN destination to send alert messages.

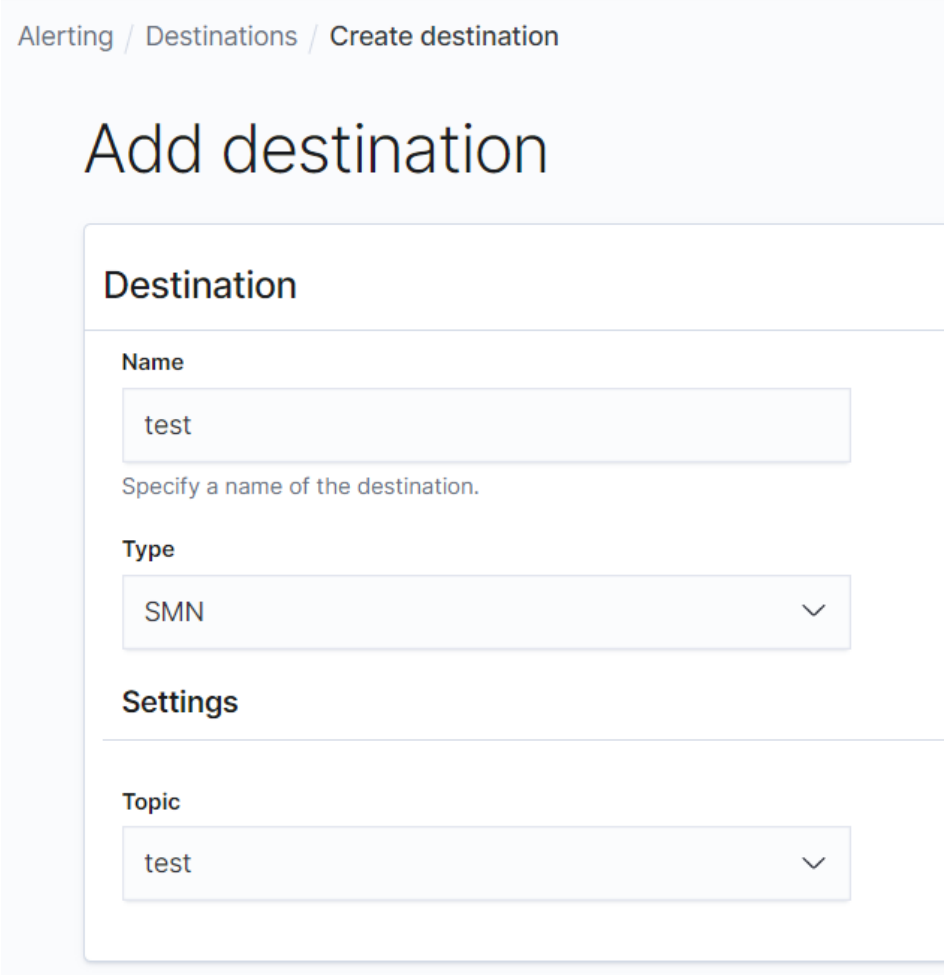
   a. On the **Alerting** page, click the **Destinations** tab, and click **Add destination** to configure destination information.

   Table 3-192 Destinations parameters

Parameter	Description
Name	User-defined destination name
Type	Type of the notification. Select SMN.
Topic	An associated SMN topic. Select the SMN topic created in **Preparations**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-65 Add destination



b. Click Create to create destinations.

c. Return to the **Destinations** page. If the new channel is displayed, it has been created successfully.

Figure 3-66 Destination list



6. Create a monitor and trigger to define the alert triggering conditions and monitor interval.

   a. Click the **Monitors** tab on the **Alerting** page and click **Create monitor** to configure monitor information.

Table 3-193 Monitor parameters

Parameter		Description
Monitor details	Monitor name	User-defined monitor name

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter		Description
	Monitor type	Monitor type, which can be: • Per query monitor • Per bucket monitor • Per cluster metrics monitor • Per document monitors • Composite monitors In this example, **Per query monitor** is selected. For more information, see **Monitors** in the OpenSearch official documentation.
	Monitor defining method	Monitor defining method. **Extraction query editor** is recommended. • **Visual editor** • **Extraction query editor** • **Anomaly detector** The options of **Monitor defining method** are determined by the **Monitor type** you selected.
	Detector	If **Monitor defining method** is set to **Anomaly detector**, select an exception detection task.
	Frequency	Select the monitoring frequency and set the monitoring interval. The options include: • **By interval** • **Daily** • **Weekly** • **Monthly** • **Custom cron expression**
Data source	Index	When **Monitor defining method** is set to **Visual editor** or **Extraction query editor**, you need to specify the index to be monitored.
	Time field	When **Monitor defining method** is set to **Visual editor**, you need to specify the time field to define counting parameters such as **count**.
Query	Metrics	When **Monitor defining method** is set to **Visual editor**, you need to set the metrics range for extracting statistics.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter		Description
	Time range for the last	When **Monitor defining method** is set to **Visual editor**, you need to set the monitoring time range for plugins.
	Data filter	When **Monitor defining method** is set to **Visual editor**, you need to set filters for data search.
	Group by	When **Monitor defining method** is set to **Visual editor**, you need to specify a field so that any value of the field will always trigger an alert.
	Define extraction query	When **Monitor defining method** is set to **Extraction query editor**, you need to enter the query statement to define the monitoring.
	Request type	When **Monitor type** is set to **Per cluster metrics monitor**, you need to specify the request type to monitor cluster metrics, such as the running status and CPU usage.

b. Click **Add trigger** to add triggers and specify the alert triggering conditions and actions to be triggered when an alert is triggered.

c. On the **Triggers** page, set the alert triggering sensitivity and message release on the destination end.

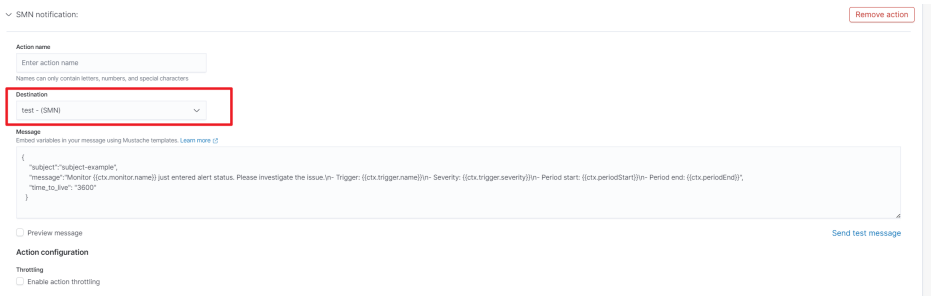
Table 3-194 Trigger parameters

Parameter		Description
Define trigger	Trigger name	User-defined trigger name
	Severity level	Sensitivity of a trigger, that is, the number of alerts that need to be triggered before a notification is sent. **1** indicates the highest sensitivity.
	Trigger condition	Trigger condition. An alert is triggered when the trigger condition is hit.
Configure actions	Action name	Trigger action name
	Destination	Select the SMN destination created in **5**.
	Message	Alert message body By default, the subject and body are defined when the destination is an email address.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter		Description
	Perform action	When **Monitor type** is set to **Per bucket monitor**, you need to specify whether to send alerts in combination. The value can be: • **Per execution**: A combination alert is sent when multiple alert triggering conditions are hit. • **Per alert**: Alerts are sent separately when multiple alert triggering conditions are hit.
	Actionable alerts	When **Monitor type** is set to **Per bucket monitor**, and **Perform action** is set to **Per alert**, you need to set the alerts that can be executed after alert triggering conditions are hit. • **De-duplicated**: Alerts that have been triggered. OpenSearch retains the existing alerts to prevent the plugin from generating duplicate alerts. • **New**: Newly created alerts. • **Completed**: Alerts that are no longer ongoing.
	Throttling	Specify the message frequency to limit the number of notifications you receive within a given span of time. Without it, high-frequency or low-severity triggers may cause information overload or result in unexpected cloud costs. For example, if this parameter is set to 10 minutes, SMN sends only one alert notification in the next 10 minutes even if the trigger condition is hit multiple times. After 10 minutes, SMN sends another alert notification if the alert condition is met.

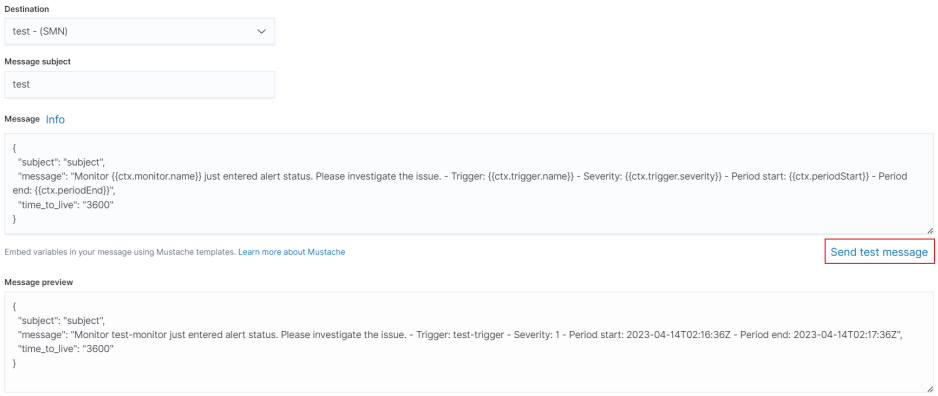
Figure 3-67 Setting the destination of a trigger action



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

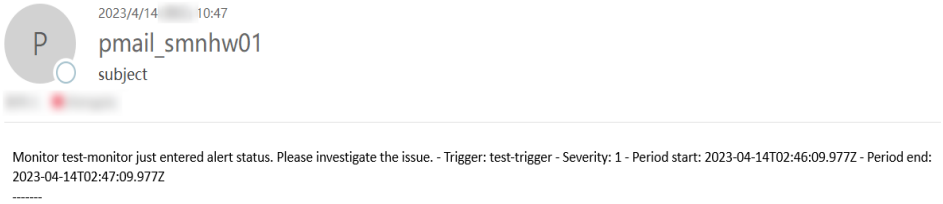
d. Click **Send test message** to test the alert notification link.

Figure 3-68 Sending a test message



e. As shown in **Figure 3-69**, if a subscriber to the SMN topic receives the message, alert notification is configured successfully.

Figure 3-69 Email notification



f. Click **Create** to go to the monitor details page.

3.14.2 Synchronizing Alerts and Metrics to Prometheus

When building a full-stack monitoring system, O&M teams often use Prometheus to aggregate infrastructure and application metrics. However, business-level data stored in OpenSearch (such as user access patterns and transaction logs) remains in document form and is not natively available as Prometheus-format time series, hindering unified analysis and alerting in Grafana dashboards. How can we expose OpenSearch aggregation results as Prometheus-compatible time series? CSS OpenSearch clusters address this gap using built-in alerting and notifications plugins, which provide a Prometheus-compatible export channel. Using the Prometheus Pushgateway, they periodically execute queries, transform the results into structured metrics, and push them to the Pushgateway where they can be scraped by Prometheus. This approach enables OpenSearch clusters to act as intelligent metric sources, supporting unified analysis of business and infrastructure data and consistent, threshold-based alerting within a single observability stack.

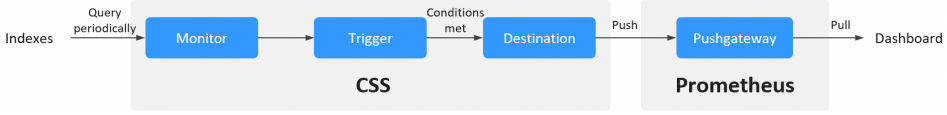
How the Feature Works

OpenSearch business metrics must be pushed by the alerting plugin, whereas Prometheus uses a pull-based model by default. The Prometheus Pushgateway bridges this gap: it receives push-based metrics and exposes them via a scrape endpoint for Prometheus. The OpenSearch alerting plugin (opensearch-alerting) and OpenSearch notifications plugin (opensearch-notifications) built into CSS

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

OpenSearch clusters provide a Prometheus-compatible channel connected to the Prometheus Pushgateway.

Figure 3-70 Alert & metric synchronization link



1. **Monitor**: Defines the aggregation query (which metrics to compute), the target indexes, and the execution schedule (query frequency).
2. **Trigger**: Evaluates query results against threshold conditions to determine if and when an alert should be generated. It also triggers an action. **Action**: Transforms OpenSearch aggregation results (in document form) into standard Prometheus JSON format.
3. **Channel**: A message channel that sends Prometheus-compatible JSON packets to the Pushgateway.
4. **Pushgateway**: A temporary storage area that receives push-based metrics from OpenSearch.
5. **Prometheus**: Periodically pulls data from the Pushgateway and stores it in its time-series database, where Grafana queries the data and visualizes metrics on unified dashboards.

For more about the alerting plugin, see **Alerting - OpenSearch Documentation**. For more about the notifications plugin, see **Notifications - OpenSearch Documentation**. For more about Prometheus Pushgateway, see **Prometheus Pushgateway**.

Constraints

- Alerts and metrics can be synchronized to Prometheus only when the OpenSearch cluster version is 2.19.0 or 3.4.0 (image version 3.4.0_26.6.0_xxx or later).
- The OpenSearch cluster must be able to access the Prometheus Pushgateway endpoint. Prometheus and OpenSearch must be connected. Otherwise, alerts cannot be sent.
- Only Gauge-type metrics (used in dashboards) can be synchronized. They include statistical or percentage values that can both increase and decrease.

Configuring Alert Synchronization

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. On the **OpenSearch Dashboards** page, expand the menu in the upper-left corner, and choose **Management > Notifications**.
5. Create a Prometheus channel to establish a physical connection between OpenSearch and Pushgateway.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

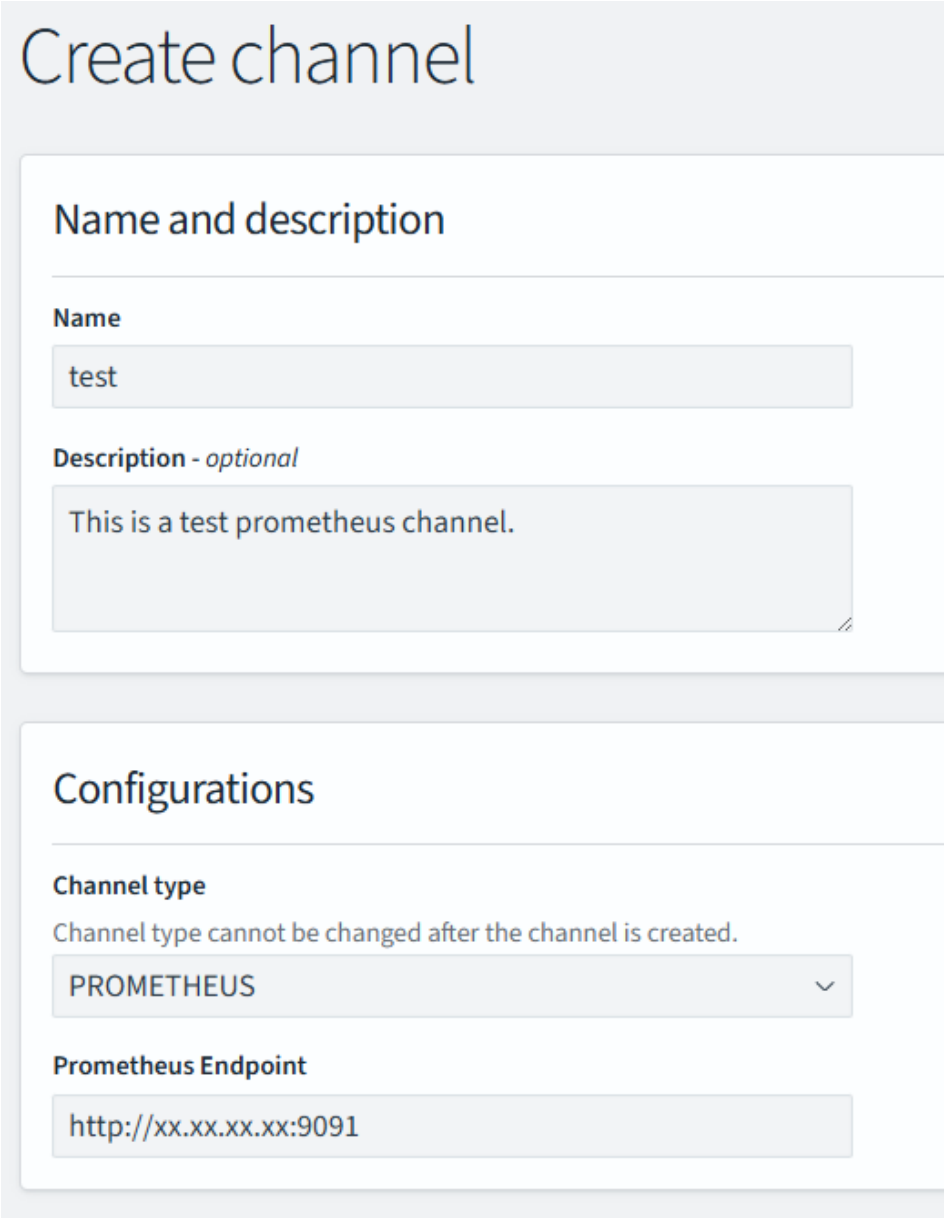
a. On the **Channels** page, click **Create channel** to configure a channel.

Table 3-195 Channel parameters

Parameter	Description
Name	Custom channel name.
Description	Channel description.
Channel type	Type of the notification. Select **PROMETHEUS**.
Prometheus Endpoint	Enter the Pushgateway address of the Prometheus monitor server. • Currently, only Prometheus Gauge dashboards can be added or deleted. Metrics are queried using specific statements and numeric values are synchronized to Pushgateway for monitoring via Prometheus. • Two types of Pushgateway addresses are supported: HTTP and HTTPS.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

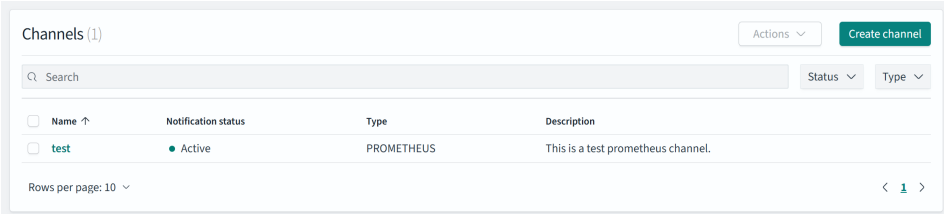
Figure 3-71 Create channel



b. Click **Create**.

c. Return to the **Channels** page. If the newly created Prometheus channel is displayed, it has been created successfully.

Figure 3-72 Channels list



6. On the OpenSearch Dashboards console, expand the menu in the upper-left corner, and choose **Alerting**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

7. Create a monitor and trigger to define the alert triggering conditions and monitor interval.

   a. On the **Alerting** page, click the **Monitors** tab. Then click **Create monitor**, and configure the data source and query frequency.

   Table 3-196 Monitor parameters

Parameter		Description
Monitor details	Monitor name	User-defined monitor name
	Monitor type	Monitor type, which can be: • Per query monitor • Per bucket monitor • Per cluster metrics monitor • Per document monitors • Composite monitors In this example, **Per query monitor** is selected. For more information, see **Monitors** in the OpenSearch official documentation.
	Monitor defining method	Monitor defining method. **Extraction query editor** is recommended. • **Visual editor** • **Extraction query editor** • **Anomaly detector** The options of **Monitor defining method** are determined by the **Monitor type** you selected.
	Detector	If **Monitor defining method** is set to **Anomaly detector**, select an exception detection task.
	Frequency	Select the monitoring frequency and set the monitoring interval. The options include: • **By interval** • **Daily** • **Weekly** • **Monthly** • **Custom cron expression**
Select data	Index	When **Monitor defining method** is set to **Visual editor** or **Extraction query editor**, you need to specify the index to be monitored.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter		Description
	Time field	When **Monitor defining method** is set to **Visual editor**, you need to specify the time field to define counting parameters such as **count**.
Query	Metrics	When **Monitor defining method** is set to **Visual editor**, you need to set the metrics range for extracting statistics.
	Time range for the last	When **Monitor defining method** is set to **Visual editor**, you need to set the monitoring time range for plugins.
	Data filter	When **Monitor defining method** is set to **Visual editor**, you need to set filters for data search.
	Group by	When **Monitor defining method** is set to **Visual editor**, you need to specify a field so that any value of the field will always trigger an alert.
	Define extraction query	When **Monitor defining method** is set to **Extraction query editor**, you need to enter the query statement to define the monitoring.
	Request type	When **Monitor type** is set to **Per cluster metrics monitor**, you need to specify the request type to monitor cluster metrics, such as the running status and CPU usage.
	Preview query and performance	Preview the query result and verify query performance under the current configuration.

b. Click **Create**. The **Create trigger** page is displayed.

c. On the **Create trigger** page, define how to convert data into Prometheus-readable formats.

Table 3-197 Trigger parameters

Parameter		Description
Define trigger	Trigger name	User-defined trigger name.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter		Description
	Severity level	Sensitivity of a trigger, that is, the number of alarms that need to be triggered before an alarm message is sent. **1** indicates the highest sensitivity.
	Trigger condition	Trigger condition. An alarm is triggered when the trigger condition is hit. You are advised to set a trigger condition that can almost always be triggered (for example, **1 > 0**) so that the query results will always be synchronized to the Pushgateway.
Configure actions	Action name	Trigger action name.
	Destination	Select the destination created in **5**.

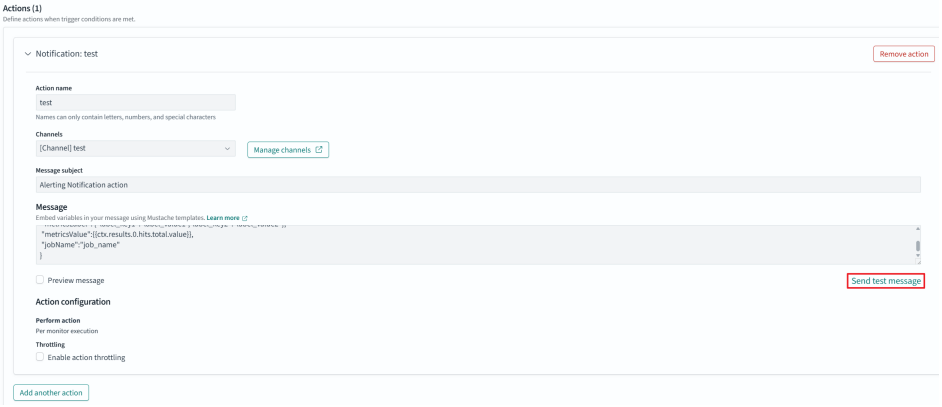
	Message	Defines the body of the message to be published, which must use the JSON format.
		Message template:
		{
		"metricsName":"hits_total_value", //Prometheus metric name
		"metricsLabel": {"label_key1":"label_value1","label_key2":"label_value2"}, //Prometheus labels
		"metricsValue":{{ctx.results.0.hits.total.value}}, //Prometheus metric values
		"jobName":"job_name", //Prometheus monitor task name
		"metricsHelp":"***" //Metric explanation. Optional.
		}

	throttling	Specify the message frequency to limit the number of notifications you receive within a given span of time. Without it, high-frequency or low-severity triggers may cause information overload or result in unexpected cloud costs. For example, if this parameter is set to 10 minutes, Prometheus sends only one alert notification in the next 10 minutes even if the trigger condition is met multiple times. After 10 minutes, Prometheus sends another alert notification if the trigger condition is met again.

d. Click **Send test message** to test the link to Prometheus.

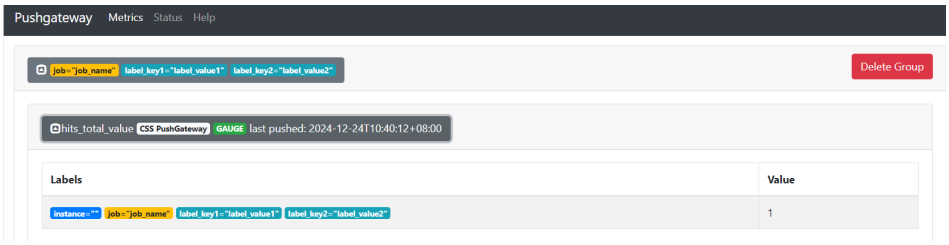
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-73 Send a test message.



e. As shown in **Figure 3-74**, Prometheus can receive a triggered message, meaning the trigger is set successfully.

Figure 3-74 Message received successfully



f. Click **Create** to go to the monitor details page.

3.15 Intelligent Diagnostics

During routine operations, you must periodically monitor the health of your OpenSearch clusters to ensure service stability. However, manual health checks are time-consuming, labor-consuming, and prone to oversight. Key issues may be easily missed. To address this, CSS provides an Intelligent O&M feature. It supports both scheduled and on-demand diagnostics, covering essential checks such as data node disk usage, cluster health status, and node CPU usage. Through these checks, O&M personnel can detect and handle potential risks in a timely manner to ensure cluster reliability and availability. By default, Intelligent O&M retains the most recent diagnostic reports, so you can always review the latest results. You can also export any report for further analysis or long-term archiving.

Constraints

- Intelligent diagnostics cannot be enabled while the cluster is in one of the following **Cluster Status**: frozen, creating, or creation failed.
- While a diagnostic task is in progress, you cannot manually start another task.
- Running diagnostics on a cluster that is in the process of a configuration change (for example, being scaled up or down, or having its node specifications changed) may lead to inaccurate results. You are advised to avoid such times.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- When intelligent diagnostics is enabled, the system is allowed by default to read cluster settings and data configuration information for diagnostic or analytical purposes.

Diagnostic Modes

Two diagnostic modes are available: scheduled and manual. Choose one based on your service requirements.

Table 3-198 Comparing the two diagnostic modes

Dimension	Scheduled Diagnostics	Manual Diagnostics
Triggering method	Automatically performed daily by schedule	Manually triggered and executed right away
When to use	Routine inspection, creating a health baseline, and long-term trend observation	Quick troubleshooting in response to cluster failures or errors, or cluster status verification before and after change operations
Typical users	O&M teams (routine maintenance)	R&D or O&M personnel (troubleshooting)
Diagnostic type	• Full check • Cluster unavailability check	• Full check • Cluster unavailability check

Configuring Scheduled Diagnostics

Configuring daily scheduled diagnostics enables continuous health risk monitoring for clusters without human intervention.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Intelligent O&M > Intelligent Diagnostics**.
5. On the **Intelligent Diagnostics** page, click **Enable Scheduled Diagnostics**. In the displayed dialog box, configure a scheduled diagnostics task.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-199 Enabling scheduled diagnostics

Parameter	Description
Diagnostic Type	Select a diagnostic type. • **Full check**: All check items are supported. You can select all or only some of the check items. • **Cluster unavailability check**: When a cluster is unavailable, run this diagnosis to quickly diagnose its health status.
Diagnostic Item	Select diagnostic items to be included in the scheduled task. • When **Diagnostic Type** is set to **Full check**, you can customize the diagnostic items. • When **Diagnostic Type** is set to **Cluster unavailability check**, the cluster unavailability check items are selected by default.
Time Zone	Select your current time zone. The start time of the scheduled diagnostic task is affected by it.
Diagnosed	Select the time to start the scheduled diagnostic task every day.

6. Click **OK** to save the task information.

   The scheduled task information is displayed on the **Intelligent Diagnostics** page, and daily reports generated by the task will be displayed below on this page or on the **Historical Reports** tab.

   Figure 3-75 Viewing a scheduled diagnostics task

   The report will show high-risk items, medium-risk items, normal items, and reminders, as well as the results and optimization suggestions for each diagnostic item. High- and medium-risk items should be prioritized.

7. On the **Intelligent Diagnostics** page, manage the scheduled diagnostic task, including modifying and disabling it.
   - Modify the scheduled diagnostic task: Click **Modify Settings**. In the displayed dialog box, modify the settings and click **OK** to update the scheduled diagnostic policy.
   - Disable the scheduled diagnostic task: Click **Disable Scheduled Diagnostics**. In the displayed dialog box, click **OK** to disable scheduled diagnostics. The scheduled task is removed from the Intelligent Diagnostics page.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Performing Manual Diagnostics

Perform a one-time manual diagnostic for quick troubleshooting or to verify cluster status. This is recommended when your cluster experiences performance issues or errors, or after changes are made to your cluster.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Intelligent O&M > Intelligent Diagnostics**.
5. On the **Intelligent Diagnostics** page, configure a diagnostic task and click **Run Diagnostics** to start the task.

Figure 3-76 Manual diagnostics

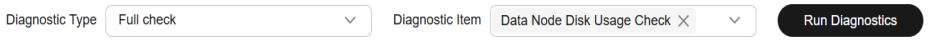


Table 3-200 Manual diagnostics

Parameter	Description
Diagnostic Type	Select a diagnostic type. • **Full check**: All check items are supported. You can select all or only some of the check items. • **Cluster unavailability check**: When a cluster is unavailable, run this diagnosis to quickly diagnose its health status.
Diagnostic Item	Select diagnostic items. • When **Diagnostic Type** is set to **Full check**, you can customize the diagnostic items. • When **Diagnostic Type** is set to **Cluster unavailability check**, the cluster unavailability check items are selected by default.

6. Wait for 1 to 3 minutes. The diagnostic report will be displayed below.

   The report will show high-risk items, medium-risk items, normal items, and reminders, as well as the results and optimization suggestions for each diagnostic item. High- and medium-risk items should be prioritized.
7. Click **Export Report** to save the diagnostic report to your local PC for further analysis or archiving.

Viewing Historical Reports

Historical diagnostic reports help you compare cluster health across different periods, which aids capacity planning and problem diagnosis.

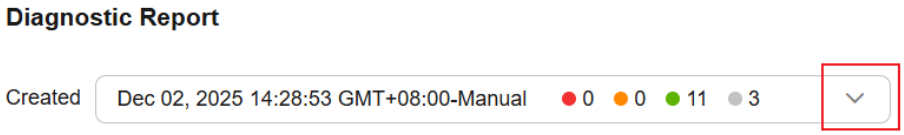
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

CAUTION

A maximum of 10 historical reports can be retained. (This is the total for both scheduled and manual diagnostic tasks.) When this limit is exceeded, the oldest reports will be deleted automatically. Export and save important reports in a timely manner.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Intelligent O&M > Historical Reports**.
5. Set **Created** to filter diagnostic reports by time range.

   Figure 3-77 Selecting historical reports

6. Click **Export Report** on the right to save the current historical report to the local PC.

Diagnostic Items

Table 3-201 lists the diagnostic items supported by Intelligent O&M.

Table 3-201 Diagnostic items

No.	Diagnostic Item	Description
1	Data Node Disk Usage Check	Identify data nodes (including cold data nodes) with the highest disk usage in the cluster. Excessive disk usage can affect cluster stability. • High risk: When a node's disk usage reaches 85%, no new index replicas will be allocated to it. • Medium risk: When a node's disk usage exceeds 80%, take immediate action to prevent further increase.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

No.	Diagnostic Item	Description
2	Data Node Disk Usage Balance Check	Check whether disk usage is balanced across all regular data nodes and cold data nodes. Unbalanced disk usage may cause performance issues. ● High risk: Maximum disk usage difference between nodes ≥ 40% ● Medium risk: 25% ≤ Maximum disk usage difference between nodes < 40%
3	Large Index Shard Check	Check for large index shards that may cause cluster load imbalance. ● High risk: Shard storage ≥ 50 GB; and the index only has one primary shard. ● Medium risk: Shard storage ≥ 50 GB; and the number of shards is not divisible by the number of data nodes or cold data nodes.
4	Cluster Node Configuration Check	Check the high availability settings of clusters. ● Single/Dual Node Risk: For clusters without a dedicated Master node, at least three Data nodes (including cold data nodes) are required to ensure cluster availability during a single node failure. ● Overloaded Master Node Risk: When there are 10 or more Data nodes, dedicated Master nodes should be configured to prevent excessive load on the Master node. ● Single Client Node Risk: A cluster containing only one dedicated Client node may result in cluster inaccessibility if that single Client node fails.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

No.	Diagnostic Item	Description
5	Cluster Health Check	Check the overall health status of all indices in the cluster. ● High risk (red): There are unallocated primary shards (i.e., some indices are unavailable). ● Medium risk (yellow): Primary shards are properly allocated, but replicas are missing (compromised data availability). ● Normal (green): All primary and replica shards are properly allocated.
6	Maximum Per-Node Shard Quantity Check	Check that the number of shards allocated to each node remains within the limit set by cluster.max_shards_per_node, preventing shard allocation failures. ● High risk: Maximum number of shards allocated to a node ≥ 90% of the per-node shard limit ● Medium risk: 85% of the per-node shard limit ≤ Maximum number of shards allocated to a node < 90% of the per-node shard limit
7	Oversized Shard Check	Check for shards that have reached or exceeded 100 GB in the cluster. Oversized shards can increase query latency and slow down fault recovery. You are advised to keep the shard size under 50 GB. ● High risk: Maximum shard size ≥ 500 GB ● Medium risk: 100 GB ≤ Maximum shard size < 500 GB
8	Hot Index Read/Write Check (Every Minute)	Identify the 10 indices with the most query and write requests within the last minute and display them in descending order. For each of these identified indices, verify that the number of its primary shards is divisible by the number of data nodes or cold data nodes to prevent load imbalance.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

No.	Diagnostic Item	Description
9	Node Disconnection Check	Check the connectivity of cluster nodes. Node disconnection will severely compromise cluster performance and therefore needs to be handled immediately.
10	Balanced Node Connections Check	Check the number of connections to port 9200 on the client nodes (or data nodes if there are no client nodes), and display nodes by their connection count in descending order.
11	Client Connection Check (Source IP Address Analysis)	Check the number of connections to port 9200 on the client nodes (or data nodes if there are no client nodes), and display all external source IP addresses by connection count in descending order as well as details about the top 10 connections.
12	Node CPU Usage Check (Interval: 1 Minute)	Measure the CPU usage of each node twice every minute, and calculate the average for each node to evaluate their load status. ● High risk: Average CPU usage ≥ 90% or no data is obtained. ● Medium risk: 80% ≤ Average CPU usage < 90%
13	JVM Heap Memory Usage Check (Interval: 1 Minute)	Measure the JVM heap memory usage of each node twice every minute, and calculate the average for each node to evaluate their load status. ● High risk: Average JVM heap memory usage ≥ 90% or no data is obtained. ● Medium risk: 80% ≤ Average JVM heap memory usage < 90%
14	Local-Disk Cluster Index Replica Check	For a cluster consisting of nodes that use local disks, check whether all indices have replicas. Otherwise, a single disk failure can lead to permanent data loss.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3.16 Managing Cluster Changes

3.16.1 Scaling Out/Up

If an OpenSearch cluster struggles to maintain performance in the face of rapid data growth or sustained high memory usage, you may scale it horizontally by adding more nodes and more node types, or do so vertically by increasing the storage capacity of existing nodes.

Table 3-202 Scaling scenarios

Type	Scenario	Change Process
Adding new nodes	If a cluster experiences rapid data growth or sustained high memory usage, you may add more nodes to it to improve cluster availability.	Procedure for adding data nodes or cold data nodes: 1. New nodes are added and cluster configuration is modified. 2. A shard redistribution is automatically triggered to reallocate certain shards to newly added nodes. 3. After the redistribution is complete, the new nodes start to handle query and indexing requests. Adding master or client nodes does not trigger data migration. During the capacity expansion, the system ensures that each shard has at least one available replica to ensure service continuity.
Adding master nodes	As your cluster's data volume grows, you can improve its stability and reliability by adding dedicated master nodes for more effective cluster node management.	Procedure for adding a new node type: Add new nodes and modify cluster configuration. Adding master nodes does not trigger data migration, and therefore does not interrupt services.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Type	Scenario	Change Process
Adding client nodes	To handle increased query and write loads as your cluster's data volume grows, you can add dedicated client nodes. This offloads request processing from your data nodes, improving overall cluster scalability.	Procedure for adding a new node type: Add new nodes and modify cluster configuration. Adding client nodes does not trigger data migration, and therefore does not interrupt services. However, when client nodes are added, the cluster's address changes from the data node address to the client node address. You must update the client configuration to use this new address. Otherwise, the client nodes cannot function.
Adding cold data nodes	As your cluster's data volume keeps growing, you may have large quantities of historical data that is infrequently accessed but still useful for analytical purposes. To optimize both storage costs and query performance, you can add cold data nodes to store such data.	Procedure for adding a new node type: Add new nodes and modify cluster configuration. Adding cold data nodes does not trigger data migration, and therefore does not interrupt services.
Increasing node storage capacity	• If a cluster experiences rapid data growth and its storage capacity becomes insufficient, you can expand the cluster nodes' storage capacity. • If the disk usage of a cluster remains high, you can expand the cluster nodes' storage capacity.	1. A request is sent to the EVS service to expand the disk capacities for all cluster nodes. 2. After disk capacity is expanded by EVS, the size of existing disk volumes is expanded on all cluster nodes. The disk capacity expansion will not interrupt ongoing services.

Impact on Billing

For a pay-per-use cluster, you can see its new price when confirming the scale-out or scale-up on the console. After the change is complete, the cluster will be billed based on the new price. For details, see **Cloud Search Service Price Calculator**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Constraints

- The node storage capacity of a cluster can only be increased—not decreased. Select an appropriate node storage capacity based on the data volume and projected data growth.
- You cannot increase the number of nodes and node storage capacity for a cluster at the same time.
- The storage capacity of the master and client nodes in a cluster cannot be expanded.
- The storage capacity of data nodes that use local disks cannot be expanded.
- For a multi-AZ cluster, the total number of client nodes must be greater than or equal to the number of AZs. (For a cluster spanning two AZs, this means a minimum of two client nodes; for a three-AZ deployment, a minimum of three client nodes.)
- Principles for adding cold data nodes to a multi-AZ cluster:
  - Node quantity requirement: A two-AZ cluster **must** have at least four cold data nodes (two in each AZ); and a three-AZ cluster **must** have at least three cold data nodes (one in each AZ).
  - Performance enhancement: To prevent uneven data distribution and ensure optimal query and ingestion performance, the number of cold data nodes should be an integer multiple of the number of AZs.
- For the range of node quantities supported by each node type, see **Table 3-203**.

Table 3-203 Node quantity ranges

Node Type	Value Range
Data nodes	• Without master nodes: 1 to 32 • With master nodes: 1 to 200 (If the combined number of data nodes and cold data nodes exceeds 100, you are advised to select a flavor with at least 16 vCPUs. This helps ensure optimal read and write performance.)
Master nodes	3, 5, 7, or 9 (must be an odd number from 3 to 9)
Client nodes	1~64
Cold data nodes	1–32

Change Impact

Before the change, learn about possible impacts and operation suggestions, and develop a plan to minimize these impacts.

Expanding the storage capacity of cluster nodes (vertical scaling) does not impact services, whereas adding new nodes or new node types (horizontal scaling) may have the following impacts:

- Impact on performance

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Adding new nodes does not interrupt services. However, after new nodes are added, data shards need to be redistributed to these nodes to balance the load, and this process will consume I/O performance. This is why you are advised to perform the operation during off-peak hours.

To minimize this impact, it is advisable to adjust the data migration rate based on the cluster's traffic cycle: increase the data migration rate during off-peak hours to shorten the task duration, and decrease it **before** peak hours arrive to ensure optimal cluster performance. The data migration rate is determined by the **indices.recovery.max_bytes_per_sec** parameter. The default value of this parameter is the number of vCPUs multiplied by 8 MB. For example, for four vCPUs, the data migration rate is 32 MB/s. You can adjust it based on the service requirements.

```
PUT /_cluster/settings
{
  "transient": {
    "indices.recovery.max_bytes_per_sec": "128MB"
  }
}
```

- Characteristics of this process

  Once started, a scaling task cannot be stopped until it succeeds or fails.

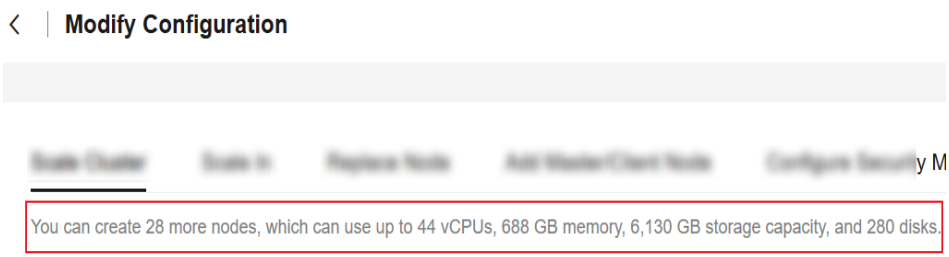
Duration

- It takes 10 to 30 minutes to add new nodes or new node types (horizontal scaling), depending on the cluster's scheduling capabilities.
- It takes 10 to 15 minutes to expand the storage capacity of cluster nodes (vertical scaling).

Prerequisites

- The cluster status is **Available**, and there are no ongoing tasks.
- Your CSS resource quotas are sufficient for the capacity expansion you are about to perform. You can check available resources on the **Modify Configuration** page.

  Figure 3-78 Checking available resources



Adding More Nodes or Increasing Node Storage Capacity

If a cluster experiences rapid data growth or sustained high memory usage, you may add more nodes to it or increase the cluster nodes' storage capacity. This helps to improve service and data availability and durability.

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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. Click the **Scale Cluster** tab.
5. Set the necessary parameters.

   Table 3-204 Adding more nodes or increasing node storage capacity

Parameter	Description
Action	Select **Scale out**.
Resources	Quantities of resources added.
Nodes	• Select the node type you want to scale in the **Node Type** column. Only one node type can be selected at a time. • In the **Nodes** column, increase the number of nodes. For the range of node quantities supported by each node type, see **Constraints**. • In the **Node Storage Type** column, increase the node storage capacity. The value range of node storage capacity is determined by the node flavor you select. The value must be divisible by 20. You cannot increase the number of nodes and node storage capacity for a cluster at the same time.

6. Click **Next**.
7. Confirm the information and click **Submit**.
8. Click **Back to Cluster List** to go back to the **Clusters** page. The **Task Status** is **Scaling out**. When **Cluster Status** changes to **Available**, the cluster has been successfully scaled out.

Adding New Node Types

For a cluster that has no dedicated master, client, or cold data nodes, as the load on its data plane increases, you may dynamically add such nodes to it to share the load of the data nodes.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. Click the **Add Node Types** tab.

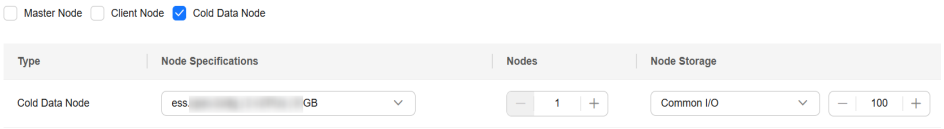
   If a cluster already has dedicated master, client, and cold data nodes, the **Add Node Types** tab is unavailable.
5. On the **Add Node Types** tab, configure the nodes.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-205 Adding new node types

Parameter	Description
Node types	Select the type of nodes you want to add. • Only the node types that the cluster does not already have are displayed. For example, if the cluster already has dedicated master nodes, only **Client Node** and **Cold Data Node** are displayed. • Only one node type can be added at a time. For example, if you need to add both master and client nodes, you need to perform this task twice.
Node Specifications	Select node specifications based on site requirements.
Nodes	Set the number of nodes to add for this node type. For the value range, see **Constraints**.
Node Storage Type	Set the node storage type. The storage capacity per master or client node is fixed to 40 GB.

Figure 3-79 Adding new node types



6. Click **Next**.
7. Confirm the information and click **Submit**.

   Return to the cluster list page. The **Task Status** of the cluster is **Scaling out**. When **Cluster Status** changes to **Available**, nodes have been added successfully.

   - When adding client nodes, you can choose whether to restart data nodes and cold data nodes and bring the Cerebro and OpenSearch Dashboards processes offline after **Cluster Status** changes to **Available**. This helps avoid conflicts.
   - When adding cold data nodes, you must restart data nodes to activate the hot tag after **Cluster Status** changes to **Available**. This ensures that the cold data nodes function properly. After cold data nodes are added, manually migrate historical data to them. For details, see **Switching Between Hot and Cold Data**.

   For details about the restart operation, see **Restarting a Cluster**.

Related Documents

- You can also upgrade the node specifications and change the EVS disk type of an OpenSearch cluster. For details, see **Changing Node Specifications**.
- During a capacity expansion, if the resources in an AZ become insufficient, you can migrate the nodes in this AZ to another that has sufficient resources,

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

and then perform the capacity expansion in the new AZ. For details, see **Switching AZs**.

3.16.2 Scaling In/Down

If an OpenSearch cluster has excess capacity due to off-peak traffic or reduced data volumes, you can reduce its nodes to optimize costs.

Table 3-206 Scaling scenarios

Type	Scenario	Change Process
Removing nodes randomly	Randomly removes cluster nodes to optimize costs.	1. Migrate the shards on the to-be-removed nodes to the remaining nodes. 2. After data migration, bring the nodes offline and modify the cluster configuration. Nodes are removed one at a time, so as to avoid interrupting services.
Removing specified nodes	Removes specified cluster nodes to optimize costs.	

Impact on Billing

For a pay-per-use cluster, you can see its new price when confirming the scale-in on the console. After the scale-in is complete, the cluster will be billed based on the new price. For details, see **Cloud Search Service Price Calculator**.

Constraints

- During a scale-in operation, the data on the to-be-removed nodes needs to be migrated to the remaining nodes. The timeout for data migration per node is 48 hours. Scale-in will fail if this timeout expires.
- When the cluster has large quantities of data, you are advised to manually adjust the data migration rate and avoid performing the migration during peak hours.
- After a scale-in operation is complete, the cluster's disk usage must be lower than 80%.
- Rules for scaling in a cluster that has no master nodes:
  - Condition: Scale-in is allowed only when **the number of data nodes + the number of cold data nodes ≥ 3**.
  - Maximum number of nodes that can be removed during a single scale-in operation: You cannot remove half or more of the cluster's current data nodes and cold data nodes.

    For example, if a cluster has three data nodes, three client nodes, and three cold data nodes, a maximum of two nodes can be removed at a time. Formula: (3+3)/2 = 3; and the number of nodes that can be removed should be less than 3, so the correct result is 2.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

  - Data reliability: After a scale-in operation, the total number of remaining data nodes and cold data nodes must be greater than the maximum number of index replicas in the cluster.

    For example, if the maximum number of replicas for all indexes in the cluster is 2, the total number of data nodes and cold data nodes after the scale-in must be at least 3.

- Rules for scaling in a cluster that has master nodes:
  - Non-master nodes (data nodes, client nodes, or cold data nodes): For each node type, the number of nodes must be at least 2 before you can proceed with a scale-in operation.
  - Master nodes: For a single scale-in operation, you can only remove less than half of the master nodes.

    For example, if a cluster has two data nodes and five master nodes, only two master nodes can be removed for the current scale-in operation. Formula: 5/2 = 2.5; and the number of nodes that can be removed should be less than 2.5, so the correct result is 2.
- Rules for removing nodes from a multi-AZ cluster:
  - Even distribution: After a scale-in operation, the difference in the number of same-type nodes across different AZs must not exceed 1.
  - Minimum number of nodes required to maintain high availability:
    - **For a two-AZ cluster**, there must be at least four data nodes (or cold data nodes), two for each AZ; and there must be at least two client nodes (if configured), one for each AZ.
    - **For a three-AZ cluster**, each available node type must have at least one node in each AZ. This means each node type must have at least three nodes in total.
  - Performance enhancement: To prevent uneven data distribution and ensure optimal query and ingestion performance, the number of data nodes or cold data nodes should be an integer multiple of the number of AZs.
- For the range of node quantities supported by each node type, see **Table 3-207**.

Table 3-207 Node quantity ranges

Node Type	Value Range
Data nodes	• Without master nodes: 1 to 32 • With master nodes: 1 to 200 (If the combined number of data nodes and cold data nodes exceeds 100, you are advised to select a flavor with at least 16 vCPUs. This helps ensure optimal read and write performance.)
Master nodes	3, 5, 7, or 9 (must be an odd number from 3 to 9)
Client nodes	1~64
Cold data nodes	1–32

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Change Impact

Before the change, learn about possible impacts and operation suggestions, and develop a plan to minimize these impacts.

- Performance

  During a scale-in, shards on the to-be-removed nodes are migrated to the remaining nodes. This process will consume I/O performance. This is why you are advised to perform the operation during off-peak hours.

  To minimize this impact, it is advisable to adjust the data migration rate based on the cluster's traffic cycle: increase the data migration rate during off-peak hours to shorten the task duration, and decrease it **before** peak hours arrive to ensure optimal cluster performance. The data migration rate is determined by the **indices.recovery.max_bytes_per_sec** parameter. The default value of this parameter is the number of vCPUs multiplied by 8 MB. For example, for four vCPUs, the data migration rate is 32 MB/s. You can adjust it based on the service requirements.

  ```
  PUT /_cluster/settings
  {
    "transient": {
      "indices.recovery.max_bytes_per_sec": "128MB"
    }
  }
  ```

- Impact on cluster load

  After a scale-in, the remaining nodes will need to handle all of the cluster's load. This may lead to higher CPU, memory, and disk I/O usage, impacting query and write performance. If shards are unevenly allocated, performance bottlenecks may occur. This is why before a scale-in, it is necessary to evaluate whether the remaining nodes have the capacity to handle the current cluster load.

- Characteristics of this process

  Once started, a scaling task cannot be stopped until it succeeds or fails.

Scale-in Duration

The following formula can be used to estimate how long a scale-in operation will take:

Scale-in duration (min) = 5 (min) x Number of nodes to be removed + Data migration duration (min)

where, 5 minutes indicates how long non-data migration operations (e.g., initialization) typically take per node. It is an empirical value.

Data migration duration (min) = Total data size of the nodes to be removed (MB) ÷ [Total number of vCPUs of the data nodes x 8 (MB/s) x 60 (s)]

where,

- 8 MB/s indicates that each vCPU can process 8 MB of data per second. It is an empirical value.
- The formulas above use estimates under ideal conditions. The actual migration speed depends on cluster load.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Prerequisites

- The cluster status is **Available**, and there are no ongoing tasks.
- All mission-critical data has been backed up. For details, see **Creating Snapshots to Back Up Data**.

Removing Nodes Randomly

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. Click the **Scale Cluster** tab.
5. Click **Scale in** to set parameters.

   Table 3-208 Removing nodes randomly

Parameter	Description
Action	Select **Scale in**.
Resources	Quantities of resources reduced.
Nodes	Reduce the number of nodes in the **Nodes** column. You can change multiple node types at the same time. For the range of node quantities supported by each node type, see **Constraints**.

6. Click **Next**.
7. Confirm the information and click **Submit**.
8. Click **Back to Cluster List** to go back to the **Clusters** page. **Task Status** is **Scaling in**. When **Cluster Status** changes to **Available**, the cluster has been successfully scaled in.

Removing Specified Nodes

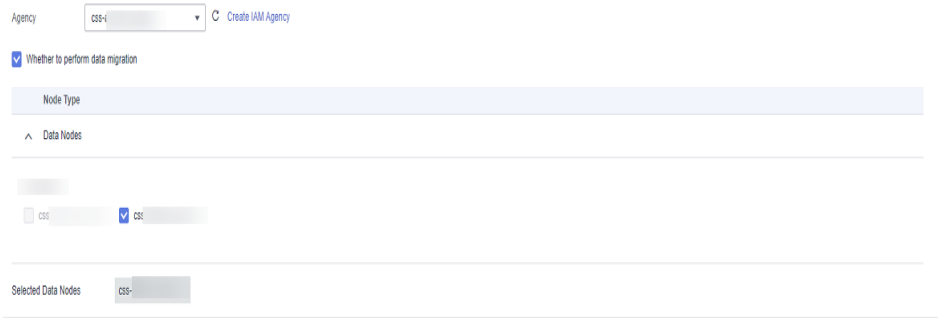
1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. On the **Modify Configuration** page, click the **Scale In Nodes** tab.
5. Set scale-in parameters.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-209 Removing specified nodes (Scale In Nodes)

Parameter	Description
Node Type	Expand the node type that needs be changed to show all nodes under it. Select the nodes you want to remove.

Figure 3-80 Removing specified nodes



6. Click **Next**.
7. Confirm the change information and click **Submit**. In the confirm dialog box, choose to migrate data, which helps to prevent data loss, and click **OK**.

   During data migration, the system migrates all data from the to-be-removed nodes to the remaining nodes, and removes these nodes upon completion of the data migration. If the data on the to-be-removed nodes has replicas on other nodes, data migration can be skipped and the cluster change can be completed faster.
8. Click **Back to Cluster List** to go back to the **Clusters** page. **Task Status** is **Scaling in**. When **Cluster Status** changes to **Available**, the cluster has been successfully scaled in.

Related Documents

- For an OpenSearch cluster, you can also optimize costs by changing node specifications and EVS disk types. For details, see **Changing Node Specifications**.
- If you want to reduce cluster nodes even though your cluster is not eligible for a scale-in operation, you can simply create a new cluster. Then you can migrate data to the new cluster using snapshots, and delete the old cluster after data is migrated.

3.16.3 Changing Node Specifications

If the workloads on an OpenSearch cluster's data plane change, you can scale the cluster vertically by changing its node specifications or node storage type.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-210 Change scenarios

Change Type	Scenario	Change Process
Changing node specifications	Typically, you increase node specifications instead of decreasing them. Common scenarios include: • If the allocation of new indexes or shards takes too long or the coordination and scheduling of nodes are inefficient, increase the master node specifications. • If too many requests need to be handled or too many results need to be aggregated, increase the client node specifications. • If data nodes are becoming slower in responding to data writing and query requests, increase the specifications of the data nodes. • If cold data query becomes slow, increase the cold data node specifications. • If the CPU or JVM metrics of cluster nodes show performance bottlenecks, increase node specifications to improve cluster performance. Alternatively, you may also decrease node specifications, but doing so will decrease the cluster's data processing and storage capacities. Exercise caution.	1. Bring a node offline. 2. Change the node specifications. 3. Restart the node and restore its data. 4. After the node recovers, proceed to another node and repeat the steps above. This goes on until all nodes are changed. The node specifications are changed one node at a time. This is to ensure that there are sufficient resources to keep services running.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Change Type	Scenario	Change Process
Changing the node storage type (disk type)	Change the node storage type if disk I/O has become a performance bottleneck, which impacts query and write performance.	1. Select a random node and migrate its data to other nodes. 2. Rebuild the node using the new storage type. 3. Add the node back into the cluster. The system automatically triggers a shard reallocation, migrating some of the shards to the new node. 4. After the node recovers, proceed to another node and repeat the steps above. The nodes are changed one at a time to prevent service interruptions.

Impact on Billing

For a pay-per-use cluster, you can see its new price when confirming the node specifications or storage type change on the console. After the change is complete, the cluster will be billed based on the new price. For details, see **Cloud Search Service Price Calculator**.

Constraints

- The node specifications and storage type cannot be changed for nodes that use local disks.
- The node specifications and storage type cannot be changed at the same time.
- The node storage type can be changed only for data nodes and cold data nodes.
- When you change the node storage type, data needs to be migrated between different nodes. The timeout for data migration per node is 48 hours. Upgrade will fail if this timeout expires. When the cluster has large quantities of data, you are advised to manually adjust the data migration rate and avoid performing the migration during peak hours.
- For a cluster without master nodes, the node specifications and node storage type can be changed only if the number of data nodes plus cold data nodes is at least three.
- For a cluster with master nodes, this operation is allowed only if the cluster has at least two data nodes.
- During a node storage type change, there is always one node that is unavailable. To ensure service continuity, make sure the **total number of**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

data nodes and cold data nodes is greater than the maximum number of index replicas plus 1. For a single-AZ or dual-AZ cluster, also make sure there are at least two nodes of each type for the cluster in each of its AZs.

- During a node specifications change, nodes are brought offline in order to make the changes. To ensure service continuity, make sure all shards have replicas.
- Make sure the disk usage is always less than 80% during the change.

Change Impact

Before changing a cluster's node specifications or storage type, it is essential to assess the potential impacts and review operational recommendations. This enables proper scheduling of the change, minimizing service interruptions.

- Performance impact (only when the node storage type is changed)

  Changing the node storage type does not interrupt services. However, data migration that occurs during this process consumes I/O performance, and taking individual nodes offline still has some impact on the overall cluster performance.

  To minimize this impact, it is advisable to adjust the data migration rate based on the cluster's traffic cycle: increase the data migration rate during off-peak hours to shorten the task duration, and decrease it **before** peak hours arrive to ensure optimal cluster performance. The data migration rate is determined by the **indices.recovery.max_bytes_per_sec** parameter. The default value of this parameter is the number of vCPUs multiplied by 8 MB. For example, for four vCPUs, the data migration rate is 32 MB/s. You can adjust it based on the service requirements.

  ```
  PUT /_cluster/settings
  {
    "transient": {
      "indices.recovery.max_bytes_per_sec": "128MB"
    }
  }
  ```

- Impact on request handling

  Taking nodes offline one at a time usually does not interrupt services. However, requests sent to offline nodes may fail. To mitigate this impact, the following measures may be taken:

  - Use a VPC endpoint or a dedicated load balancer to handle access requests to your cluster, which makes sure that requests are automatically routed to available nodes.
  - Enable an exponential backoff & retry mechanism on the client (configure three retries).
  - Perform this operation during off-peak hours.

- Impact on index replicas

  Shards that have no replicas will become unavailable when the nodes that store them are taken offline, causing service interruptions. You are advised to add replicas for all important indexes before making the change described in this topic.

- Impact on OpenSearch Dashboards and Cerebro

  Changing the node storage type for a cluster will cause OpenSearch Dashboards and Cerebro to be rebuilt. During this period, they are temporarily

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

unavailable. During a node specifications change, if OpenSearch Dashboards and Cerebro become unavailable because the node that runs them is taken offline, refresh the web page or try to log in again, and the system will reschedule them to an available node.

- Characteristics of this process

  Once started, a change task cannot be stopped until it succeeds or fails. A change task failure only impacts a single node, and does not interrupt services if there are data replicas, but the failed node still needs to be restored promptly.

If a change of the node specifications or storage type is urgently needed, submit a **service ticket** to contact technical support and perform a pre-change evaluation and the necessary checks.

Change Durations

- The following formulas can be used to estimate how long the node specifications change will take:

  Change duration (min) = 10 (min) x Total number of nodes to change + Data recovery duration (min)

  where,

  – 10 minutes indicates how long non-data recovery operations (e.g., initialization) typically take per node. It is an empirical value.

  – The total number of nodes is the sum of the number of data nodes, master nodes, client nodes, and cold data nodes in the cluster.

  Data recovery duration (min) = Total data size (MB)/[Total number of vCPUs of the data nodes x 8 (MB/s) x 60 (s)]

  where,

  – 8 MB/s indicates that each vCPU can process 8 MB of data per second. It is an empirical value.

  – The formulas above use estimates under ideal conditions. The actual data recovery speed depends on cluster load.

- The following formulas can be used to estimate how long the node storage type change will take:

  Change duration (min) = 15 (min) x Total number of nodes to change + Data migration duration (min)

  where,

  – 15 minutes indicates how long non-data migration operations (e.g., initialization) typically take per node. It is an empirical value.

  – The total number of nodes is the sum of the number of data nodes, master nodes, client nodes, and cold data nodes in the cluster.

  Data migration duration (min) = Total data size (MB)/[Total number of vCPUs of the data nodes x 8 (MB/s) x 60 (s)]

  where,

  – 8 MB/s indicates that each vCPU can process 8 MB of data per second. It is an empirical value.

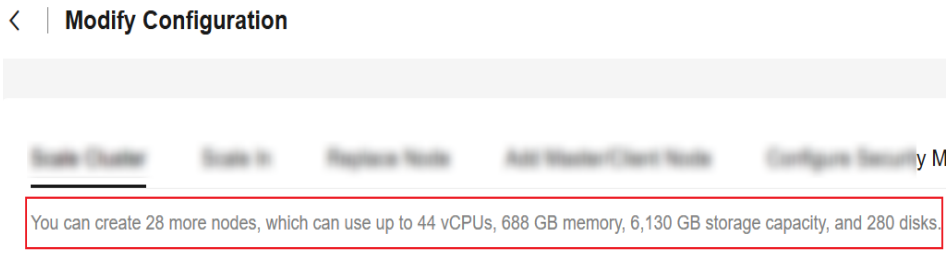
  – The formulas above use estimates under ideal conditions. The actual migration speed depends on cluster load.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Prerequisites

- The cluster status is **Available**, and there are no ongoing tasks.
- Your CSS resource quotas are sufficient for the capacity expansion you are about to perform. You can check available resources on the **Modify Configuration** page.

  Figure 3-81 Checking available resources

- All mission-critical data has been backed up before a node storage type change. This is to prevent data loss. For details, see **Creating Snapshots to Back Up Data**.

Changing the Node Specifications and Storage Type

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters** > **OpenSearch**.
3. Check that all service data has replicas so that services will not be interrupted during the specifications change.
   a. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
   b. In the left navigation pane, choose **Dev Tools**.
   c. Run the **GET _cat/indices?v** command in OpenSearch Dashboards.
      - If the returned **rep** value is greater than **0**, data replicas exist. Go to the next step.
      - If the returned **rep** value is 0, there are no data replicas. You are advised to manually create a snapshot for the cluster before moving on to the next step. For details, see **Creating Snapshots to Back Up Data**.
4. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
5. Click the **Scale Cluster** tab.
6. Click **Change specifications** to set parameters.

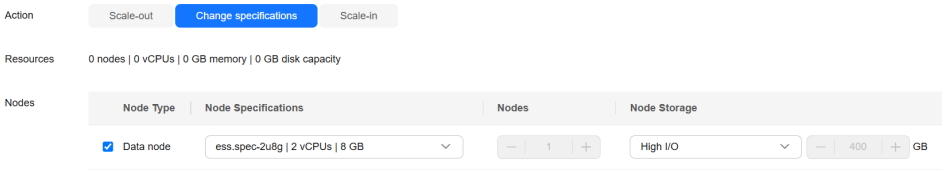
   Table 3-211 Specifications change

Parameter	Description
Action	Select **Change specifications**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Resources	Shows the change of resources for this operation.
Nodes	Configure the changes you want to make. 1. Select a node type in the **Node Type** column. 2. Select a new flavor in the **Node Specifications** column, or select a new storage type in the **Node Storage** column. The node specifications and storage type cannot be changed at the same time.

Figure 3-82 Changing specifications



7. Click **Next**.
8. Confirm the information and click **Submit**.
9. In the displayed dialog box, confirm check items, and click **OK** to start the specifications change.
   - Check items for node specifications change: **Verify index copies & nodes** and **Cluster status check**.
   - Check items for node storage type change: **Check cluster load**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-212 Check item description

Item	Description
Verify index replicas & nodes	• Verifying the existence of index replicas helps improve data security. You can skip this step, but the absence of index replicas may impact data reliability during a node specifications change. • Verifying data nodes ensures the cluster has enough capacity to handle requests. Skipping this check risks service interruptions when changing node specifications. Proceed with caution. Verification rules: • If you select **Verify index copies & nodes** and the cluster has no master nodes, each index is required to have at least one replica and the cluster must have at least three data nodes plus cold data nodes. • If you select **Verify index copies & nodes** and the cluster has a master node, each index is required to have at least one replica, but there is no requirement on how many nodes the cluster must have.
Cluster status check	During a node specifications change, the cluster status is checked by default to improve the success rate and ensure data security. The nodes are changed one at a time. For each node, the system changes its specifications, restarts it, and checks that all its processes are started successfully before moving on to the next node. In emergencies (for example, when a cluster is overloaded and services are faulty, which may prevent a specifications change request from being delivered), you can skip cluster status check so that more resources can be made available for cluster recovery. However, doing so may cause the cluster to become faulty and interrupt services. Exercise caution.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Item	Description
Check cluster load	During a node storage type change, data migration between nodes and the stopping and restarting of nodes will consume cluster resources, causing the cluster load to increase. A cluster load check can identify possible overload risks for a cluster and reduce the likelihood of an overload condition causing the node storage type change to fail. The cluster load check items are as follows: • **nodes.thread_pool.search.queue < 1000**: Check that the maximum number of requests in the search queue is less than 1000. • **nodes.thread_pool.write.queue < 200**: Check that the maximum number of requests in the write queue is less than 200. • **nodes.process.cpu.percent < 90**: Check that the maximum CPU usage is less than 90%. • **nodes.os.cpu.load_average/Number of vCPUs < 80%**: Check that the number of running processes plus the number of processes waiting for CPUs is less than 80% of the total number of vCPUs.

NOTE

If the change request fails to be submitted and a message is displayed indicating that the cluster needs to be upgraded, it means the current cluster version does not support a node storage type change. Upgrade the cluster to the latest image version and then try again. For a detailed upgrade guide, see **Upgrading the Cluster Version**.

10. Click **Back to Cluster List** to go back to the **Clusters** page. The **Cluster Status** is **Configuration modified**. When **Cluster Status** changes to **Available**, the cluster node specifications have been successfully changed.

3.16.4 Replacing a Faulty Node

If a node in an OpenSearch cluster is faulty, you can replace it to restore services.

The node replacement process is as follows:

1. Migrate data from the node that needs to be replaced to other available nodes.
2. Rebuild a new node using this node's current ID, IP address, specifications, and AZ.
3. Add the new node into the cluster. The system automatically triggers a shard reallocation, migrating some of the shards to the new node.

This process does not interrupt services because data is migrated from the replaced node to other available nodes.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Constraints

- Only one node can be replaced at a time. Each new node is rebuilt using the ID, IP address, specifications, and AZ of the node it is replacing.
- The configurations you modified manually will not be retained after node replacement. For example, if you have manually added a return route for the original node, you need to add it again for the new node after the node replacement is complete.
- If the node you want to replace is a data node or cold data node, pay attention to the following precautions:
  - When a data node or cold data node is replaced, its data is first migrated to other data nodes. This means the **total number of data nodes and cold data nodes must be greater than the maximum number of index replicas plus 1**.
  - In the AZ that contains the data node or cold data node to be replaced, there has to be at least another data node or cold data node.
  - If the cluster has no master nodes, the total number of data nodes plus cold data nodes must be at least three.
  - The precautions above do not apply if you are replacing a faulty node, regardless of its type. This is because faulty nodes are not included in **_cat/nodes**.

Change Impact

Before replacing a node, it is essential to assess the potential impacts and review operational recommendations. This enables proper scheduling of the node replacement, minimizing service interruptions.

- Impact on performance

  Replacing a node does not interrupt services. However, data migration that occurs during this process consumes I/O performance, and taking individual nodes offline still has some impact on the overall cluster performance.

  To minimize this impact, it is advisable to adjust the data migration rate based on the cluster's traffic cycle: increase the data migration rate during off-peak hours to shorten the task duration, and decrease it **before** peak hours arrive to ensure optimal cluster performance. The data migration rate is determined by the **indices.recovery.max_bytes_per_sec** parameter. The default value of this parameter is the number of vCPUs multiplied by 8 MB. For example, for four vCPUs, the data migration rate is 32 MB/s. You can adjust it based on the service requirements.

  ```
  PUT /_cluster/settings
  {
    "transient": {
      "indices.recovery.max_bytes_per_sec": "128MB"
    }
  }
  ```

- Impact on request handling

  While a node is being replaced, requests sent to it may fail. To mitigate this impact, the following measures may be taken:

  - Use a VPC endpoint or a dedicated load balancer to handle access requests to your cluster, which makes sure that requests are automatically routed to available nodes.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

  - Enable an exponential backoff & retry mechanism on the client (configure three retries).
  - Perform node replacement during off-peak hours.
- Characteristics of this process

  Once started, a node replacement task cannot be stopped until it succeeds or fails. A task failure only impacts a single node, and does not interrupt services if there are data replicas, but the failed node still needs to be restored promptly.

Node Replacement Duration

The following formula can be used to estimate how long it will take to replace a specified node of a cluster:

Change duration (min) = 15 (min) + Data migration duration (min)

where, 15 minutes indicates how long non-data migration operations (e.g., initialization) typically take per node. It is an empirical value.

Data migration duration (min) = Total data size (MB)/[Total number of vCPUs of the data nodes x 8 (MB/s) x 60 (s)]

where,

- 8 MB/s indicates that each vCPU can process 8 MB of data per second. It is an empirical value.
- The formulas above use estimates under ideal conditions. The actual migration speed depends on cluster load.

Prerequisites

- The cluster status is **Available**, and there are no ongoing tasks.
- All mission-critical data has been backed up. For details, see **Creating Snapshots to Back Up Data**.

Replacing a Specified Node

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. On the **Modify Configuration** page, click the **Replace Node** tab.
5. On the **Replace Node** tab, set the parameters as needed.

Table 3-213 Replacing a specified node

Parameter	Description
Node Type	Select the node you want to replace. You can expand a node type to check all the nodes under it.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-83 Replacing a specified node



6. Click **Submit**. In the data migration confirmation dialog box, choose to migrate data, and click **OK**.

   During data migration, the system migrates all data from the to-be-replaced node to the remaining nodes, and replace the node upon completion of the data migration. If the data on the to-be-replaced nodes has replicas on other nodes, data migration can be skipped and the cluster change can be completed faster.

7. Click **Back to Cluster List** to go back to the **Clusters** page. The **Task Status** is **Upgrading**. When **Cluster Status** changes to **Available**, the node has been successfully replaced.

3.16.5 Upgrading the Cluster Version

In CSS, you can upgrade the version of an OpenSearch cluster to use feature and performance enhancements or fix known issues.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-214 Upgrade scenarios

Upgrade Type	Scenario	Upgrade Process
Same-version upgrade	Update the cluster's kernel patches. The cluster is upgraded to the latest image of the current version to fix known issues or optimize performance. For example, if the cluster version is **1.3.6(1.3.6_24.3.3_0102)**, upon a same-version upgrade, the cluster will be upgraded to the latest image **1.3.6(1.3.6_24.3.4_0109)** of version 1.3.6. (The version numbers used here are examples only.)	1. Bring a node offline, and migrate its data to other nodes. 2. Upgrade the node to the target version. 3. Add the node back into the cluster. The system automatically triggers a shard reallocation, migrating some of the shards to the new node. 4. Confirm that the node status is normal. Then upgrade the remaining nodes one by one. Nodes in clusters are upgraded one by one to avoid interrupting services.
Cross-version upgrade	Upgrade a cluster to the latest image of the target version to enhance functionality or normalize software versions. For example, if the cluster version is **1.3.6(1.3.6_24.3.3_1224)**, upon a cross-version upgrade, the cluster will be upgraded to the latest image **2.19.0(2.19.0_25.9.0_1107)** of version 2.19.0. (The version numbers used here are examples only.)	

Version Support

The supported target versions vary depending on the current version. For details, see **Table 3-215**.

Table 3-215 Version upgrade paths

Current Version	Target Version
OpenSearch 1.3.6	OpenSearch 2.19.0
OpenSearch 2.11.0	OpenSearch 2.19.0
OpenSearch 2.17.1	OpenSearch 2.19.0

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Constraints

- A maximum of 20 clusters can be upgraded at the same time.
- During an upgrade, the data on the upgraded nodes needs to be migrated to other nodes. The timeout for data migration per node is 48 hours. Upgrade will fail if this timeout expires. When the cluster has large quantities of data, you are advised to manually adjust the data migration rate and avoid performing the migration during peak hours.

Upgrade Impact

Before upgrading a cluster, it is essential to assess the potential impacts and review operational recommendations. This enables proper scheduling of the upgrade time, ensuring a smooth upgrade process and minimizing service interruptions.

- Impact on performance

  The nodes of a cluster are upgraded one at a time to ensure service continuity. However, data migration that occurs during the upgrade consumes I/O performance, and taking individual nodes offline still has some impact on the overall cluster performance.

  To minimize this impact, it is advisable to adjust the data migration rate based on the cluster's traffic cycle: increase the data migration rate during off-peak hours to shorten the task duration, and decrease it **before** peak hours arrive to ensure optimal cluster performance. The data migration rate is determined by the **indices.recovery.max_bytes_per_sec** parameter. The default value of this parameter is the number of vCPUs multiplied by 8 MB. For example, for four vCPUs, the data migration rate is 32 MB/s. You can adjust it based on the service requirements.

  ```
  PUT /_cluster/settings
  {
    "transient": {
      "indices.recovery.max_bytes_per_sec": "128MB"
    }
  }
  ```

- Impact on request handling

  During the upgrade, nodes are upgraded one at a time. Requests sent to the node that is being upgraded may fail. To mitigate this impact, the following measures may be taken:

  - Use a VPC endpoint or a dedicated load balancer to handle access requests to your cluster, which makes sure that requests are automatically routed to available nodes.
  - Enable an exponential backoff & retry mechanism on the client (configure three retries).
  - Perform the upgrade during off-peak hours.

- Rebuilding OpenSearch Dashboards and Cerebro

  OpenSearch Dashboards and Cerebro will be rebuilt during the upgrade, making them temporarily unavailable. Additionally, due to cross-version compatibility issues, OpenSearch Dashboards may become unavailable during the upgrade. These problems will go away once the upgrade is completed.

- Characteristics of the upgrade process

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Once started, an upgrade task cannot be stopped until it succeeds or fails. An upgrade failure only impacts a single node, and does not interrupt services if there are data replicas.

Upgrade Duration

The following formula can be used to estimate how long it takes to upgrade a cluster:

Upgrade duration (min) = 15 (min) x Total number of nodes to be upgraded + Data migration duration (min)

where,

- 15 minutes is how long non-data migration operations (e.g., initialization and image upgrade) typically take per node. It is an empirical value.
- The total number of nodes is the sum of the number of data nodes, master nodes, client nodes, and cold data nodes in the cluster.

Data migration duration (min) = Total data size (MB)/[Total number of vCPUs of the data nodes x 8 (MB/s) x 60 (s) x Concurrency]

where,

- Typically, when concurrency is higher than 1, the actual data migration rate can only reach half of the throughput intended by the configured concurrency, depending on the cluster load.
- 8 MB/s indicates that each vCPU can process 8 MB of data per second. It is an empirical value.

The formulas above use estimates under ideal conditions. In practice, it is advisable to add 20% to 30% of redundancy.

Pre-Upgrade Check

To ensure a successful upgrade, you must check the items listed in the following table before performing an upgrade.

Table 3-216 Pre-upgrade checklist

Check Item	Check Method	Description	Normal Status
Cluster status	System check	After an upgrade task is started, the system automatically checks the cluster status. Clusters whose status is **green** or **yellow** can work properly and have no unallocated primary shards.	The cluster is available and there are no ongoing tasks.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Check Item	Check Method	Description	Normal Status
Node quantity	System check	After an upgrade task is started, the system automatically checks node quantities. To ensure service continuity, there has to be at least two nodes of each type for a cluster in each of its AZs. For a cluster with master nodes, there has to be at least two data nodes. For a cluster without master nodes, the number of data nodes plus cold data nodes must be three at least.	• Number of nodes of each type in each AZ ≥ 2 • For a cluster with master nodes: Number of data nodes ≥ 2 • For a cluster without master nodes: Number of data nodes + Number of cold data nodes ≥ 3
Disk capacity	System check	After an upgrade task is started, the system automatically checks the disk capacity. During the upgrade, nodes are brought offline one by one and then new nodes are created. Ensure that the total disk capacity of the remaining nodes is sufficient for handling all the data of the cluster, and that the nodes' disk usage stays below 80%.	After a single node is brought offline, the total disk capacity of the remaining nodes is sufficient for handling all the data of the cluster, and the nodes' disk usage stays below 80%.
Data replicas	System check	Check that the remaining data nodes and cold data nodes can handle the maximum number of primary and standby shards of indexes in the cluster. There should not be unallocated replicas during the upgrade.	Number of data nodes + Number of cold data nodes > Maximum number of index replicas + 1
Data backup	System check	Before the upgrade, back up data to prevent data loss caused by upgrade failures. When submitting an upgrade task, you can choose whether to check for full index snapshots.	Check whether data has been backed up.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Check Item	Check Method	Description	Normal Status
Resources	System check	After an upgrade task is started, the system automatically checks resources. Resources will be created during the upgrade. Ensure that resources are available.	Resources are available and sufficient.
Custom plugins	System and manual check	Perform this check only if custom plugins are installed in the source cluster. If a cluster uses custom plugins, upload all plugin packages for the target version on the plugin management page before the upgrade. During the upgrade, the custom plugins will be installed on the new nodes. Otherwise, the custom plugins will be lost after a successful cluster upgrade. After an upgrade task starts, the system automatically checks whether the required custom plugin packages have been uploaded. However, you must verify that the uploaded plugin packages are correct and compatible with the target version. If a plugin package is invalid or incompatible, the upgrade will fail. **NOTE** After the upgrade is complete, the status of the custom plugin is reset to **Uploaded**.	The plugin package of the cluster to be upgraded has been uploaded to the plugin list.
Custom configurations	System check	During the upgrade, the system automatically synchronizes the content of the cluster configuration file **opensearch.yml**.	Clusters' custom configurations are not lost after the upgrade.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Check Item	Check Method	Description	Normal Status
Non-standard operations	Manual check	Check whether non-standard operations have been performed in the cluster. Non-standard operations refer to manual operations that are not recorded. These operations cannot be automatically passed on during the upgrade, for example, modification of the **opensearch_dashboards.yml** configuration file, system settings, and return routes.	Some non-standard operations are compatible. For example, the modification of a security plugin can be retained through metadata, and the modification of system configuration can be retained using images. Some non-standard operations, such as the modification of the **opensearch_dashboards.yml** file, cannot be retained, and you must back up the file in advance.
Compatibility check	System and manual check	After a cross-version upgrade task is started, the system automatically checks whether the source and target versions have incompatible configurations. If a custom plugin is installed for a cluster, the version compatibility of the custom plugin needs to be manually checked.	Configurations before and after the cross-version upgrade are compatible.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Check Item	Check Method	Description	Normal Status
Check Cluster Loads	System and manual check	If the cluster is heavily loaded, there is a high probability that the upgrade will get stuck or fail. You are advised to check the cluster load before the upgrade and perform the upgrade only during off-peak hours. You can also choose to check the cluster load while configuring upgrade information.	• **nodes.thread_pool.search.queue < 1000**: Check that the maximum number of requests in the search queue is less than 1000. • **nodes.thread_pool.write.queue < 200**: Check that the maximum number of requests in the write queue is less than 200. • **nodes.process.cpu.percent < 90**: Check that the maximum CPU usage is less than 90%. • **nodes.os.cpu.load_average/Number of vCPUs < 80%**: Check that the number of running processes plus the number of processes waiting for CPUs is less than 80% of the total number of vCPUs.

Performing the Upgrade Task

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Cluster Snapshots** tab, and perform a full data backup. For details, see **Manually Creating a Snapshot**.

   When creating an upgrade task, you can choose to check whether the full index data has been backed up using snapshots. This helps to prevent data loss in case of an upgrade failure.
5. Click the **Version Upgrade** tab, and set upgrade parameters.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-217 Upgrade parameters

Parameter	Description
Upgrade Type	Select an upgrade type. • Same-version upgrade: upgrade kernel patches to the latest images within the current cluster version. • Cross-version upgrade: upgrade a cluster to the latest image of the target version.
Target Image	Image of the target version. When you select an image, the image name and target version details are displayed. The supported target versions are displayed in the drop-down list of **Target Image**. If no target image is available, possible causes are as follows: • The current cluster is of the latest version. • The current cluster was created before 2023 and has vector indexes. • The new version images are unavailable in the current region. • The current cluster does not support the upgrade type you have selected.

6. Click **Submit**.
7. In the displayed dialog box, configure upgrade check and acceleration options.
   a. Select whether to enable **Check Full Index Snapshot**.

      When enabled, CSS checks for valid snapshots by matching indexes by name. Snapshots help prevent potential data loss caused by upgrade failures.

      CSS cannot check the content or backup times of snapshots. You should manually check existing snapshots. If any of them is over one month old, create the latest snapshot.

   b. Select whether to enable **Check Cluster Loads**.

      During an upgrade, data migration and node restarts will consume cluster resources and increase loads. When this option is enabled, CSS evaluates overload risks for the cluster and reduces the likelihood of a cluster upgrade failure caused by an overload.

      The check items are as follows:

      - **nodes.thread_pool.search.queue < 1000**: Check that the maximum number of requests in the search queue is less than 1000.
      - **nodes.thread_pool.write.queue < 200**: Check that the maximum number of requests in the write queue is less than 200.
      - **nodes.process.cpu.percent < 90**: Check that the maximum CPU usage of cluster nodes is less than 90%.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- **nodes.os.cpu.load_average/Number of vCPUs < 80%**: Check that the number of running processes plus the number of processes waiting for CPUs is less than 80% of the total number of vCPUs.

If any of the results is abnormal, wait until the load drops or actively optimize it before performing the upgrade.

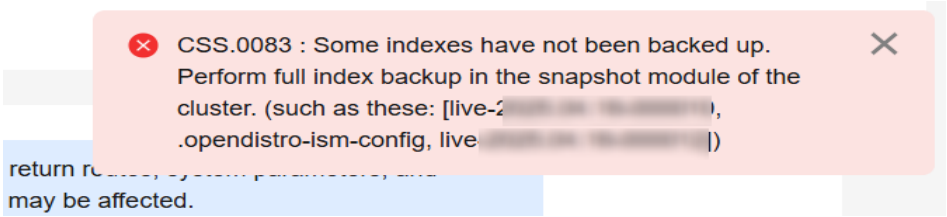
c. Set **Data migration concurrency control**.

Increasing the data migration concurrency can accelerate the upgrade process, but faster data migration leads to higher I/O usage. A higher concurrency is likely to lead to higher cluster load, which may impact cluster performance. You are advised to retain the default value **1**. The value must be less than half the number of data nodes.

8. Click **OK**. The system performs a pre-upgrade check. If the check passes, the upgrade process begins.

   If the pre-upgrade check fails, error information is displayed in the upper right corner of the console. Adjust the cluster configuration accordingly and try again.

   Figure 3-84 Upgrade check errors

   When **Task Status** in the task list below changes to **Successful**, the upgrade is completed.

9. Confirm the upgrade result.

   In the cluster list, find the target cluster, and check the cluster version in the **Version** column to see if the upgrade is successful.

Checking the Update Task

On the upgrade page, you can check the current upgrade task in the **Upgrade Records** area to learn the task progress and status.

Expand the task list and click **View Progress** to check the upgrade progress and node status.

If **Task Status** is **Failed**, you can retry the task or terminate it.

- Retry the task: Click **Retry** in the **Operation** column.
- Terminate the task: Click **Terminate** in the **Operation** column. Before terminating a cross-version upgrade task, ensure that none of the nodes have been upgraded.

  After an upgrade task is terminated, nodes that have failed to be upgraded will remain in the current state, while those that have been successfully upgraded will not be rolled back to the old version. As a result, the nodes in the same cluster may have different versions. You are advised to restart the

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

upgrade as soon as relevant issues are handled, ensuring that all nodes have the same version.

3.16.6 Switching AZs

CSS supports cross-AZ deployment. By switching AZs for a cluster, you can make more resources available to the cluster or improve the cluster's availability.

You can **Add AZ** or **Migrate AZ**.

- **Add AZ**: Add one or two AZs for a single-AZ cluster, or add one AZ for a dual-AZ cluster to enhance cluster availability.
- **Migrate AZ**: Migrate data from the current AZ to another AZ that has sufficient resources for the cluster.

The AZ change process is as follows:

1. Migrate data from one node to other available nodes.
2. Rebuild the node for the target AZ. Keep the node ID, IP address, and specifications unchanged.
3. Add the new node into the cluster. The system automatically triggers a shard reallocation, migrating some of the shards to the new node. Then repeat these steps on all the other nodes.

The nodes are changed one at a time to prevent service interruptions.

Constraints

- To ensure service continuity, the total number of data nodes plus cold data nodes in the cluster must be at least three.
- During the change, nodes are brought offline one by one and then new nodes are created. Ensure that the total disk capacity of the remaining nodes is sufficient for handling all the data of the cluster, and that the nodes' disk usage stays below 80%.
- To make sure that all index shards in the cluster can be allocated to the remaining data nodes and cold data nodes, the **total number of data nodes and cold data nodes must be greater than the maximum number of index replicas plus 1**.
- Before a change completes, some nodes may have already been moved to a new AZ. In this case, the AZs before and after the change are both displayed. After the change succeeds, the new AZs and their nodes will be displayed properly.
- When you **add an AZ** for a cluster, the current AZ must be retained. When adding one or two AZs for a single-AZ cluster, you must change AZs for all nodes. When adding one AZ for a dual-AZ cluster, you can choose to change AZs for specific types of nodes or all nodes in the cluster. For example, for a dual-AZ cluster, you can use three AZs for its master nodes, while still using two AZs for the rest of its nodes.
- When changing AZs, the cluster is rebuilt by minimizing the number of migrated nodes. For a specific node type, the AZ attribute is updated for migrated nodes, while the YML configurations are also updated for unmigrated nodes. Therefore, all nodes of that type must be restarted for the changes to take effect. Exception: When master nodes and data nodes co-

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

exist, modifying only data nodes requires restarting both the master and data nodes. For other node types, you only need to restart nodes of that type.

- When adding AZs for an existing cluster for high availability, follow the node allocation rules below:
  - To ensure high service availability and prevent individual node overload in a multi-AZ deployment, the following node count requirements must be met:
    - **For a two-AZ cluster**, there must be at least four data nodes (or cold data nodes), two for each AZ; and there must be at least two client nodes (if configured), one for each AZ.
    - **For a three-AZ cluster**, each available node type must have at least one node in each AZ. This means each node type must have at least three nodes in total.
    - For node types that are not deployed (for example, no cold data nodes), the minimum node count requirement does not apply.
  - Performance enhancement: To prevent uneven data distribution and ensure optimal query and ingestion performance, the number of data nodes or cold data nodes should be an integer multiple of the number of AZs.
- When migrating an AZ (**Migrate AZ**), you can select only one target AZ. For this operation, you can choose to migrate only specific types of nodes or all nodes in the cluster. For example, for a dual-AZ cluster, you may move the master nodes from one AZ to the other, while keeping the rest of the nodes unchanged. If you migrate from one AZ to another, there is no need to restart the cluster. If multiple AZs are migrated, you must restart the cluster to make the change take effect.

Change Impact

Before changing a cluster's AZs, it is essential to assess the potential impacts and review operational recommendations. This enables proper scheduling of the change, minimizing service interruptions.

- Performance

  Changing AZs for a cluster does not interrupt services. However, data migration that occurs during this process consumes I/O performance, and taking individual nodes offline still has some impact on the overall cluster performance.

  To minimize this impact, it is advisable to adjust the data migration rate based on the cluster's traffic cycle: increase the data migration rate during off-peak hours to shorten the task duration, and decrease it **before** peak hours arrive to ensure optimal cluster performance. The data migration rate is determined by the **indices.recovery.max_bytes_per_sec** parameter. The default value of this parameter is the number of vCPUs multiplied by 8 MB. For example, for four vCPUs, the data migration rate is 32 MB/s. You can adjust it based on the service requirements.

```
PUT /_cluster/settings
{
  "transient": {
    "indices.recovery.max_bytes_per_sec": "128MB"
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  }
}
```

- Impact on request handling

  While a node is offline, requests sent to it may fail. To mitigate this impact, the following measures may be taken:

  - Use a VPC endpoint or a dedicated load balancer to handle access requests to your cluster, which makes sure that requests are automatically routed to available nodes.
  - Enable an exponential backoff & retry mechanism on the client (configure three retries).
  - Perform this operation during off-peak hours.

- Characteristics of this process

  Once started, an AZ change task cannot be stopped until it succeeds or fails.

Change Duration

The following formula can be used to estimate how long it will take to change AZs for a cluster:

Change duration (min) = 15 (min) x Total number of nodes involved + Data migration duration (min)

where, 15 minutes indicates how long non-data migration operations (e.g., initialization) typically take per node. It is an empirical value.

Data migration duration (min) = Total data size of the nodes to be migrated (MB) ÷ [Total number of vCPUs of the data nodes x 8 (MB/s) x 60 (s)]

where,

- 8 MB/s indicates that each vCPU can process 8 MB of data per second. It is an empirical value.
- The formulas above use estimates under ideal conditions. The actual migration speed depends on cluster load.

Prerequisites

- The cluster status is **Available**, and there are no ongoing tasks.
- Make sure that the cluster has not had any non-standard modifications, such as the configuration of custom return routes and system parameters. Such modifications, if ever made, will be lost after the AZ change, which may affect your services.
- All mission-critical data has been backed up. For details, see **Creating Snapshots to Back Up Data**.

Changing AZs

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

4. Click the **Change AZ** tab.
5. On the **Change AZ** page, set parameters.

CAUTION

If the number of data nodes plus cold data nodes in a cluster is not divisible by the number of AZs, data in the cluster may be unevenly distributed after the AZ change is completed. For example, if the cluster has two AZs and three data nodes, one node may need to store twice as much data as the other two nodes. This uneven distribution of data may affect the cluster's query and ingestion performance.

Table 3-218 Parameters for changing AZs

Parameter	Description
Operation Type	• **Add AZ**: Add one or two AZs for a single-AZ cluster, or add one AZ for a dual-AZ cluster. • **Migrate AZ**: Migrate data from one AZ to another.
Node Type	Select a node type for the AZ change. Select one node type or **All nodes** to change their AZ. When adding one or two AZs for a single-AZ cluster, you can only select **All nodes**.
Current AZ	Current AZ of a cluster
Target AZ	AZ to add or migrate to. • **Add AZ**: Select up to three AZs, which must include all your current AZs. • **Migrate AZ**: Select only one target AZ, which cannot be your current AZ.

6. Click **Submit**. In the displayed dialog box, choose whether to check for the presence of a full index snapshot, and click **OK** to start the change.

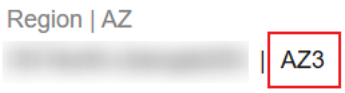
   NOTE

   You are advised to select **Check full index snapshot**. This ensures that all data has been backed up, so that in case the task fails, the data can be restored using this snapshot.

   When **Task Status** in the task list below changes to **Successful**, the AZ change is completed.
7. (Optional) Except for single-AZ migration, all other AZ change scenarios require a cluster restart to take effect.
8. Confirm the result.
   a. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
   b. On the **Overview** tab, check the cluster's AZs in the **Basic Information** area to determine whether the change is successful.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-85 Checking the cluster's AZs



Checking the Progress of an AZ Change Task

In the **Jobs** area on the **Change AZ** tab, find the AZ change task.

Expand the task list and click **View Progress** to check the task progress and node status.

If **Task Status** is **Failed**, you can retry the task or terminate it.

- Retry a task: Click **Retry** in the **Operation** column.
- Terminate a task: Click **Terminate** in the **Operation** column. Before terminating a task, make sure that AZ has been successfully changed for none of the nodes.

3.16.7 Changing the Security Mode

With CSS, when an OpenSearch cluster's security needs change, you can change its security mode settings.

Configure the security mode based on the security needs of your cluster.

CAUTION

HTTPS can only be disabled for OpenSearch 1.3.6, 3.4.0, and 2.19.0 clusters with security mode enabled. For other versions, HTTPS access cannot be disabled.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-219 Cluster security modes

Cluster Type	Key Settings	Description	Applicable Scenario
Non-security mode cluster	**Security Mode**: disabled	Access to such a cluster requires no user authentication, and data will be transmitted in plaintext using HTTP.	Use when creating a cluster for internal testing or workloads that have a low security standard. • Advantage: easy to access the cluster. • Disadvantage: poor security, as anyone can access the cluster. When security mode is disabled, public network access and Dashboards public network access cannot be enabled. To ensure data security, deploy the cluster in a secure environment. Do not expose the cluster's network interface to the public network.
Security-mode cluster	Security-mode cluster + HTTP: • **Security Mode**: enabled • **HTTPS Access**: disabled	Such a cluster requires user authentication. It supports access control and data encryption, and it uses HTTP to transmit data in plaintext.	Use to balance security and performance. • Advantage: User authentication improves cluster security. HTTP-based access ensures high performance of the cluster. User permissions can be configured to ensure proper isolation. • Disadvantage: Public network access is not supported. To ensure data security, deploy the cluster in a secure environment. Do not expose the cluster's network interface to the public network.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Cluster Type	Key Settings	Description	Applicable Scenario
	Security-mode cluster + HTTPS: • **Security Mode**: enabled • **HTTPS Access**: enabled	Such a cluster requires user authentication. It supports access control and data encryption, and it uses HTTPS to encrypt communication and enhance data security.	Use when security takes precedence over performance and public network access is required. • Advantage: User authentication improves cluster security. HTTPS enhances cluster security by encrypting all communication over the public network. User permissions can be configured to ensure proper isolation. • Disadvantage: When HTTPS is used, data encryption and decryption introduce computational overhead and impact the cluster's read and write performance.

Table 3-220 lists the various types of security mode changes supported for CSS clusters.

Table 3-220 Security mode change scenarios

Action	Scenario	Change Process
Switching from the non-security mode to security mode	Non-security mode → Security mode + HTTP: Change a cluster from non-security mode to security mode + HTTP.	1. Select a node and modify the OpenSearch, OpenSearch Dashboards, and Cerebro configuration files. 2. Restart the OpenSearch, OpenSearch Dashboards, and Cerebro processes and restore data. 3. After the node recovers, proceed to another node and repeat the steps above. This goes on until all nodes are changed.
	Non-security mode → Security mode + HTTPS: Change a cluster from non-security mode to security mode + HTTPS.	

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Action	Scenario	Change Process
Switching from security mode to non-security mode	Security mode → Non-security mode: • Change a cluster from security mode + HTTP to non-security mode. • Change a cluster from security mode + HTTPS to non-security mode.	
Switching between HTTP and HTTPS in security mode	HTTP → HTTPS: Change a cluster from security mode + HTTP to security mode + HTTPS.	
	HTTPS → HTTP: Change a cluster from security mode + HTTPS to security mode + HTTP.	

Change Impact

Before changing the security mode for a cluster, it is essential to assess the potential impacts and review operational recommendations. This enables proper scheduling of the change, minimizing service interruptions.

Table 3-221 Impact overview

Action	Service Interruption	Authentication Mode	Performance	Public Network Access	Security Account
Non-security mode → Security mode + HTTP	Yes	Authentication required	N/A	Disallowed	N/A
Non-security mode → Security mode + HTTPS	Yes	Authentication required	Downgraded	Allowed	N/A

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Action	Service Interruption	Authentication Mode	Performance	Public Network Access	Security Account
Security mode → Non-security mode	Yes	Authentication not required	Enhanced	Automatically disabled	Permanently deleted
HTTP → HTTPS	Yes	No change	Downgraded	Allowed	N/A
HTTPS → HTTP	Yes	No change	Enhanced	Automatically disabled	N/A

Impact description:

- The security mode can be enabled or disabled for a cluster, and a security-mode cluster can use either HTTPS or HTTP.
- Service interruption: When a cluster's security mode is changed, the cluster automatically restarts, leading to temporary service unavailability.
- Authentication mode: Changing a cluster's security mode also changes its user authentication mechanism. The client authentication logic must be updated accordingly. If client adaptation cannot be completed in a timely manner, services may be interrupted.
- Performance impact: Given the same hardware configuration, switching from HTTP to HTTPS leads to a performance loss (such as throughput) of around 20% under high concurrency. Conversely, when a cluster switches from HTTPS to HTTP, cluster performance improves.
- Public network access: Only security-mode clusters that use HTTPS support public network access.
- Security account: When the security mode is disabled for a cluster, the system permanently deletes the cluster's security account.
- Tool impact: If the security mode is changed for a cluster that has an ongoing OpenSearch Dashboards session, an error message will be displayed on OpenSearch Dashboards. To rectify the error, clear the cache and launch OpenSearch Dashboards again.

Changing the security mode for a cluster changes its accessibility mode, possibly causing service interruptions. You should perform this operation before services are brought online or when service interruptions can be tolerated.

Change Duration

The following formula can be used to estimate how long it will take to change the security mode for a cluster:

Change duration (min) = 5 (min) x Total number of nodes to change + Data recovery duration (min)

where,

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- 5 minutes indicates how long non-data recovery operations (e.g., initialization) typically take per node. It is an empirical value.
- The total number of nodes is the sum of the number of data nodes, master nodes, client nodes, and cold data nodes in the cluster.

Data recovery duration (min) = Total data size (MB)/[Total number of vCPUs of the data nodes x 8 (MB/s) x 60 (s)]

where,

- 8 MB/s indicates that each vCPU can process 8 MB of data per second. It is an empirical value.
- The formulas above use estimates under ideal conditions. The actual data recovery speed depends on cluster load.

Prerequisites

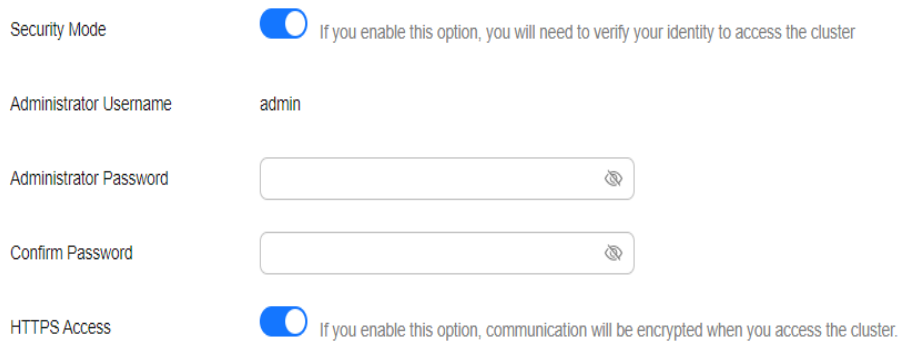
- All mission-critical data has been backed up. For details, see **Creating Snapshots to Back Up Data**.
- The cluster status is **Available**, and there are no ongoing tasks.
- Check whether load balancing is enabled for the cluster. If yes, disable load balancing first. Enable load balancing again after the security mode is changed. This prevents errors in accessing the cluster through the load balancer during the change.

Switching from the Non-Security Mode to Security Mode

You can change a non-security mode cluster to a security-mode cluster that uses HTTP or HTTPS. After a cluster's security mode is enabled, user authentication is required for accessing the cluster.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. Choose the **Change Security Mode** tab.
5. Enable the security mode. Enter and confirm the administrator password of the cluster.

Figure 3-86 Switching from the non-security mode to security mode



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

6. Enable or disable **HTTPS Access**.
   - If you enable **HTTPS Access**, HTTPS is used to encrypt cluster communication, and public network access can be enabled for the cluster.
   - If you disable **HTTPS Access**, HTTP is used, and public network access is disallowed for the cluster.
7. Click **Submit**, and confirm the information. The cluster list is displayed.

   The **Task Status** of the cluster is **Changing security mode**. When the cluster status changes to **Available**, the security mode has been successfully changed.

Switching from Security Mode to Non-Security Mode

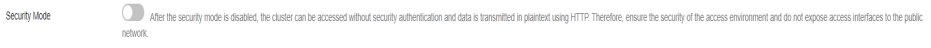
You can change a security-mode cluster that uses HTTP or HTTPS to a non-security cluster. After a cluster's security mode is disabled, user authentication is no longer required for accessing the cluster.

WARNING

- Clusters with the security mode disabled do not require user authentication, and HTTP is used to transmit data in plaintext. Make sure such clusters are deployed in a secure environment, and do not expose their network interface to the public network.
- When a cluster is switched from the security mode to non-security mode, indexes used by the security-mode cluster will be deleted. Back up data before changing the security mode.
- If a public IP address has been assigned to a security-mode cluster, unassign it before changing the security mode.
- If OpenSearch Dashboards public network access is enabled for a security-mode cluster, disable it before changing the security mode.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. Choose the **Change Security Mode** tab.
5. Disable the security mode.

   Figure 3-87 Switching from security mode to non-security mode

6. Click **Submit**. In the displayed dialog box, confirm the information. The cluster list page is displayed.

   The **Task Status** of the cluster is **Changing security mode**. When the cluster status changes to **Available**, the security mode has been successfully changed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Switching between HTTP and HTTPS in Security Mode

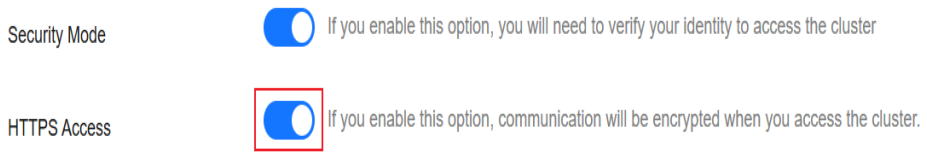
You can change the protocol of a security cluster.

WARNING

- If a public IP address has been assigned to a security-mode cluster, unassign it before changing from HTTPS to HTTP.
- HTTPS can only be disabled for OpenSearch 1.3.6, 2.19.0, and 3.4.0 clusters for which the security mode is enabled. For other versions, HTTPS access cannot be disabled.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. Choose the **Change Security Mode** tab.
5. Enable or disable **HTTPS Access**.

   Figure 3-88 Configuring the protocol

   - If you enable **HTTPS Access**,

     HTTPS is used to encrypt cluster communication, and you can enable public network access for the cluster.
   - If you disable **HTTPS Access**, an alarm message is displayed. Click **OK** to disable the protocol.

     Cluster communication is no longer encrypted and public network access cannot be enabled.
6. Click **Submit**, and confirm the information. The cluster list is displayed.

   The **Task Status** of the cluster is **Changing security mode**. When the cluster status changes to **Available**, the security mode has been successfully changed.

3.17 Backup and Restoration

3.17.1 Creating Snapshots to Back Up Data

CSS allows you to use snapshots to back up and restore the data of an OpenSearch cluster. By storing a snapshot in an OBS bucket, you save a point-in-time copy of the cluster's data. By restoring this snapshot, you can restore the

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

cluster to a previous state. There are two ways to create snapshots: automatic and manual.

- Automatic snapshot creation: Snapshots are automatically created daily or weekly based on a preset time schedule, ensuring ongoing data protection. By configuring an automatic snapshot policy, you reduce manual operations while improving backup reliability and efficiency.
- Manual snapshot creation: You create snapshots manually for special occasions, for example, prior to a high-risk operation (such as a cluster upgrade). This ensures you can restore data using snapshots in case anything should go wrong. Manual snapshots provide additional flexibility.

Impact on Billing

When you use snapshots for data backup, the generated snapshots are stored in OBS buckets, which will result in additional costs. For details, see **Object Storage Service Billing**section "Billing" in *Object Storage Service User Guide*.

Constraints

- Creating cluster snapshots will increase CPU load and disk I/O. To minimize impact on system performance, you are advised to schedule it to happen during off-peak hours.
- When creating or restoring a snapshot, you can only perform certain operations, including accessing OpenSearch Dashboards, viewing metrics, and deleting other snapshots. However, you cannot perform the following operations: modifying configuration, restarting or deleting the cluster, deleting the snapshot that is being created or restored, and creating or restoring another snapshot. While a snapshot is being created or restored for a cluster, any automatic snapshot creation task initiated for that cluster will be canceled.
- When you create the first snapshot in a cluster, a full backup is performed on all data. Subsequent snapshots are all incremental, storing only the changes since the previous snapshot. Because each incremental snapshot relies on its predecessors, you must restore them in the correct sequence.
- In a cluster where storage and compute are decoupled, an index that is both frozen and closed cannot be backed up.
- For an index with read/write splitting between leader and follower enabled, synchronization cannot be stopped while a snapshot is being created for it. To stop synchronization, you must wait until snapshot creation is complete.
- An index that is being backed up via snapshot cannot be deleted or closed. This constraint also applies to deletion or closing operations triggered by ISM policies. To perform such operations, wait until snapshot creation is complete.

Prerequisites

- The OBS bucket used for storing cluster snapshots has been created. The OBS bucket must be in the same region as the target OpenSearch cluster, and the storage class must be **Standard** or **Warm**.
- The cluster status is **Available**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Enabling Cluster Snapshots

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Cluster Snapshots** tab, and check whether cluster snapshots are enabled.
   - If they are enabled, skip this step.
   - If they are disabled, go to the next step.
5. On the **Cluster Snapshots** tab, click **Enable Snapshot**. In the displayed dialog box, configure the necessary settings.

Figure 3-89 Enable Snapshot

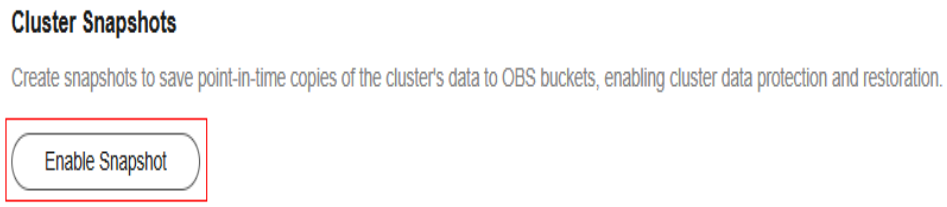


Table 3-222 Enabling cluster snapshots

Parameter	Description

OBS Bucket	From the drop-down list, select an OBS bucket for storing snapshots.
	If no OBS buckets meet your requirements, click **Create Bucket** to go to the OBS console and create one. For details, see **Creating a Bucket**.
	An IAM user must be granted the following OBS permissions to view available OBS buckets:
	`obs:bucket:getBucketStoragePolicy`
	`obs:bucket:listAllMyBuckets`

Backup Path	Snapshot storage path in the OBS bucket. The backup path cannot: • Contain the following characters: **\:*?"<>|'{}** • Start with a slash (/). • Start or end with a period (.). • Contain more than two consecutive slashes (/) or periods (.). • Exceed 512 characters. **CAUTION** Only standard OBS storage can be used to store snapshots. Do not apply OBS lifecycle rules to the backup path.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Maximum Backup Rate (per Second)	The parameter sets the maximum backup rate per node. When it is exceeded, flow control is triggered to prevent excessive resource usage and ensure system stability. **Value format**: number (0–9999) + unit (KB, MB, GB, TB, PB, or B) **Default value**: 40 MB The value 0MB means there is no limit on how fast data is backed up to snapshots. An overly high backup rate may lead to excessive resource usage, which may impact cluster stability. Configure this parameter carefully to maintain optimal performance. The actual backup rate may not reach the configured value, as it depends on factors such as OBS performance and disk I/O.

Maximum Recovery Rate (per Second)	The parameter sets the maximum recovery rate per node. When it is exceeded, flow control is triggered to prevent excessive resource usage and ensure system stability.
	Value format: number (0–9999) + unit (KB, MB, GB, TB, PB, or B)
	The default value is 0MB, indicating no limit.
	An overly high recovery rate may lead to excessive resource usage, which may impact cluster stability. Configure this parameter carefully to maintain optimal performance.
	The actual recovery rate is also affected by the value of **indices.recovery.max_bytes_per_sec**. Whichever is smaller between the two parameters will be used. Additionally, the actual recovery rate may not reach the configured value, as it depends on factors such as OBS performance and disk I/O.
	NOTE
	• To check the value of **indices.recovery.max_bytes_per_sec**, run the following command:
	`GET _cluster/settings`
	• To modify **indices.recovery.max_bytes_per_sec**, run the following command:
	`PUT _cluster/settings`
	`{`
	`  "transient": {`
	`  "indices.recovery.max_bytes_per_sec": "100mb"`
	`  }`
	`}`

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
IAM Agency	Select an IAM agency to grant the current account the permission to access and use OBS. To store snapshots to an OBS bucket, you must have the required OBS access permissions. • If you are configuring an agency for the first time, click **Automatically Create IAM Agency** to create **css-obs-agency**. • If there is an IAM agency automatically created earlier, you can click **One-click authorization** to have the OBS Administrator permissions deleted automatically, and have the following custom policies added automatically instead to implement more refined permissions control. "obs:bucket:GetBucketLocation", "obs:object:GetObjectVersion", "obs:object:GetObject", "obs:object:DeleteObject", "obs:bucket:HeadBucket", "obs:bucket:GetBucketStoragePolicy", "obs:object:DeleteObjectVersion", "obs:bucket:ListBucketVersions", "obs:bucket:ListBucket", "obs:object:PutObject" • When OBS buckets use SSE-KMS encryption, the IAM agency must be granted KMS permissions. You can click **Automatically Create IAM Agency** and **One-click authorization** to have the following custom policies created automatically. "kms:cmk:create", "kms:dek:create", "kms:cmk:get", "kms:dek:decrypt", "kms:cmk:list" • To use **Automatically Create IAM Agency** and **One-click authorization**, the following minimum permissions are required: "iam:agencies:listAgencies", "iam:roles:listRoles", "iam:agencies:getAgency", "iam:agencies:createAgency", "iam:permissions:listRolesForAgency", "iam:permissions:grantRoleToAgency", "iam:permissions:listRolesForAgencyOnProject", "iam:permissions:revokeRoleFromAgency", "iam:roles:createRole" • To use an IAM agency, the following minimum permissions are required: "iam:agencies:listAgencies", "iam:agencies:getAgency", "iam:permissions:listRolesForAgencyOnProject", "iam:permissions:listRolesForAgency" **WARNING** The agency name can contain only letters (case-sensitive), digits, underscores (_), and hyphens (-). Otherwise, the backup will fail.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Determine whether to enable automatic snapshot creation based on service needs.

- To enable it, select **Automatic Snapshot Creation**, and configure necessary settings by referring to **Configuring Automatic Snapshot Creation**.
- To disable it, unselect **Automatic Snapshot Creation**, in which case, you will need to manually create snapshots.

6. Click **OK** to enable cluster snapshots.

   The configuration information will be displayed on the **Cluster Snapshots** tab.

Configuring Automatic Snapshot Creation

You can enable automatic snapshot creation after cluster snapshots are already enabled.

1. On the **Cluster Snapshots** tab, click **Modify Settings**. In the displayed dialog box, select **Automatic Snapshot Creation**, and configure necessary settings.

Figure 3-90 Modifying Settings

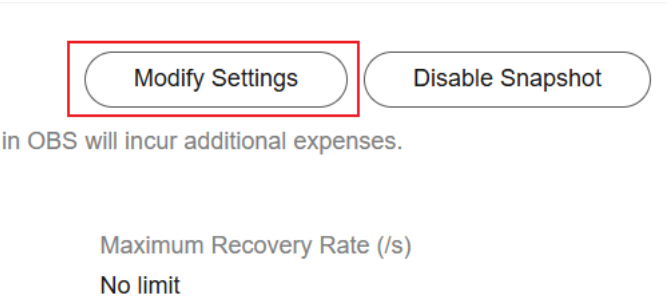


Table 3-223 Configuring automatic snapshot creation

Parameter	Description
Snapshot Name Prefix	A snapshot name consists of the snapshot name prefix and timestamp, for example, *snapshot-1566921603720*. The snapshot name prefix can contain 1 to 32 characters. It must start with a lowercase letter, and it can only contain lowercase letters, digits, hyphens (-), and underscores (_). The default value is **snapshot**.
Time Zone	Select a time zone for the backup start time.
Backup Start Time	Specify the start time of auto backup. Select a value from the drop-down list. The interval can be **Daily**, **Hourly**, or weekly (by selecting a specific day of the week), and the backup time can be set to any hour from 00:00 to 23:00 (full hours only).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Retained Snapshots	Number of automatic snapshots to be retained. The value ranges from 1 to 90. The system automatically deletes excess snapshots every half hour. (The expiration deletion policy applies only to the snapshots that were automatically created at the same frequency as the current automatic snapshot creation policy.) **NOTE** If the snapshot creation interval is short or if the data size of indexes is large, the number of automatic snapshots retained may not reach the value set using this parameter.
Index	Specify the name of the index to be backed up. • You can back up a specified index. To specify multiple indexes, use commas (,) to separate them, for example, **index1,index2,index3**. • You can use an asterisk (*) to match multiple indexes. For example, **index*** indicates that all indexes with the prefix **index** will be backed up. If you do not specify this parameter, all indexes in the cluster are backed up by default. The index name can contain 0 to 1024 characters. It cannot contain spaces, uppercase letters, or special characters **"\<|>/?** **NOTE** You can run the **GET /_cat/indices** command to query the names of all indexes in the cluster.

2. Click **OK** to enable automatic snapshot creation.

   After automatic snapshot creation is enabled, the **Automatic Snapshot Creation** parameter in the **Cluster Snapshots** area is shown as **Enabled**.

   Figure 3-91 Automatic snapshot creation enabled



   Snapshots automatically created according to the snapshot policy as well as those manually created are displayed in the cluster snapshot task list.

   When the snapshot task status changes to **Available**, the snapshot is created. If snapshot creation fails, click on the right of the snapshot list to view the number of failed tasks and learn the failure causes. A maximum of 20 failed

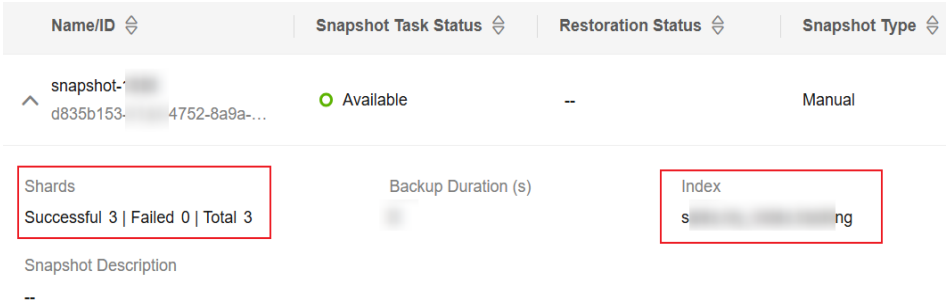
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

tasks can be displayed. When the snapshot function is disabled or the cluster is deleted, the failure records are also cleared.

3. Check whether data is successfully backed up.

   In the cluster snapshot task list, click ⌄ on the left of a snapshot name. The snapshot details are displayed. Check the shards and indexes that have been backed up to see if the backup is successful.

Figure 3-92 View Details



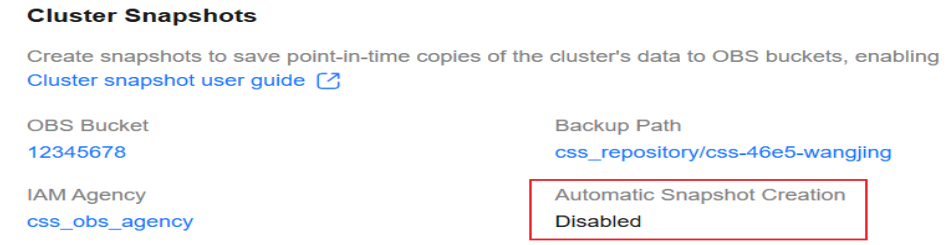
4. To disable automatic snapshot creation, click **Modify Settings**. In the displayed dialog box, deselect **Automatic Snapshot Creation**, and click **OK**. In the displayed dialog box, choose whether to delete snapshots created automatically. Then click **OK** to disable automatic snapshot creation.

CAUTION

- When disabling automatic snapshot creation, if you select **Delete snapshots created automatically**, only existing snapshots will be deleted. Snapshots that are being created will still be created.
- If automatic snapshot creation is disabled without deleting automatically created snapshots, all such snapshots in the cluster cannot be manually deleted later once automatic snapshot creation is enabled again. Instead, they can only be deleted by the system based on the snapshot lifecycle policy. For example, if **Retained Snapshots** is set to **10** in the policy, the system automatically deletes excess snapshots (oldest first) on the next half hour when the number of existing snapshots exceeds this limit.

After automatic snapshot creation is disabled, the **Automatic Snapshot Creation** parameter in the **Cluster Snapshots** area is shown as **Disabled**.

Figure 3-93 Disabling automatic snapshot creation



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Manually Creating a Snapshot

When snapshots are enabled for a cluster, you can manually create snapshots for it.

1. Under **Cluster Snapshot Tasks**, click **Manually Create Snapshot**. In the displayed dialog box, configure the snapshot policy.

   Figure 3-94 Manually creating a snapshot

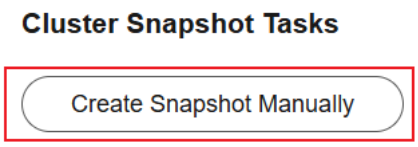
   

   Table 3-224 Parameters for manually creating a snapshot

Parameter	Description
Snapshot Name	Set the snapshot name. Unlike automatically created snapshots, manually created snapshots do not have time information automatically added to their names. The value can contain 4 to 64 characters and must start with a letter. Only lowercase letters, digits, hyphens (-), and underscores (_) are allowed.
Index	Specify the name of the index to be backed up. • You can back up a specified index. When specifying multiple indexes, use commas (,) to separate them, for example, **index1,index2,index3**. • You can use an asterisk (*) to match multiple indexes. For example, **index*** indicates that all indexes with the prefix **index** will be backed up. If you do not specify this parameter, all indexes in the cluster are backed up by default. The index name can contain 0 to 1024 characters. It cannot contain spaces, uppercase letters, or special characters **"\<|>/?** **NOTE** You can run the **GET /_cat/indices** command to query the names of all indexes in the cluster.
Snapshot Description	Add a snapshot description. The value contains 0 to 256 characters and cannot contain special characters **!<>=&\"'**

2. Click **OK** to start creating snapshots.

   Snapshots manually created as well as those automatically created according to the snapshot policy are displayed in the cluster snapshot task list.

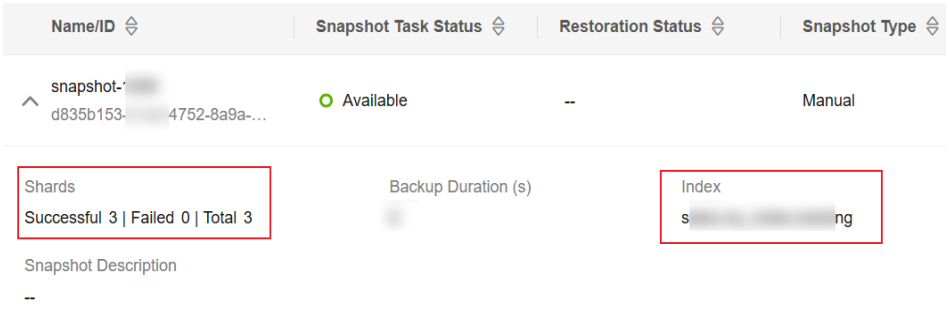
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

When the snapshot task status changes to **Available**, the snapshot is created. If snapshot creation fails, click on the right of the snapshot list to view the number of failed tasks and learn the failure causes. A maximum of 20 failed tasks can be displayed. When the snapshot function is disabled or the cluster is deleted, the failure records are also cleared.

3. Check whether data is successfully backed up.

   In the cluster snapshot task list, click ⌄ on the left of a snapshot name. The snapshot details are displayed. Check the shards and indexes that have been backed up to see if the backup is successful.

Figure 3-95 View Details

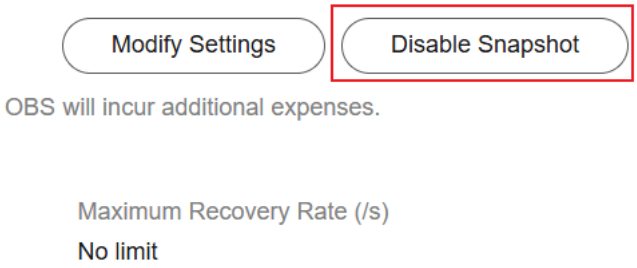


Disabling Cluster Snapshots

If you no longer need the cluster snapshot function, you can disable it to reclaim storage resources.

On the **Cluster Snapshots** page, click **Disable Snapshot**. In the displayed dialog box, click **OK**. Disabling cluster snapshots does not automatically delete existing snapshots. Instead, you need to manually delete them on the OBS console.

Figure 3-96 Disable Snapshot



3.17.2 Restoring Data Using Snapshots

During routine O&M, the ability to rapidly restore data upon data loss or corruption is crucial for OpenSearch clusters. A snapshot allows you to restore a point-in-time copy of a cluster to the original cluster or to another eligible cluster. This process fetches the backed-up index data, metadata, and shard structure from OBS and reconstruct them in the selected destination cluster, ensuring data consistency. This capability is essential for operations like data rollback, cluster migration, and disaster recovery. With snapshots for CSS clusters, you can restore data quickly and reliably to ensure service continuity.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Constraints

- Creating cluster snapshots will increase CPU load and disk I/O. To minimize impact on system performance, you are advised to schedule it to happen during off-peak hours.
- If the source cluster is unavailable, you can restore its snapshots to a designated destination cluster.
- When creating or restoring a snapshot, you can only perform certain operations, including accessing OpenSearch Dashboards, viewing metrics, and deleting other snapshots. However, you cannot perform the following operations: modifying configuration, restarting or deleting the cluster, deleting the snapshot that is being created or restored, and creating or restoring another snapshot. While a snapshot is being created or restored for a cluster, any automatic snapshot creation task initiated for that cluster will be canceled.
- When you create the first snapshot in a cluster, a full backup is performed on all data. Subsequent snapshots are all incremental, storing only the changes since the previous snapshot. Because each incremental snapshot relies on its predecessors, you must restore them in the correct sequence.
- Before snapshot restoration is completed, the indexes being restored to the destination cluster cannot be queried.
- If the two clusters have indexes with the same names but different shard structures, such indexes cannot be restored.
- The version of the destination cluster must be the same as (recommended) or later than that of the source cluster. In an OpenSearch 3.4.0 cluster, you can restore snapshots created in OpenSearch 2.x or 3.4.0. In an OpenSearch 2.19.0 cluster, you can restore snapshots created in Elasticsearch 7.x or OpenSearch 2.19.0. In an OpenSearch 1.3.6 cluster, you can restore snapshots created in Elasticsearch 6.x, Elasticsearch 7.x, or OpenSearch 1.3.6.

Prerequisites

In the source cluster's snapshot task list, there are snapshots whose **Snapshot Task Status** is **Available**. The status of the destination cluster is also **Available**.

Restoring Data

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the source cluster. The cluster information page is displayed.
4. Click the **Cluster Snapshots** tab.
5. In the cluster snapshot task list, select a snapshot, and click **Restore** in the **Operation** column. In the displayed dialog box, configure necessary settings.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-97 Restoring a snapshot

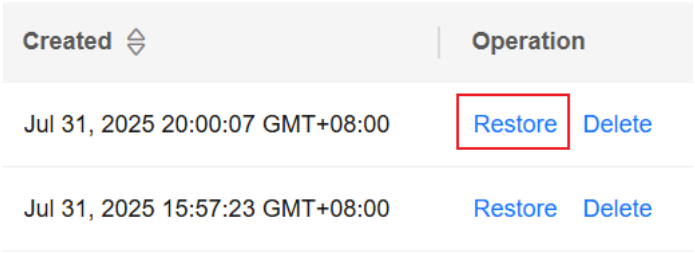


Table 3-225 Restoration settings

Parameter	Description
Index	Specify the name of the index you want to restore. • The value is a string of 0 to 1024 characters that cannot contain uppercase letters, spaces, or the following special characters: **"\<|>/?**. • When restoring an index whose name is prefixed with **.kibana**, the index name must be specified. • The **.opendistro_security** index cannot be restored. • You can use an asterisk (*) to match multiple indexes. For example, **index*** indicates that all indexes with the prefix **index** will be restored. When an asterisk (*) is used for index matching, the **.opendistro_security** index and any system indexes whose name is prefixed with **.kibana** are filtered out by default. • You can restore indexes by specifying their names, for example, **index1,index2,index3**. By default, this parameter is left blank. That is, no index name is specified, and all indexes will be restored.
Rename Pattern	Index name matching rule. Enter a regular expression. Indexes that match the regular expression will be restored. The **Rename Pattern** and **Rename Replacement** take effect only when they are both configured at the same time. For an example, see **Index, Rename Pattern, and Rename Replacement: Parameters and Examples**. The value is a string of 0 to 1024 characters that cannot contain uppercase letters, spaces, or the following special characters: **"\<|>/?,**. For example, **index_(.+)** indicates that all indexes whose name starts with **index_** will be renamed upon restoration.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Rename Replacement	Rule for index renaming. Upon restoration, matching indexes are renamed according to the defined rule. The **Rename Pattern** and **Rename Replacement** take effect only when they are both configured at the same time. For an example, see **Index, Rename Pattern, and Rename Replacement: Parameters and Examples**. The value is a string of 0 to 1024 characters that cannot contain uppercase letters, spaces, or the following special characters: **"\<|>/?,**. For example, **restored_index_$1** indicates that **restored_** will prefix the name of all restored indexes.

Cluster	Select the cluster where you want to restore the data to. You can select the current cluster or another.
	• You must specify a cluster whose status is **Available**. If the status of the current cluster is **Unavailable**, you cannot restore the snapshot to the current cluster.
	• When you choose to restore data to another cluster, make sure the destination cluster has the same or later version than the current cluster, and that the two clusters are in the same region. After a frozen index is restored using a snapshot, the index is stored on the local node and is read-only by default. To make it writable, run the following command to configure the index:
	`PUT index_name/_settings`
	`{`
	`  "index.blocks.write": null`
	`}`

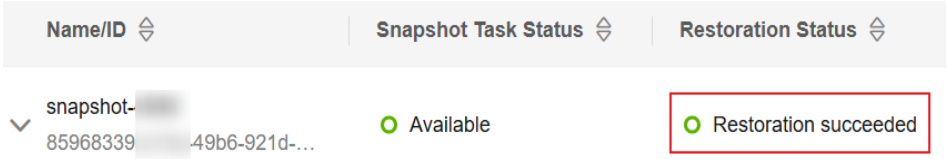
Overwrite same-name indexes in destination cluster	Whether to overwrite same-name indexes in the destination cluster. By default, this option is unselected. Restoring a snapshot will overwrite existing data in the destination cluster. When there are same-name indexes in the destination cluster, you need to select this option in order to restore same-name, same-shard structure indexes. Indexes with a different shard structure cannot be restored.

6. Click **OK** to start restoring the snapshot.

   If restoration succeeds, **Snapshot Task Status** of the snapshot in the snapshot task list will change to **Restoration succeeded**, and the index data captured in the snapshot is reconstructed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-98 Restoration succeeded



Verifying Snapshot Restoration

After snapshot restoration, go to the destination cluster to check the restored index information.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. In the left navigation pane, choose **Dev Tools**.

   The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.
5. Run the following command to check index information.

```
GET _cat/indices
```

Index, Rename Pattern, and Rename Replacement: Parameters and Examples

This section describes the **Index**, **Rename Pattern**, and **Rename Replacement** parameters. It also provides examples to show how to configure them.

- **Retaining the original index name when restoring an index**

  Specify **Index** for the index to be replaced.
- **Renaming an index when restoring it**

  To rename an index upon restoring it, specify **Index**, **Rename Pattern**, and **Rename Replacement**. Their meanings are as follows:

  - **Index**: Specifies the name or pattern (wildcard supported) of the indexes you wish to restore. It must include the indexes that need to be renamed. This parameter defines the indexes that you wish to restore using snapshots to a designated destination cluster.
  - **Rename Pattern**: Uses a regular expression to specify indexes that you wish to rename among the indexes to be restored (specified by **Index**). Only indexes whose names match the regular expression exactly will be renamed.
  - **Rename Replacement**: Defines the naming rule for restored indexes that match **Rename Pattern**. This rule is a character string that uses a capture group (for example, $1) to reuse the text captured by **Rename Pattern**.

Example:

- **Index** is set to **index_*, old_index**: **index_*** indicates all indexes whose name starts with **index_**, and **old_index** specifies a single index.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- **Rename Pattern** is set to **index_(.+)**: This specifies that indexes whose name starts with **index_** will be renamed.
- **Rename Replacement** is set to **restored_index_$1**: This replaces the matched index names with **restored_index_** plus the text of the capture group.

The corresponding API request is as follows:

```
POST /_snapshot/{repo_name}/{snapshot_name}/_restore
{
  "indices": "index_*, old_index",
  "ignore_unavailable": true,
  "include_global_state": true,
  "rename_pattern": "index_(.+)",   // This corresponds to the Rename Pattern parameter.
  "rename_replacement": "restored_index_$1",  // This corresponds to the Rename Replacement parameter.
}
```

Restoration results for different indexes:

- The source index **index_logs** is renamed as **restored_index_logs**.
- The source index **index_users** is renamed as **restored_index_users**.
- The source index **old_index** does not match rename_pattern, and is therefore not renamed. The original name **old_index** is retained after restoration.

3.17.3 Deleting a Cluster Snapshot

While snapshots offer an efficient way to back up OpenSearch clusters, retaining unnecessary snapshots leads to inflated storage costs. To free up space and manage expenses, it is essential to delete them in a timely manner.

Constraints

- Deleting a snapshot will permanently delete its data.
- The deletion of automatically generated snapshots is subject to the following constraints:
  - While automatic snapshot creation is enabled, the system automatically deletes snapshots on the nearest half hour after their retention period expires.
  - After automatic snapshot creation is disabled, existing auto-created snapshots will not be deleted automatically, but they can be deleted manually.
  - If automatic snapshot creation is re-enabled, the system resumes automatic deletion of all auto-created snapshots upon expiration, including these created before automatic snapshot creation was previously disabled.

Manually Deleting a Snapshot

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Cluster Snapshots** tab.
5. In the cluster snapshot task list, select the snapshot you want to delete, and click **Delete** in the **Operation** column. In the displayed dialog box, enter **DELETE** and click **OK**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

A deleted snapshot will not be displayed in the snapshot list.

3.18 Managing Clusters

3.18.1 Viewing Cluster Details

On the CSS management console, you can check an OpenSearch cluster's status, version, nodes, configurations and settings, and resource allocation on both the cluster list and the cluster overview page.

Introduction to the Cluster List

A cluster list displays all clusters of the same type under the current account. If there are a large number of clusters, the cluster list will consist of multiple pages, showing clusters of all states.

Clusters are listed in chronological order by default in the cluster list, with the most recent clusters displayed at the top.

You can click **Export** to download the full cluster list.

Table 3-226 Cluster list parameters

Parameter	Description
Name/ID	Name and ID of a cluster. The cluster ID is automatically generated by the system and uniquely identifies a cluster.
Cluster Status	Current status of a cluster. • Available: The cluster is running properly. • Creating: The cluster is being created. • Processing: The cluster is being restarted, scaled out, backed up, or restored. • Abnormal: The cluster failed to be created or is unavailable. **WARNING** If a cluster is unavailable, you can delete it or use a snapshot created when it was available to restore its data to another cluster. However, you cannot scale out this cluster, access it through OpenSearch Dashboards, create snapshots for it, or use an existing snapshot to restore its data to itself. Do not try to import data into a cluster that is unavailable, as the data may get lost. For such a cluster, you can check its monitoring data or restart it to try to fix the problem. If the fault persists, contact technical support.
Task Status	Status of a task, such as cluster restart, scale-out, or scale-in.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Version	Cluster version and image version. For example, if the version is **1.3.6(1.3.6_x.x.x_x.x.x)**, **1.3.6** is the cluster version and **1.3.6_x.x.x_x.x.x** enclosed in parentheses is the image version. **CAUTION** • For a cluster created a long time ago, its image version may not be available because it is likely not recorded in the database. • If **EOS** is displayed below the version number, the current version has reached EOS and hence poses certain service availability risks. You are advised to upgrade the cluster version as soon as possible. For details, see **Upgrading the Cluster Version**.
Private Network Address	Private network address and port number of a cluster. You can use it to access the cluster. If the cluster has multiple nodes, the private network addresses and port numbers of all nodes are displayed.
VPC Endpoint Address	IP address or private domain name of the VPC endpoint associated with the cluster.
Security Mode	Whether the security mode is enabled for the cluster. Only authorized users can access a security-mode cluster.
Enterprise Project	Enterprise project that a cluster belongs to.
Tags	Displays cluster tags. You can click **Edit Tag** to add, delete, or modify cluster tags.
Cluster Description	Provides supplementary information about the cluster.
Created	Cluster creation time.
Billing Mode	Billing mode of a cluster.
Operation	Operations that can be performed on a cluster, such as checking metrics, restart, and deletion. If an operation is unavailable, the menu button is grayed out.

Viewing Basic Cluster Information

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters** > **OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, check the cluster's basic information, node configuration, and network information.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-227 Basic cluster information

Parameter	Description
Cluster ID	Unique ID of a cluster, which is automatically generated by the system. Each cluster ID is unique within the same region.
Cluster Name	Cluster name. You can click to change it. **Initial Cluster Name** cannot be changed.
Initial Cluster Name	This name is unique and cannot be changed. It is only used in the cluster configuration file in YAML.
Cluster Version	Cluster version. For example, if the version is **1.3.6(1.3.6_x.x.x_x.x.x)**, **1.3.6** is the cluster version and **1.3.6_x.x.x_x.x.x** enclosed in parentheses is the image version. **CAUTION** • For a cluster created a long time ago, its image version may not be available because it is likely not recorded in the database. • If **EOS** is displayed below the version number, the current version has reached EOS and hence poses certain service availability risks. You are advised to upgrade the cluster version as soon as possible. For details, see **Upgrading the Cluster Version**.
Cluster Status	Current status of a cluster
Task Status	Current task status of a cluster. If there is no ongoing task, **--** is displayed.
Region | AZ	The region and AZ where the cluster is located.
Enterprise Project	Enterprise project that a cluster belongs to. You can click a project name to check the basic project information.
Cluster Description	Provides supplementary information about the cluster. You can click to modify the description.

Table 3-228 Node information

Parameter	Description
Data Node Specifications	The specifications of data nodes used in the cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Data Nodes	The number of data nodes in the cluster.
Data Node Storage Type	The storage type of data nodes in the cluster.
Data Node Capacity (GB)	Storage capacity per data node in the cluster.
Master Node Specifications	The specifications of master nodes used in the cluster. This parameter is displayed only when dedicated master nodes are used in the cluster.
Master Nodes	The number of master nodes in the cluster. This parameter is displayed only when dedicated master nodes are used in the cluster.
Master Node Storage Type	The storage type of master nodes in the cluster. This parameter is displayed only when dedicated master nodes are used in the cluster.
Master Node Capacity (GB)	Storage capacity per master node in the cluster. This parameter is displayed only when dedicated master nodes are used in the cluster.
Client Node Specifications	The specifications of client nodes used in the cluster. This parameter is displayed only when dedicated client nodes are used in the cluster.
Client Nodes	The number of client nodes in the cluster. This parameter is displayed only when dedicated client nodes are used in the cluster.
Client Node Storage Type	The storage type of client nodes in the cluster. This parameter is displayed only when dedicated client nodes are used in the cluster.
Client Node Capacity (GB)	Storage capacity per client node in the cluster. This parameter is displayed only when dedicated client nodes are used in the cluster.
Cold Data Node Specifications	The specifications of cold data nodes used in the cluster. This parameter is displayed only when cold data nodes are used in the cluster.
Cold Data Nodes	The number of cold data nodes in the cluster. This parameter is displayed only when cold data nodes are used in the cluster.
Cold Data Node Storage Type	The storage type of cold data nodes in the cluster. This parameter is displayed only when cold data nodes are used in the cluster.
Cold Data Node Capacity (GB)	Storage capacity per cold data node in the cluster. This parameter is displayed only when cold data nodes are used in the cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Cluster Storage Capacity (GB)	Total storage capacity of the cluster. • Quota: the upper limit of the storage capacity allocated to the cluster. • Used: storage space already used by the cluster. The cluster's disk usage can be calculated as follows: Disk usage = (Used/Quota) x 100%
Node Details	View the details of all cluster nodes. Click **View**. In the displayed dialog box, view node details, including the node name/ID, status, and IP address.

Table 3-229 Network information

Parameter	Description
VPC	VPC to which a cluster belongs
Current Subnet	Subnet to which a cluster belongs. If the current subnet cannot meet your service requirements, click **Switch Subnet**. In the displayed dialog box, select another subnet. If no existing subnets meet your requirements, click **Create Subnet** to create a new subnet. **CAUTION** • After switching the subnet, the new subnet is used for adding nodes, including scaling, adding dedicated master/client nodes, and enabling VPC Endpoint. IP addresses will be allocated to newly added nodes from the new subnet. • A cluster can be associated with a maximum of two subnets. • Subnet switching does not affect the network settings of existing nodes.
Used Subnet	Subnet once used by the cluster.
Security Group	Security group to which a cluster belongs. To modify the security group of a cluster, click **Change Security Group** on the right. **WARNING** The selected security group must allow port 9200 in the inbound direction. Otherwise, the cluster may be inaccessible to external services. Furthermore, to enable OpenSearch Dashboards private network access, port 5601 must also be allowed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Security Mode	Security mode of a cluster. • Enabled: The security mode is enabled for the current cluster. • Disabled: The security mode is disabled for the current cluster. You can click to change the cluster's security mode. For details, see **Changing the Security Mode**.
Reset Password	Change the **admin** password for a security-mode cluster. Click **Reset** to change the password. Requirements for admin passwords: • The password can contain 8 to 32 characters. • The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. The following special characters are allowed: **~!@#$%^&*()-_=+|[{}];:,<.>/?** • Do not use the administrator name, or the administrator name spelled backwards, as the password. • Change the password periodically. This parameter is displayed only when **Security Mode** is **Enabled**.
HTTPS Access	Whether to enable HTTPS for a cluster. • Disabled: The cluster uses HTTP. For a cluster with the security mode disabled, HTTPS is disabled by default. • Enabled: The cluster uses HTTPS. Only security-mode clusters support HTTPS. If **HTTPS Access** is enabled, you can click **Download Certificate** to obtain the certificate used for access control. Currently, the security certificate cannot be used in a public network environment. For details about how to switch the network protocol for a security-mode cluster, see **Switching between HTTP and HTTPS in Security Mode**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Public Network Access	Public network access information of the cluster. • For a security-mode cluster with public network access enabled, the configured public IP address is displayed. • For a security-mode cluster with public network access disabled, **--** is displayed. When public network access is enabled for a cluster, you are advised to use a whitelist for access control. This parameter is displayed only when **Security Mode** is **Enabled**.
Bandwidth	Public network bandwidth of the cluster. Click **Edit** to change the bandwidth size. This parameter is displayed only when the security mode and public network access are both enabled for the cluster.
Public Network Access Control	You can configure public network access control for a security-mode cluster by IP addresses or IP address ranges. Click **Modify** to modify the access control whitelist. • If a whitelist is configured, only IP addresses that are on this whitelist can access the cluster over the public network. Click **+Add**. In the displayed text box, enter IP addresses or CIDR blocks that are allowed to access the cluster from the public network. Separate them using commas (,). Each value must be unique. An example of valid values: **192.168.1.1,10.0.0.0/24**. Examples of invalid values: **0.0.0.0**, **xx.xx.xx.xx/0**, **172.16.0.0-172.16.255.255**, non-standard formats (e.g., **192.168.1**), and duplicate values. • If no whitelist is configured, all public IP addresses can access the cluster. However, this can be a security risk and should be avoided. This parameter is displayed only when the security mode and public network access are both enabled for the cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Private IPv4 Address	Private IPv4 address and port number of a cluster. If the cluster has only one node, the IP address and port number of the only node are displayed, for example, **10.62.179.32:9200**. If the cluster has multiple nodes, the IP addresses and port numbers of all nodes are displayed, for example, **10.62.179.32:9200,10.62.179.33:9200**.

Table 3-230 Additional information

Parameter	Description
Billing Mode	Billing mode of a cluster.
Created	When a cluster was successfully created.

3.18.2 Checking Default Cluster Plugins

Some of the CSS enhancements over open-source OpenSearch are delivered as default cluster plugins. These plugins are automatically injected to each cluster node during cluster creation and are loaded when the JVM process starts. CSS allows you to check these plugins through the web console or via a standard RESTful API.

Constraints

The default cluster plugins cannot be manually uninstalled.

Checking the Plugins on the CSS Console

Quickly check all default plugins supported by the current cluster without writing any code.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Cluster Settings** > **Plugins**.
5. On the **Default Plugins** tab, check the default plugins supported by the current cluster.

Checking the Plugins via an API

Check the loading of default plugins on specific cluster nodes through an API, for example, when you need to troubleshoot the tokenizer.

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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. In the left navigation pane, choose **Dev Tools**.
5. Run the following command to check the cluster's plugins:

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

Example response:

```
name                 component                    version
css-test-ess-esn-1-1 analysis-dynamic-synonym     1.0.1
css-test-ess-esn-1-1 analysis-icu                 1.1.6
css-test-ess-esn-1-1 analysis-ik                  1.0.1
......
```

Table 3-231 Response parameters

Parameter	Description
name	Node name. Use this field to determine whether each plugin has been loaded on all nodes.
component	Unique plugin ID. This name is required when you configure mappings or the tokenizer.
version	Plugin version.

3.18.3 Modifying Cluster Parameters

When using an OpenSearch cluster, you may need to adjust cluster behavior based on service requirements. For example, frontend applications may need to access cluster APIs across domains, query performance may need to be optimized by adjusting the cache, or the OpenSearch Dashboards session timeout may need to be extended. CSS allows you to modify the **opensearch.yml**, **opensearch_dashboards.yml**, and **system.config** configuration files online to flexibly adjust core cluster parameters and visualization platform parameters. You can use this feature for scenarios such as configuring cross-domain access, optimizing query performance, and customizing OpenSearch Dashboards behavior.

Configuration File Description

A CSS cluster relies on the following configuration files. Each file has a different purpose and modification method, and changes to one file do not affect the others. You must modify them separately.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 3-232 Configuration files

Configuration File	Configuration Range	When Changes Take Effect
opensearch.yml	Core cluster configuration parameters, including cross-origin access, index reindexing, query caching, and thread pools.	After modification, all nodes load the new configuration when the cluster is manually restarted.
opensearch_dashboards.yml	Independent configuration parameters for OpenSearch Dashboards, the visualization platform, such as session timeout, default redirect page, and user interface display.	After modification, only OpenSearch Dashboards loads the new configuration when it is restarted. Cluster nodes are not affected. The changes apply only when OpenSearch Dashboards is accessed through a public or private network address. They do not apply if OpenSearch Dashboards is accessed in one click from the console.
system.config	Metadata file for the SAML identity provider (IdP). For details, see **SAML - OpenSearch Documentation**.	After configuration, a **descriptor.xml** file is generated in the OpenSearch config directory. All nodes load this file when the cluster is manually restarted.

How to apply the changes:

1. Modify parameters: Edit the parameter values for the required modules on the parameter configuration page.
2. Save changes: CSS writes the changes to the corresponding configuration file.
3. Restart the cluster or process: Nodes or components reload the new configuration file.
4. Apply the changes: The new configuration takes effect after the restart is complete.

Constraints

- **system.config** can be modified only for OpenSearch 2.19.0.
- Exercise caution when configuring custom parameters. Make sure custom parameters are supported by the corresponding component. Otherwise, the cluster or the corresponding component may fail to restart.
- After modifying parameters in **opensearch.yml** and **system.config**, you must manually restart the cluster for the changes to take effect.
- A maximum of 20 modification records are displayed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- You are advised to modify parameters and restart clusters during off-peak hours to minimize impact on services.

Modifying OpenSearch Configuration (opensearch.yml)

Modify core OpenSearch cluster parameters, including cross-domain access, index reindexing, query caching, and thread pools.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Cluster Settings** > **Parameter Settings** > **OpenSearch**.
5. Click **Edit** to modify module parameters as required.

Table 3-233 Module parameters

Module Name	Parameter	Description
Cross-domain Access	http.cors.allow-credentials	Whether to carry authentication information in cross-domain requests, that is, whether to contain the Access-Control-Allow-Credentials field in the response header. The value can be: • **true**: The response header contains the Access-Control-Allow-Credentials field. • **false** (default value): The response header does not contain the Access-Control-Allow-Credentials field.
	http.cors.allow-origin	Source IP addresses or domain names allowed for cross-domain access. When there are multiple values, separate them with commas (,). Example: **192.168.122.122:9200** or **192.168.1.1:9200, 192.168.1.2:9200**
	http.cors.allow-headers	Request header fields for cross-domain access. When there are multiple values, separate them using a comma (,) and a space. Example: **X-Requested-With, Content-Type, Content-Length** The value can contain a maximum of 1024 characters. It can only contain letters, digits, hyphens (-), underscores (_), and slashes (/).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Module Name	Parameter	Description
	http.cors.max-age	Maximum retention duration of pre-check responses in the browser cache. Unit: s Value range: 0 to 31,536,000 (approximately 1 year) Default value: 1,728,000 (20 days)
	http.cors.enabled	Whether to enable cross-domain access. The value can be: • **true**: Enable cross-domain access. • **false** (default): Disable cross-domain access.
	http.cors.allow-methods	HTTP methods allowed for cross-domain access. When there are multiple values, separate them using a comma (,) and a space. Example: **OPTIONS, GET, POST** Value range: OPTIONS, HEAD, GET, POST, PUT, and DELETE
Reindexing	reindex.remote.whitelist	Which remote hosts are allowed for remote reindexing operations. Use commas (,) to separate multiple values. Example: **192.168.122.122:9200** or **192.168.1.1:9200, 192.168.1.2:9200**
Custom Cache	indices.queries.cache.size	Maximum heap space allocated to the query cache. Unit: percentage (%) Value range: 1 to 100 Default value: 10%
Queue Size in a Thread Pool	thread_pool.force_merge.size	Size of the thread pool used for force merge operations. Value range: a positive integer Default value: 1
Custom	-	Click **Add** to add or modify custom parameters supported by OpenSearch. **WARNING** Set the values of any custom parameters to those supported by OpenSearch. Otherwise, the cluster may fail to restart. Exercise caution when performing this operation.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

6. After modifying the file, click **Save**. In the displayed dialog box, confirm the parameters and select the restart option, and then click **OK**.

   If **Job Status** is **Succeeded** in the parameter settings task list below, the change has been saved.

7. Click **Restart** in the upper right corner to restart the cluster and apply the change.
8. After restart, check that **Cluster Status** is **Available** in the cluster list. This state indicates the parameter changes have taken effect.

Modifying OpenSearch Dashboards Configuration (opensearch_dashboards.yml)

Modify OpenSearch Dashboards visualization platform parameters, such as the session timeout, default redirect page, and user interface display language.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Cluster Settings** > **Parameter Settings** > **Dashboards**.
5. Click **Edit** to modify module parameters as required.

Table 3-234 Module parameters

Module Name	Parameter	Description
Base Path	server.basePath	Custom value for Base Path. When you need to access OpenSearch Dashboards through a reverse proxy (such as Nginx), this value is appended to the proxy address as a subpath. For example, if the reverse proxy address is **http://example.com** and **server.basePath** is set to **/test**, you can access OpenSearch Dashboards through the address **http://example.com/test**. The value can be: • Must start with **/**. • Must not contain consecutive slashes (for example, **//**). • Can only contain letters, digits, hyphens (**-**), underscores (_), periods (**.**), and slashes (**/**). • Can contain multi-level paths, but its length cannot exceed 254 characters. • Must not end with a slash (/).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Module Name	Parameter	Description
	server.rewriteBasePath	Whether OpenSearch Dashboards or a reverse proxy will remove the server.basePath prefix from the request path. This parameter is optional. If no reverse proxy is used, ignore this parameter. The value can be: • **true**: OpenSearch Dashboards will automatically remove the server.basePath prefix from the request path. • **false** (default): The reverse proxy (such as Nginx) will remove the server.basePath prefix from the request path. In this case, you can only access OpenSearch Dashboards through the reverse proxy.
Custom	-	Click **Add** to add or modify custom parameters supported by OpenSearch Dashboards. **WARNING** Set the values of custom parameters to those supported by OpenSearch Dashboards. Otherwise, OpenSearch Dashboards may fail to restart. Proceed with caution.

6. After modifying the file, click **Save**. In the displayed dialog box, confirm the parameters and select the restart option, and then click **OK**. Modifying OpenSearch Dashboards settings will restart OpenSearch Dashboards, making it temporarily unavailable.

   If **Job Status** is **Succeeded** in the parameter settings task list below, the parameter changes have taken effect.

Modifying System Configuration (system.config)

CAUTION

system.config can be modified only for OpenSearch 2.19.0.

Modify system configuration parameters. A **descriptor.xml** file is generated or modified in the same directory as the **opensearch.yml** configuration file.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

4. Choose **Cluster Settings** > **Parameter Settings** > **System**.
5. Click **Edit** and enter the content of the **descriptor.xml** file in the **System** area.

   The file size cannot exceed 100 KB (102,400 bytes or characters, including line breaks).
6. After modifying the file, click **Save**. In the displayed dialog box, confirm the parameters and select the restart option, and then click **OK**.

   If **Job Status** is **Succeeded** in the parameter settings task list below, the change has been saved.
7. Click **Restart** in the upper right corner to restart the cluster and apply the change.
8. After restart, check that **Cluster Status** is **Available** in the cluster list. This state indicates the new **descriptor.xml** file has taken effect.

Related Documents

- After configuration changes, you must restart the cluster to apply the changes. For details, see **Restarting a Cluster**.
- For more parameters supported by OpenSearch, see the **OpenSearch official documentation**.
- For more parameters supported by OpenSearch Dashboards, see **OpenSearch Dashboards official documentation**.

FAQ

- **Q1: What should I do if the cluster status shows Configuration error after modifying parameters?**

  Configuration error indicates that the modified configuration file contains invalid parameter values. Check the specific error message in the parameter configuration task list, correct the parameter values as prompted, save the changes, and then restart the cluster again. If the error is caused by a custom parameter, verify whether the parameter is officially supported by the corresponding component.
- **Q2: What will happen if I forget to restart the cluster after modifying parameters?**
  - If you do not restart the cluster after modifying parameters in **opensearch.yml** and **system.config**, the changes do not take effect and the cluster continues to run with the old configuration. **Task Status** in the cluster list shows **Configuration Not Updated**. You are advised to restart the cluster immediately after modifying parameters.
  - Modifying parameters in **opensearch_dashboards.yml** automatically restarts the OpenSearch Dashboards process, so there is no need to manually restart the cluster. The new configuration takes effect immediately after the OpenSearch Dashboards process is restarted.

3.18.4 Managing Cluster Tags

As business grows, an enterprise may operate hundreds of OpenSearch clusters, and cluster names alone will be insufficient for categorization and identification. For example, tasks such as resource monitoring or automated operations via scripts require the ability to quickly locate clusters based on specific attributes. On

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

CSS, you can use tags (key-value pairs) to label resources, enabling efficient filtering and batch management. When used together with Tag Management Service (TMS), tags help you implement enterprise-grade resource governance.

Constraints

- Each cluster can have a maximum of 20 tags.
- If your organization has configured tag policies for CSS, add tags to clusters based on these policies. If a tag does not comply with existing tag policies, cluster creation may fail. Contact the administrator to learn more about these policies.
- To use the same tag to identify multiple cloud resources for better resource grouping, predefine tags in Tag Management Service (TMS). For details, see **Predefined Tags Overview**.

Managing Cluster Tags in Batches

Manage tags for multiple clusters in batches, including add, modify, and delete. This feature is useful in cases such as environment refactoring and large-scale migrations.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, select one or more clusters.
4. Click **Edit Tag** above the cluster list.
   - Add

     In the displayed dialog box, click **+Add**, and add new tags in the form of key-value pairs. The tags will be added for all selected clusters.
   - Modify

     In the displayed dialog box, modify existing tag values. The changes will be applied to all selected clusters, overwriting existing tags. Avoid this operation if you have custom tags that you wish to keep.
   - Delete

     In the displayed dialog box, click **Delete** next to the tag you want to delete. This deletes the tag for all selected clusters. The deletion cannot be undone.
5. Click **OK** to save the changes.
6. Verify the result: In the cluster list, hover the mouse over the icon in the **Tag** column for each cluster. Check that the tag information has been updated for all selected clusters.

Managing Tags for a Single Cluster

Add, modify, or delete tags for a single cluster.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

4. Click the **Tags** tab.
5. Click **Edit Tag**. In the displayed dialog box, modify the cluster's tags.
   - Add

     In the displayed dialog box, click **+Add**, and add new tags in the form of key-value pairs.
   - Modify

     In the displayed dialog box, modify existing tag keys and values.
   - Delete

     In the displayed dialog box, click **Delete** next to the tag you want to delete.
6. Click **OK**.
7. Verify the result: In the tag list, check that the tag keys and values have been updated.

Searching for Clusters by Tag

Locate clusters by tag from among a large number of resources.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. Click the search box above the cluster list. Select a tag key, operator, and tag value to search.
   - You can select a tag key and value from the drop-down list. The system returns a list of clusters that exactly match the tag key and value.
   - In case you use multiple tags to start a search: With the same tag key but different values, the system returns a list of clusters that match any of the tag values; If the tag keys are different, the system returns a list of clusters that match all of these keys and tags.

   The cluster list is filtered in real time to display matching clusters only.

3.18.5 Associating a Cluster with an Enterprise Project

In practice, it is common for multiple departments (such as R&D and quality management) within a large organization to share a single Huawei Cloud account. Enterprise projects let you manage OpenSearch clusters by department or project, enabling clearer resource visibility and more fine-grained access control across regions within the shared account.

How the Feature Works

Enterprise projects are a cloud resource management feature provided by Huawei Cloud that allow you to group cloud resources by project, department, or business. For more information, see **Enterprise Project Management Service**.

A newly created cluster belongs to the **default** project by default. To move a cluster from the default project to a custom project, you simply perform a resource migration. There is no need to re-create the cluster.

Here, resource migration is the process of moving a CSS cluster from one enterprise project (for example, **default**) to another (for example, **TEST**).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- Impact: Resource migration only changes the management metadata. It does not restart the cluster or interrupt services.
- Permission control change: After the migration, only users who have been granted permissions on the new project (or all resources) can manage the cluster.

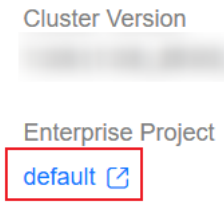
Prerequisites

- Enterprise Project Management Service (EPS) has been enabled for your account. For details, see section "Enabling Enterprise Project" in *Enterprise Management User Guide*Enabling Enterprise Project.
- You have created the destination enterprise project (for example, **TEST**). For details, see section "Creating an Enterprise Project" in *Enterprise Management User Guide*Creating an Enterprise Project.

Migrating a Cluster Between Enterprise Projects

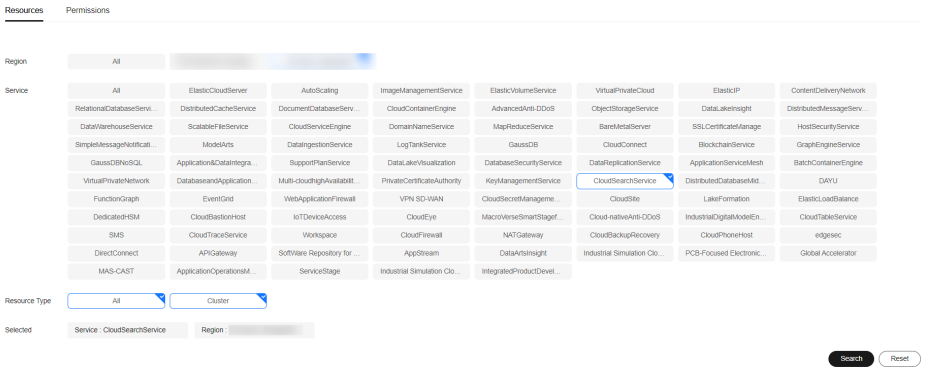
1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, find **Enterprise Project** in the **Basic Information** area, and click the enterprise project name under it. The **Project Management** console is displayed.

   Figure 3-99 Going to the Project Management console

5. On the **Resources** tab on the Project Management console, select the region of the current cluster, and select **CloudSearchService** as **Service**. Then click **Search**. The CSS clusters are displayed in the resource list below.

   Figure 3-100 Filtering CSS clusters

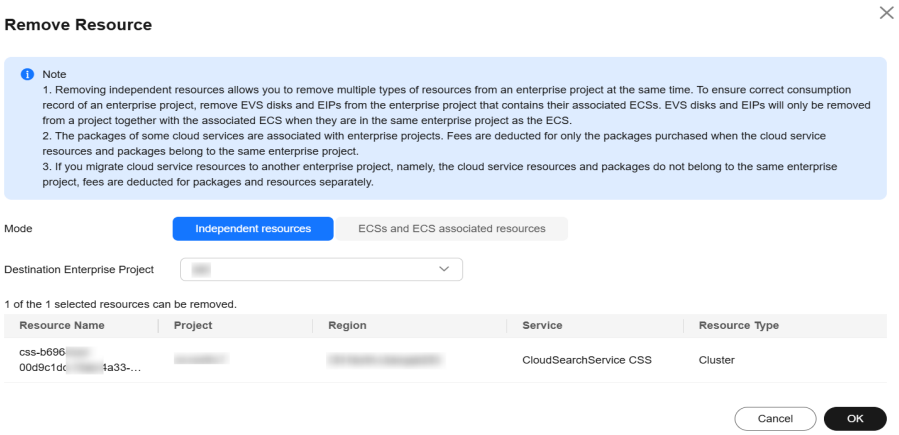
   

6. Select the cluster whose enterprise project you want to modify and click **Remove**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

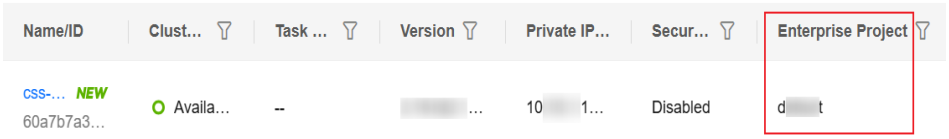
7. In the **Remove Resource** dialog box, specify **Mode** and select **Destination Enterprise Project**. Confirm the change information, and click **OK**.

Figure 3-101 Remove Resource



8. After the change is complete, return to the OpenSearch cluster list on the CSS console and check the cluster's new enterprise project to confirm the operation result.

Figure 3-102 Enterprise Project



FAQ

- **Question 1: Does changing the enterprise project change a cluster's IP address?**

  Changing a cluster's enterprise project does not change its IP address. It merely changes the logical ownership of cloud resources. There is no change at the network layer.

- **Question 2: What should I do if, after a cluster's enterprise project is changed, the cluster is hidden from users under the original project?**

  After the change, the cluster management permissions are transferred to the new project. Check whether your account has the read permission on the new (destination) enterprise project.

3.18.6 Restarting a Cluster

A cluster restart may be necessary for purposes such as configuration updates or fault recovery. CSS OpenSearch clusters support two restart modes: quick restart and rolling restart. To balance cluster availability and restart efficiency, select the mode best suited to your operational requirements.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Comparing the Two Restart Modes

Table 3-235 Introducing the two restart modes

Restart Mode	Quick Restart	Rolling Restart
Impact	High: The cluster becomes unavailable for both read and write operations for the entire duration of the upgrade.	Low: Only the node that is being restarted is unavailable. The cluster as a whole remains available all the time.
How It Works	Parallel operations: All selected nodes are stopped and restarted at once.	Serial operations: Restart node A > Wait for it to recover > Restart node B > Wait for it to recover, and so on
Data Security	There is risk of data loss if you forcibly restart nodes that are still serving write requests.	Data security is guaranteed as the system automatically handles shard allocation.
Time Required	All nodes are restarted simultaneously. The total restart time is typically shorter.	The selected nodes are restarted one at a time. After the first node, each node can restart only after the previous node has fully recovered. This process takes longer.
When to Use	Use this mode in development or testing environments, where clusters are completely deadlocked, or where service interruption is acceptable.	Use this model in production environments, where high availability takes top priority and clusters store large amounts of data.
Node Quantity	There is no limit on the number of nodes.	There are at least three nodes.

Constraints

Clusters whose **Task Status** is **Frozen** or that have ongoing tasks cannot be restarted.

Before You Start

- Check the target cluster's health status. Check the **Task Status** column to ensure there are no ongoing tasks.
- Pause high-concurrency data ingestion tasks to prevent write failures caused by primary shard changes during the restart.
- Avoid restarting a cluster during peak hours, as doing so may impact data integrity and service availability.
- Before performing a **rolling restart**, check that indexes have at least one replica configured.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Restart Considerations for Different Node Types

Before restarting nodes by node type or node name, review the following considerations based on the role of the target nodes.

Table 3-236 Restart considerations

Node Type	Restart Considerations

Data nodes and cold data nodes	• In production environments, you are advised to perform rolling restart in the following sequence: **Pause shard allocation > Restart one node > Wait for shard recovery to complete > Restart the next node**. For details, see **Related Operation: Managing Shard Allocation Policies**.
	• For clusters with a large number of shards, you are advised to increase the delayed shard allocation timeout to prevent a node's temporary unavailability from triggering unnecessary shard reallocation and placing excessive load on the cluster. The default timeout is 1 minute and can be adjusted as needed:
	`PUT _all/_settings`
	`{`
	`  "settings": {`
	`  "index.unassigned.node_left.delayed_timeout": "5m"`
	`  }`
	`}`

Master nodes	• For production clusters, do not restart all master nodes at once. Master nodes manage cluster metadata. Restarting all master nodes at once triggers metadata reallocation and prolongs cluster recovery.
	• You are advised to restart master nodes one at a time by node name, with the current master node restarted last.
	Run the following command to identify the current master node (the node with ***** in the **master** column is the current master node):
	`GET /_cat/nodes?v`
	Example response:
	`ip heap.percent ram.percent cpu load_1m node.role master name`
	`192.168.67.132 54 96 5 0.79 dimr * css-ess-esn-3-1`
	`192.168.100.80 22 94 4 0.58 dimr - css-ess-esn-1-1`

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Node Type	Restart Considerations
Client nodes	• Client nodes are responsible only for request forwarding and preprocessing. Restarting a client node does not trigger shard reallocation, but requests routed to that node will fail. Before restart a client node, check whether any applications connect directly to the node's IP address. • Client nodes can be restarted in batches and recover quickly. You can also switch the application access IP addresses in advance to avoid request failures.

Quick Restart

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and choose **More > Restart** in the **Operation** column.
4. On the **Restart Cluster** page, set parameters.

   Table 3-237 Configuring a quick restart

Parameter	Description
Restart Mode	Select **Quick Restart**.
Select By	Select **Node type** or **Node name**. • **Node type**: Select one or more node types for restart. For example, you can use this option to restart all client nodes to release connections. • **Node name**: Restart only one node at a time. Use this option to restart individual nodes, for example, to restore faulty nodes.
Value	Select one or more node types or a node name, depending on the setting of **Select By**. See **Restart Considerations for Different Node Types** for more information.

5. Manually type in **YES**, and click **OK** to start the restart task.

   WARNING

   The cluster will be unavailable for the entire duration of a quick restart.

6. When the cluster status changes from **Processing** to **Available**, the cluster has restarted successfully.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Rolling Restart

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and choose **More > Restart** in the **Operation** column.
4. On the **Restart Cluster** page, set parameters.

   Table 3-238 Configuring a rolling restart

Parameter	Description
Restart Mode	Select **Rolling Restart**. This option is unavailable when the cluster has fewer than three nodes.
Select By	The default value is **Node type**.
Value	Select node types for the restart. See **Restart Considerations for Different Node Types** for more information. • **Select all**: Select all node types. • Select specific node types: For instance, to restart cold data nodes, select **Cold Data Nodes** only. The system will skip all other nodes.

5. Manually type in **YES**, and click **OK** to start the restart task.
6. When the cluster status changes from **Processing** to **Available**, the cluster has restarted successfully.

Related Operation: Managing Shard Allocation Policies

NOTE

During a rolling restart, the system automatically handles shard allocation, so normally you do not need to manually change this setting. This operation is primarily intended for scenarios where nodes are manually restarted in batches by node type or node name during a quick restart.

Node restart can trigger shard reallocation, especially when nodes are manually restarted in batches by node type or node name during a quick restart. You are advised to disable automatic shard allocation before restarting the nodes and manually restore the automatic shard allocation policy after the restart is complete. This prevents a large number of shards from being reallocated when nodes go offline, which could otherwise cause the cluster to remain unavailable for an extended period.

1. Disable automatic shard reallocation.

```
PUT /_cluster/settings
{
  "persistent": {
    "cluster.routing.allocation.enable": "none"
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. After all target nodes have been restarted, restore the automatic shard allocation policy.

```
PUT /_cluster/settings
{
  "persistent": {
    "cluster.routing.allocation.enable": "all"
  }
}
```

FAQ

- **Question 1: What should I do if a rolling restart is stuck in Restarting for a long time?**

  During a rolling restart, after the last node is restarted, the system waits for the cluster to recover (health status turns green/yellow). If the cluster has extremely large indexes or index shards are still being restored, the task will be stuck in **Restarting** for a long time. You can run the **GET _cat/recovery?v** command to check the shard recovery progress.

- **Question 2: How do I perform a rolling-like restart on a cluster with just two nodes?**

  The rolling restart option is unavailable for clusters with just two nodes, but you can achieve a similar process manually:

  a. Check the number of index replicas. To ensure service continuity, ensure that each index has at least one replica.

  b. Select **Fast Restart**, then select **Node name** for **Select By**. Select one of the nodes to restart it.

  c. Wait for this node to fully recover and the cluster status to change to **Available**. Then restart the other node.

3.18.7 Deleting a Cluster

When an OpenSearch cluster is no longer required (for example, after a project is decommissioned or a temporary environment is retired), you can delete the cluster to release compute and storage resources and stop billing. Cluster deletion is an irreversible high-risk operation that permanently removes all data in the cluster. Before deleting a cluster, ensure that all the required data has been backed up.

Before You Start

Cluster deletion is an irreversible high-risk operation. Before proceeding, confirm the following:

- **Data has been backed up**: Ensure that cluster data has been backed up to an OBS bucket using snapshots. Without backups, the data will be permanently lost once deleted.
- **There are no ongoing tasks**: Check cluster monitoring metrics and ensure that both the average indexing rate and average query rate are 0, indicating that the cluster has no ongoing tasks.
- **Snapshot availability has been confirmed**: If snapshots have been enabled for the cluster and snapshots are still retained in the OBS bucket, you can use these snapshots to restore data later if necessary. For details, see **Can a Deleted CSS Cluster Be Restored?**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- **VPC Endpoint permissions have been confirmed**: If the cluster has enabled the VPC Endpoint service, the account that you use must have permission to delete VPC endpoints. Otherwise, VPC endpoint resources cannot be released after cluster deletion, and you will continue to be charged for them. For details about how to configure this permission, see **Permissions**.
- Dedicated load balancer configuration has been confirmed: If the cluster is connected through a dedicated load balancer (ELB), deleting the cluster releases the load balancer's listener resources. However, you must manually delete the ELB to prevent further charges.

Deleting a Pay-per-Use Cluster

The cluster is deleted immediately, resources are released, and pay-per-use billing stops.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > OpenSearch**.
3. In the cluster list, find the target cluster, and choose **More > Delete** in the **Operation** column.
4. In the displayed dialog box, read the risk warning carefully and enter **DELETE** in the text box.
5. Click **OK**.

   After the cluster is deleted, it is removed from the cluster list.

FAQ

- **Q1: Will snapshots stored in an OBS bucket be deleted together with the cluster?**

  No. Deleting a cluster only releases its compute and storage resources. Snapshots stored in OBS will be retained. You can use these snapshots to restore data to a new cluster. However, note that you will continue to be charged for OBS storage. If you do not need these snapshots, manually delete them.
- **Q2: Can data be restored if a cluster is accidentally deleted?**

  If the cluster has snapshots stored in an OBS bucket, you can restore its data from one of these snapshots. If snapshots were never enabled or no snapshots were ever created for the cluster, then there is no way to restore cluster data. For production clusters, enable an automatic snapshot policy to create snapshots periodically and reduce the risk of data loss.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

4 Logstash

4.1 Procedure

This topic introduces the end-to-end procedure for using CSS Logstash to build data ingestion and processing pipelines for enterprise-grade search and analytics platforms, covering key steps from environment setup to application rollout.

Figure 4-1 Procedure for using CSS Logstash

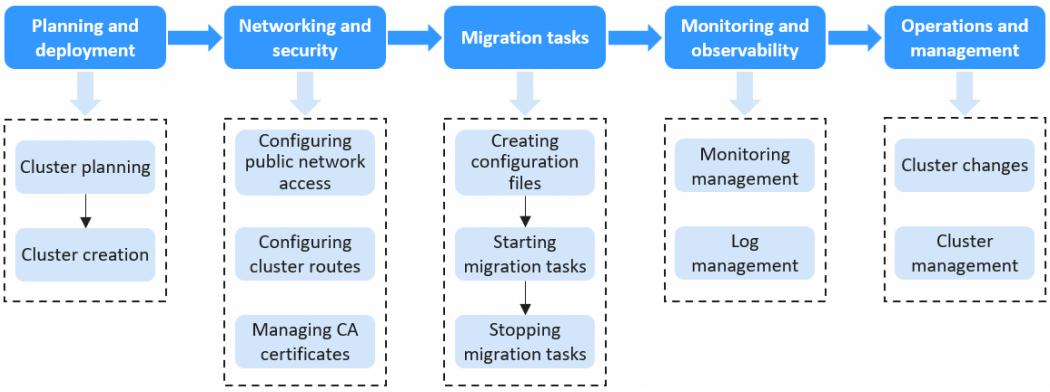


Table 4-1 Procedure for using Logstash

Stage	Key Step	Description	Details
Planning and deployment	Cluster planning	Based on intended use cases, plan node specifications and multiple-AZ deployment in advance to minimize the need for changes after cluster creation.	**Planning Cluster AZs and HA** **Planning Node Specifications and Capacity**
	Creating a cluster	Create a Logstash cluster based on your planning and service requirements.	**Creating a Cluster (New Version)** **Creating a Cluster**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Stage	Key Step	Description	Details
Networking and security	Network and security settings	If the Logstash cluster and the data source reside in different VPCs or they can only communicate through the public network, you must configure routes or enable public network access for the Logstash cluster to connect the two.	**Configuring Public Network Access** **Configuring Routes for a Cluster** **Managing CA Certificates**
Migration tasks	Configuring migration tasks	Through the configuration center of the Logstash cluster, create and start a Logstash configuration file to migrate data between different locations.	**Creating a Configuration File** **Starting Migration Tasks** **Stopping Migration Tasks**
Monitoring and observability	Monitoring management	Cloud Eye enables efficient visualization and monitoring of key cluster metrics, allowing the O&M team to monitor cluster health and performance bottlenecks in real time. This helps ensure high service availability.	**Using Cloud Eye to Monitor Clusters** **Supported Metrics**
	Log management	The O&M team can analyze run logs and operation audit records to quickly demarcate issues, analyze performance, and ensure security and compliance. Logs can be backed up to OBS for long-term storage, meeting diverse, enterprise-grade operational and auditing requirements.	**Querying Logs** **Backing Up Logs to OBS** **Viewing Audit Logs**
Operations and management	Managing cluster changes	You can easily manage cluster changes, such as cluster scaling and changing node specifications, while ensuring service continuity.	**Adding New Nodes** **Reducing Nodes** **Changing Node Specifications**

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Stage	Key Step	Description	Details
	Cluster management	Cluster resources can be isolated by enterprise project and tagged for refined management. You can easily restart and delete clusters, ensuring efficient cluster management.	**Viewing Cluster Details** **Managing Cluster Tags** **Associating a Cluster with an Enterprise Project** **Forcibly Restarting a Cluster** **Deleting a Cluster**

4.2 Cluster Planning

4.2.1 Planning Cluster AZs and HA

This topic describes how to improve cluster availability by deploying each cluster across multiple AZs, including defining the rules that govern how cluster nodes are distributed across AZs.

An availability zone (AZ) is a physical region where resources use independent power supplies and networks. AZs in the same region can communicate with each other through internal networks but are physically isolated.

Multi-AZ deployment is a high availability feature provided by CSS. Deploying a cluster across two AZs located in the same region can help prevent data loss and lower the possibility of service outages.

Suggestions on Multi-AZ Deployment

If you select multi-AZ deployment when creating a cluster, CSS automatically enables cross-AZ HA to ensure that cluster nodes will be evenly distributed across the selected AZs (the difference between the number of nodes in each AZ cannot exceed 1).

A Logstash cluster can span a maximum of two AZs.

Node Distribution Rules

When a multi-AZ cluster is created, nodes of all types are evenly distributed across different AZs. A maximum of two AZs can be used. **Table 4-2** shows the node distribution when the number of AZs varies.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

WARNING

When creating a multi-AZ cluster, select at least two nodes. Otherwise, multi-AZ cluster deployment will fail.

Table 4-2 Node distribution when the numbers of nodes and AZs vary

Node Quantity	Single AZ	Two AZs	
	AZ1	AZ1	AZ2
1	1	Not supported	
2	2	1	1
3	3	2	1
4	4	2	2
...	...	...	...

4.2.2 Planning Node Specifications and Capacity

This topic provides suggestions on selecting node specifications and configuring the node storage type, storage capacity, and node quantity for a Logstash cluster, helping you properly plan the capacities of your cluster.

Node Configuration Suggestions

Table 4-3 Node configuration

Parameter	Configuration Suggestions
Node Specifications	In the node flavor list, **vCPUs | Memory** indicates the number of vCPUs and memory capacity available for each flavor, and **Recommended Storage** indicates the supported storage capacity range. We recommend that you select node specifications based on core metrics of your Logstash cluster, such as the CPU load, memory requirements, and I/O features. **Logstash Node Specifications** describes the application scenarios and core features of different node specifications. It can help you properly plan your cluster. For more information about different node specifications, see **ECS Types**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Configuration Suggestions
Node Storage Type and Capacity	Select an appropriate storage type and capacity for cluster nodes. • Select an EVS disk type: – If you select **persisted** (disk storage) for event buffering, you are advised to select EVS disks with better performance, such as **Ultra-high I/O** and **Extreme SSD**. – If you use memory queues, select high I/O EVS disks. **NOTE** By default, Logstash uses in-memory bounded queues to buffer events. The corresponding configuration item is **queue.type**, which is available when you configure pipelines in the configuration file. For more on EVS disk performance, see **Disk Types and Performance**. • The node storage capacity is fixed at 40 GB.
Nodes	The number of nodes in a Logstash cluster ranges from 1 to 100. Logstash nodes are used to ingest, parse, process, and transfer data. The number of nodes determines the data migration speed. Select the number of Logstash nodes based on service requirements. When the Logstash cluster has two or more nodes, all nodes use the same configuration files. This mode works when Logstash is a consumer of Kafka data.

Logstash Node Specifications

Logstash nodes support EVS disks only. EVS disks are a virtual block storage service that is independent of ECSs. They provide high reliability and fast elasticity, making them ideal for workloads that require high data reliability and highly scalable storage capacity.

The table below describes the application scenarios and core features of different node specifications. It can help you properly plan your cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 4-4 Comparing different node specifications

CPU Architecture	Node Flavor	Description
x86	Compute-intensive	**Core advantages** • High-performance CPU: designed for high computational load, ideal for CPU-intensive tasks. • Optimized network I/O: supports high throughput in both the inbound and outbound directions (such as network plugins). **Application scenarios** • CPU-intensive plugins: plugins that involve heavy CPU computation, such as grok (regular expression parsing) and dissect (structured log parsing). • Hybrid load tasks: tasks that involve both heavy CPU computation and network I/O loads, such as real-time log ingestion and data cleaning. • Large-scale data processing at high speed, such as log aggregation and event stream processing. **Precautions** • Set the number of task threads (pipeline.workers) to equal the number of vCPUs. This optimizes CPU utilization while avoiding resource contention. • Pay attention to warnings on I/O performance bottlenecks. If the load of network plugins (such as beats and http) is heavy, make sure there is sufficient network bandwidth.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

CPU Architecture	Node Flavor	Description
	General computing-plus-AC	**Core advantages** • Dedicated CPU: no resource contention between different instances, stable performance at a relatively low cost, ideal for high-priority tasks. • Low latency: guaranteed efficiency for CPU-intensive plugins. **Application scenarios** • High CPU-load tasks: real-time log parsing and complex field extraction (grok and ruby) • Mission-critical service processes: reliable performance required (such as financial transaction log processing) • Multi-thread processing: high-concurrency tasks **Precautions** If tasks mainly involve high network I/O, you are advised to use ultra-high I/O disks.
	General computing	**Core advantages** Balanced configuration: default specifications, suitable for medium-scale data processing tasks. **Application scenarios** • Medium-scale log processing, such as enterprise log ingestion and monitoring data aggregation. • Low CPU-load tasks: mainly network I/O (such as file and kafka plugins). • Standard deployment: meets the needs of general use cases, no need for special tuning. **Memory usage evaluation** Estimate the needed memory capacity using this formula: **Average size of each piece of data processed by Logstash x (pipeline.workers x pipeline.batch.size)** Example: If the average data size is 1 KB, pipeline.workers = 4, and pipeline.batch.size = 1000, the memory size is ~4 MB.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

CPU Architecture	Node Flavor	Description
	Memory-optimized	**Core advantages** • Large memory capacity: suitable for memory-intensive tasks, reduced disk I/O pressure. • Optimized memory queues: data cached in memory queues, more efficient data processing. **Application scenarios** • Large-scale log aggregation: log analytics platform, security information and event management (SIEM), etc. • Complex data transformation: tasks that involve temporary storage of large amounts of data (for example, the aggregate plugin). **Precautions** • Monitor the memory usage in real time to avoid out of memory (OOM). • This cost is high. Use this flavor for memory-intensive tasks that have high priorities.

4.3 Creating a Cluster (New Version)

Logstash clusters created in CSS can be used for efficient ingestion, parsing, processing, and transmission of vast quantities of data. They support real-time log analytics, data integration and transformation, and cross-system data transfers while providing capabilities for data monitoring, alarming, as well as in-depth analysis and visualization.

You can create a Logstash cluster in either of the following ways:

- Method 1: Create a Logstash cluster on the CSS management console. This topic provides the operation guide for the default UI (that is, the new UI). For the operation guide for the old UI, see **Creating a Cluster**.

  NOTE

  To improve efficiency in creating clusters, CSS has made several enhancements on the cluster creation page in terms of ease-of-use. Being more intuitive, the new UI streamlines cluster creation procedures.

- Method 2: Create a Logstash cluster using CSS APIs. For details, see **Creating a Cluster**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

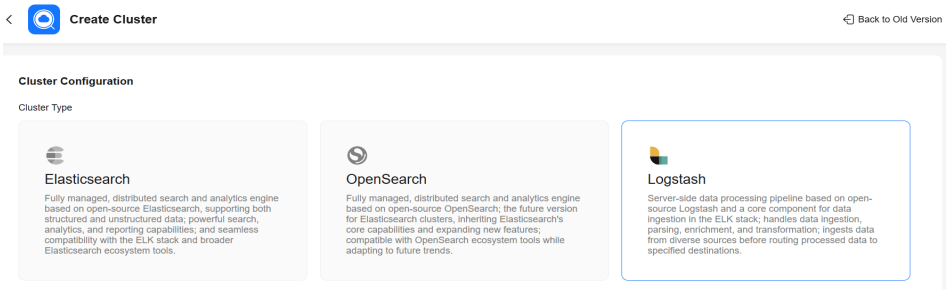
Prerequisites

- You have planned the Logstash cluster configuration by following the guidelines in **Cluster Planning**, including the cluster size, node configuration, and storage capacity requirements.
- You have created a VPC, subnet, and security group beforehand, or you can create them when creating the cluster. For more information, see Virtual Private Cloud User Guide.

Opening the Create Cluster Page

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. In the upper right corner, click **Create Cluster**. The new-version UI for creating a cluster is displayed by default.

Figure 4-2 Create Cluster (new version)



Cluster Configuration

Select the cluster type and version.

Table 4-5 Cluster Configuration

Parameter	Description
Cluster Type	Select **Logstash**.
Cluster Version	Select a cluster version from the drop-down list.

Basic Settings

Select the region and AZ of the cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 4-6 Basic settings

Parameter	Description
Region	Select the region where the cluster is located. A region is the location of a physical data center. Regions are defined based on their geographical location and network latency. For lower network latency and quicker resource access, select the nearest region.
AZ	Select one or more AZs associated with the cluster region. An AZ is a physical region where resources use independent power supplies and networks. AZs are physically isolated but interconnected through an internal network. A maximum of two AZs can be configured. For details about the use of multiple AZs, see **Suggestions on Multi-AZ Deployment**.

Configuring Logstash Nodes

Configure the nodes of the Logstash cluster.

Table 4-7 Configuring Logstash nodes

Parameter	Description
CPU Architecture	Select the CPU architecture of the Logstash nodes. **x86** nodes are supported. The architectures actually supported may vary depending on the regional environment.
Node Specifications	Select the specifications of the Logstash nodes. Click **Available**. On the displayed page, select a flavor that suits your needs. The nodes of a Logstash cluster support EVS disks only. For more information about different node specifications, see **ECS Types**. In the node flavor list, **vCPUs | Memory** indicates the number of vCPUs and memory capacity available for each flavor, and **Recommended Storage** indicates the supported storage capacity range. The node flavors available may vary depending on the region you select.
Node Storage Type and Capacity	Select the storage type of the Logstash nodes. • Available EVS disk types vary depending on your region. For more on EVS disk performance, see **Disk Types and Performance**. • The node storage capacity is fixed at 40 GB.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Nodes	Set the number of nodes for the Logstash cluster. The number of nodes determines the data migration speed. Select the number of Logstash nodes based on service needs. Value range: 1 to 100.

Network Settings

Configure the VPC, subnet, IP addresses, and security group of the cluster.

Table 4-8 Configuring network settings

Parameter	Description
VPC	Select a VPC for the cluster for proper network isolation. If none of the existing VPCs meets your requirements, click **Create VPC** to create one. (Make sure you have the permission to create VPCs.) For details, see **Creating a VPC and Subnet**.
Subnet	Select a subnet for the cluster. A subnet improves network security by providing exclusive network resources that are isolated from other networks. Select a subnet in the current VPC. If no existing subnets meet your requirements, click **Create Subnet** to create a new subnet. For details, see **Creating a VPC and Subnet**.
IPv4 Address	Assign IPv4 addresses to cluster nodes. • **Assign automatically**: Private IPv4 addresses will be automatically assigned to cluster nodes. • **Assign manually**: Manually assign private IPv4 addresses to cluster nodes. Before assigning an IP address, click **View In-Use IP Address** to check IP addresses that are already in use. If you manually specify fewer IP addresses than the total number of cluster nodes, the system will automatically assign random private IPv4 addresses to the remaining nodes.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Security Group	Select a security group for the cluster. A security group serves as a virtual firewall that provides access control policies for clusters. To select a security group that meets your requirements, click **View Security Group** to go to the security group list, where you can check the details of each security group. The selected security group must allow port 9600 in the inbound direction. Otherwise, the cluster may be inaccessible to external services. If none of the existing security groups meets your requirements, click **Create Security Group** to create one. (Make sure you have the permission to create security groups.) For details, see **Creating a Security Group**.

Cluster Management

Configure the cluster name, enterprise project, and more.

Table 4-9 Cluster management information

Parameter	Description
Cluster Name	User-defined cluster name. The cluster name must start with a letter and can contain 4 to 32 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
Add Description	Add a description for the cluster for easy recognition. Click **+ Add Description** and then enter a description of the cluster. The value can contain 0 to 128 characters.
Enterprise Project	Associate the cluster with an enterprise project. An enterprise project groups cloud resources, so you can manage resources and members by project. The default project is **default**. If enterprise projects are enabled, you can select an enterprise project from the drop-down list. You can click **View Enterprise Projects** to go to the Enterprise Project Management Service console and check enterprise project details. For more information, see **Accessing the Enterprise Center**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Tags	Adding tags to clusters helps you identify and manage your cluster resources. Click **+ Add**. You can select existing tag keys and values from the drop-down lists or enter new ones. If you want to use the same tag to identify multiple cloud resources for better resource grouping, we recommend that you predefine tags in Tag Management Service (TMS). For details, see **Predefined Tags**. Each cluster can have a maximum of 20 tags. If your organization has configured tag policies for CSS, add tags to clusters based on these policies. Tags that do not comply with predefined tag policies will cause a cluster creation failure. Contact the administrator to learn about the tag policies.

Submit

1. In the **Configuration Summary** panel, review the cluster configuration details.

   Any mandatory fields that are missing are displayed in red, and you have to set them in the configuration area.
2. Click **Create Now**.

   Cluster creation time depends on the number of nodes. Clusters with a large number of nodes typically take longer to create.
3. Return to the cluster list and check the newly created cluster.

   If the cluster is created successfully, **Cluster Status** changes to **Available**.

4.4 Creating a Cluster

You can create a Logstash cluster in either of the following ways:

- Method 1: Create a Logstash cluster on the CSS management console.
- Method 2: Create a Logstash cluster using CSS APIs. For details, see **Creating a Cluster**.

Prerequisites

You have planned the Logstash clusters that need to be created by following the instructions in **Cluster Planning**.

Creating a Cluster

1. Log in to the **CSS management console**.
2. On the **Dashboard** page, click **Create Cluster** in the upper right corner. The **Create Cluster** page is displayed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Alternatively, choose **Clusters** > **Logstash** in the navigation tree on the left. Click **Create Cluster** in the upper right corner.

3. On the **Basic Configuration** page, configure basic information and resources for the Logstash cluster.

Table 4-10 Basic configuration of the Logstash cluster

Parameter	Description
Region	Select the region where the cluster is located. Resources in different regions cannot communicate with each other over an internal network. For lower network latency and quicker resource access, select the nearest region.
AZ	Select AZs associated with the cluster region. A maximum of two AZs can be configured. For details about the use of multiple AZs, see **Suggestions on Multi-AZ Deployment**.
Type	Select **Logstash**.
Version	Select a cluster version from the drop-down list.
Name	User-defined cluster name. The cluster name must start with a letter and can contain 4 to 32 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
Cluster Description	Add a description for the cluster for easy recognition. The value can contain 0 to 128 characters.
Nodes	Number of nodes in the cluster, in the range 1 to 100.
CPU Architecture	**x86**. The supported types depend on the actual regional environment.
Node Specifications	Specifications of nodes in the cluster. You can select a node flavor based on your needs. Each cluster supports only one specification. For details, see **ECS Types**.
Node Storage Type	Supported node storage types: Common I/O, High I/O, Ultra-high I/O. **NOTE** If the type of storage in use is not supported, the storage type is not displayed.
Node Storage Capacity	It can only be set to 40 GB for a Logstash cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Enterprise Project	When creating a CSS cluster, you can bind an enterprise project to the cluster if you have enabled the enterprise project function. Select an enterprise project from the **Enterprise Project** drop-down list, or click **View Enterprise Project** to go to the **Enterprise Project Management Service** page and check existing enterprise projects.

4. Click **Next: Network**.
5. On the **Network** page, configure the network and security group settings for the Logstash cluster.

Table 4-11 Parameter description

Parameter	Description
VPC	Specify a VPC to isolate the cluster's network. Click **View VPC** to go to the VPC management console and check the created VPCs. If no VPC is available, contact the CSS administrator to create a new VPC. For details, see **Creating a VPC with a Subnet**. **NOTE** The 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, improving network security. Select a subnet needed by the cluster in the current VPC. If no existing subnets meet your requirements, click **Create Subnet** to create a new subnet. For details, see **Creating a VPC and Subnet**.
Security Group	A security group serves as a virtual firewall that provides access control policies for clusters. Select a security group for the cluster. Click **View Security Group** to go to the security group list, where you can view details about security groups. **NOTE** To enable cluster access, ensure that port 9600 is allowed by the security group.

6. Click **Next: Confirm Configuration**. Confirm the settings, and click **Create Now** to create the cluster.
7. Return to the cluster list and check the newly created cluster. If the cluster is created successfully, **Cluster Status** changes to **Available**. Cluster creation

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

time depends on the number of nodes. Clusters with a large number of nodes typically take longer to create.

If cluster creation fails, try creating the cluster again by rectifying the errors returned.

4.5 Network and Security Settings

4.5.1 Configuring Public Network Access

CSS Logstash is a data ingestion service that ingests data from designated data sources on the public network and writes it to designated storage systems. The data is subsequently used for purposes such as big data analytics and log management. An NAT gateway and return routes need to be configured to ensure proper data ingestion, storage, and processing across networks.

- NAT gateway: Configure an NAT gateway for the Logstash cluster to enable the cluster to initiate connection requests and connect to Internet data sources.
- Return routes: They direct Logstash nodes to connect to the IP addresses of data sources on the public network and back, ensuring proper data ingestion and transmission.

Constraints

- An NAT gateway and EIP are required to enable public network access for a Logstash cluster. They must be purchased additionally and be in the same region as the Logstash cluster.
- The SNAT rule added to the NAT gateway applies only to resources in the specified subnet. Make sure you use the subnet of the Logstash cluster.

Configuring Public Network Access for a Logstash Cluster

Step 1 Obtain the VPC and subnet of the Logstash cluster for configuring an NAT gateway later.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Overview** tab. In the **Network Information** area, record the cluster's VPC and subnet information.

Step 2 Create a public NAT gateway for the Logstash cluster. For details, see **Creating a Public NAT Gateway**.

Table 4-12 describes the key parameters. Set other parameters based on service requirements.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 4-12 Configuring a public NAT gateway

Parameter	Description
Region	Use the region of the Logstash cluster.
Specifications	Select a value based on the needed bandwidth. The maximum number of connections supported by SNAT varies depending on the specifications you choose.
VPC	Select the VPC recorded in **Step 1**.
Subnet	Select the subnet recorded in **Step 1**.

Step 3 Add SNAT rules on the NAT gateway to allow the Logstash cluster to access the public network through a shared EIP on the gateway. For details, see **Adding an SNAT Rule**.

Table 4-13 describes the key parameters. Set other parameters based on service requirements.

Table 4-13 Adding an SNAT rule

Parameter	Description
Scenario	Select **VPC**.
CIDR Block	Select **Existing**, and then select the VPC recorded in **Step 1**.
Public IP Address Type	Select **EIP**, and then select an IP address that meets your requirements. If no EIP is available, click **View EIP** to go to the EIP console and . You are advised to configure the EIP as follows: • **Billing Mode**: **Pay-per-use** • **Billed By**: **Traffic**

----End

Configuring Routes for a Logstash Cluster

Configure a return route for the Logstash cluster to make sure the Logstash nodes can connect to the IP addresses of data sources on the public network. The return route must cover all these IP addresses.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, click **Add Route** under **Cluster Route** in the **Network Information** area.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

5. In the displayed dialog box, configure the route information.

Table 4-14 Adding a route

Parameter	Description
IP Address	Enter the IP address (or a CIDR block) of a data source, for example, **10.10.1.0**. If the IP addresses of data sources are scattered, configure multiple routes for them.
Subnet Mask	Enter the subnet mask used with the IP address. The subnet mask must align with the network part of the IP address. That is: • The network bits (the 1s) of the subnet mask must encompass those of the IP address. • The host bits (the 0s) of the subnet mask must accommodate the IP address's host bits. The subnet mask should not cover too wide a range, avoiding unnecessary IP address exposure or waste of the address space. 255.255.255.0 (that is, /24) is recommended, which applies to most internal networking scenarios.

6. Click **OK** to add the route.

   After the route is added, click **View Route** to check the route information. You can click **Add Route** to add another route; or click **Delete Route** to delete one—In the **Delete Route** dialog box, select a route, enter **DELETE** manually, and then click **OK**.

Testing Connectivity

Test the connectivity between the Logstash cluster and the data sources.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Select the **Configuration Center** tab, and click **Test Connectivity**.
5. In the **Test Connectivity** dialog box, enter the IP address and port number of a data source, and click **Test**.

Figure 4-3 Testing connectivity



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

If **Available** is displayed, the network between the Logstash cluster and the data source is connected.

NOTE

You can test a maximum of 10 IP addresses at a time. Click **Add** to add multiple IP addresses and click **Test** at the bottom to test connectivity to all these IP addresses at the same time.

4.5.2 Configuring Routes for a Cluster

By default, a CSS Logstash cluster (server) cannot access data sources that are located in a different VPC—it cannot pull data from or push data to the data sources. To enable it, you need to configure routes for the Logstash cluster that point to the data sources.

1. Establish a VPC peering connection to connect two VPCs.
2. On CSS, configure routes for the Logstash cluster that point to the data sources, enabling the cluster to pull data from or push data to the data sources.
3. Verify network connectivity between the data sources and Logstash.

Configuring routes to enable cluster communication offers the following advantages:

- Improved cross-VPC communication performance: A route table specifies the traffic forwarding paths, so that traffic does not have to pass through the public network.
- Secure communication: Private networks can communicate through VPC peering connections, which is more secure than transmitting data over the public network.
- Flexible access control policies: You can customize access permissions for specific subnets.
- Support for a hybrid cloud architecture: Services can communicate across VPCs in different environments.

Constraints

Adding cluster routes will alter network connections, possibly causing disruptions in the network. Exercise caution.

Prerequisites

- The Logstash cluster is available; and the security group used by the cluster allows outbound traffic for all protocols, and inbound traffic for the ICMP protocol as well as the TCP ports associated with the destination addresses.
- The data sources are available, and the security group used by the server of the data sources allows inbound traffic on the TCP ports associated with the destination addresses. For example, with an Elasticsearch cluster, port 9200 must be allowed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Establishing a VPC Peering Connection

Before configuring routes for a Logstash cluster, establish a connection between the cluster's VPC and that of the data sources, that is, a VPC peering connection.

- If the Logstash cluster and the data sources belong to the same account, see .
- If the Logstash cluster and the data sources belong to different accounts, see .

Set the peer VPC to the VPC where the Logstash cluster is located and the local VPC to the VPC where the data sources are located.

When adding routes for a VPC peering connection, configure the destination addresses based on service requirements.

If the Logstash cluster and the data sources share the same security group, there is no need to configure security group rules for the instances in the VPC at either end of the VPC peering connection.

Configuring Routes for a Logstash Cluster

On CSS, configure routes for the Logstash cluster that point to the data sources, enabling the cluster to pull data from or push data to the data sources.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, click **Add Route** under **Cluster Route** in the **Network Information** area.
5. In the displayed dialog box, configure the route information.

Table 4-15 Adding a route

Parameter	Description
IP Address	Enter the IP address (or a CIDR block) of a data source, for example, **10.10.1.0**. If the IP addresses of data sources are scattered, configure multiple routes for them.
Subnet Mask	Enter the subnet mask used with the IP address. The subnet mask must align with the network part of the IP address. That is: • The network bits (the 1s) of the subnet mask must encompass those of the IP address. • The host bits (the 0s) of the subnet mask must accommodate the IP address's host bits. The subnet mask should not cover too wide a range, avoiding unnecessary IP address exposure or waste of the address space. 255.255.255.0 (that is, /24) is recommended, which applies to most internal networking scenarios.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

6. Click **OK** to add the route.

   After the route is added, click **View Route** to check the route information. You can click **Add Route** to add another route; or click **Delete Route** to delete one—In the **Delete Route** dialog box, select a route, enter **DELETE** manually, and then click **OK**.

Testing Connectivity

Test the connectivity between the Logstash cluster and the data sources.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Select the **Configuration Center** tab, and click **Test Connectivity**.
5. In the **Test Connectivity** dialog box, enter the IP address and port number of a data source, and click **Test**.

   Figure 4-4 Testing connectivity



   If **Available** is displayed, the network between the Logstash cluster and the data source is connected.

   NOTE

   You can test a maximum of 10 IP addresses at a time. Click **Add** to add multiple IP addresses and click **Test** at the bottom to test connectivity to all these IP addresses at the same time.

4.5.3 Managing CA Certificates

When a Logstash cluster needs to connect to external data sources (such as Elasticsearch or databases) over HTTPS, it must have CA certificates that can verify the identity of the servers. Otherwise, the connection is rejected because the certificates are not trusted. For connections to Elasticsearch/OpenSearch clusters provided by CSS, CSS provides default CA certificates out of the box. When Logstash needs to connect to third-party data sources, such as self-managed Elasticsearch, MySQL, or Kafka, you must manually upload the required custom CA certificates to verify server identity and secure data transmission.

How the Feature Works

CA certificates are the foundation of trust for TLS/SSL connections. During a TLS handshake, the server sends its certificate to the client. The client then uses its

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

trusted CA certificate to verify the validity of the server's certificate. Once the server's identity is verified, an encrypted connection is established. **Figure 4-5** illustrates the TLS handshake process that relies on CA certificates.

Figure 4-5 TLS handshake process

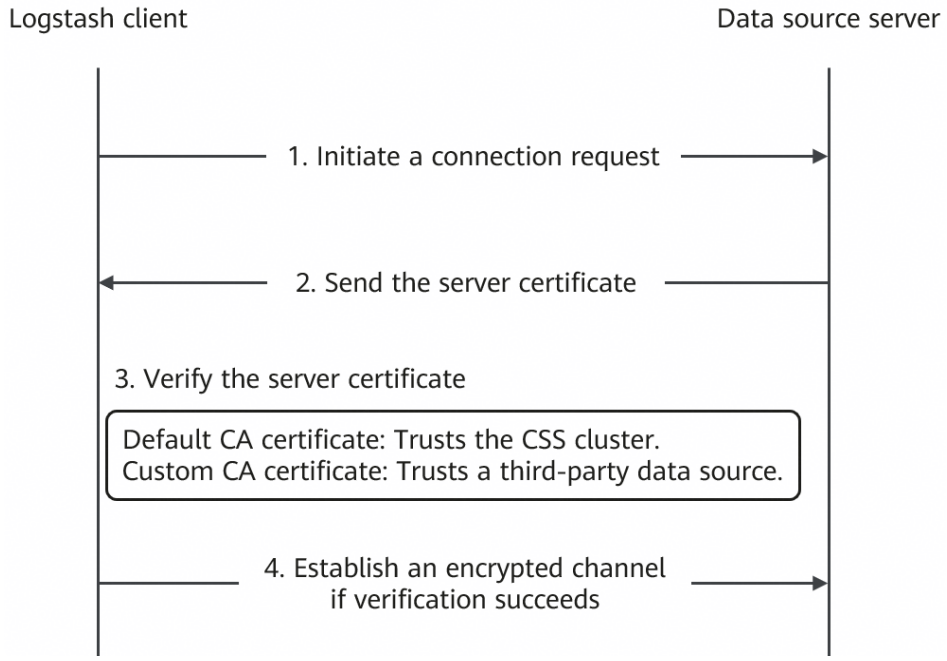


Table 4-16 compares the two types of CA certificates supported by CSS Logstash.

Table 4-16 CA certificate comparison

Comparison Item	Default CA Certificate	Custom CA Certificate
Source	Provided by CSS	Prepared and uploaded by users
Applicable scenario	Connecting to CSS Elasticsearch/OpenSearch clusters (with security mode and HTTPS enabled)	Connecting to third-party data sources, such as self-managed Elasticsearch, MySQL, and Kafka
Configuration effort	No configuration required. Ready for use out of the box.	Certificate files must be prepared and uploaded to OBS in advance.
Quantity limit	Fixed and cannot be modified.	A maximum of 50 certificates can be uploaded per cluster.
Usage	Directly reference the certificate via its path.	Upload the certificate to OBS and then to the Logstash cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Requirements for Custom Certificate Files

Prepare custom certificate files according to the following requirements. Certificate upload will fail if these requirements are not met.

Table 4-17 Requirements for custom certificate files

Constraint	Requirement
Naming rules	The name must start with a letter and can contain only letters, digits, hyphens (-), underscores (_), and periods (.). The length must be 4 to 32 characters.
File formats	.cer, .crt, .rsa, .jks, .pem, .p10, .pfx, .p12, .csr, .der, .keystore
File size	Each file must not exceed 1 MB.
Quantity limit	A maximum of 50 custom certificates can be uploaded to a single cluster.

Viewing Default CA Certificates

Obtain the name and path of the default CA certificate provided by CSS, and reference the certificate in your Logstash configuration file to establish trust with CSS Elasticsearch/OpenSearch clusters.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Certificate Management** > **Default Certificates**.
5. On the **Default Certificates** page, record the certificate name and path.

   Usage: In your Logstash configuration file, specify <*certificate path*><*certificate name*> in the **cacert** parameter. (Example: /opt/logstash/extend/certs)

Uploading a Custom CA Certificate

Import custom CA certificate files into the certificate management system of your Logstash cluster to generate certificate paths that can be referenced in Logstash configuration files. This enables Logstash to establish trust when connecting to third-party data sources.

1. Prepare and upload a custom certificate to an OBS bucket.
   a. Prepare a certificate file that meets the requirements described in **Requirements for Custom Certificate Files**.
   b. Upload the certificate file to an OBS bucket. For details, see **Uploading an Object**.
      - The OBS bucket must be in the same region as the Logstash cluster, and its storage class must be **Standard**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- If SSE-KMS is enabled for the OBS bucket, the account used for operations must have the following KMS permissions. (Choose **Permissions > Policies/Roles** on the IAM console to check your permissions.)

```
"kms:cmk:create",
"kms:dek:create",
"kms:cmk:get",
"kms:dek:decrypt",
"kms:cmk:list"
```

2. Upload the certificate to the Logstash cluster.
   a. Log in to the **CSS management console**.
   b. In the navigation pane on the left, choose **Clusters > Logstash**.
   c. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
   d. Choose **Certificate Management** > **Custom Certificates**.
   e. On the **Custom Certificates** page, click **Upload Certificate**. In the displayed dialog box, configure the required parameters.

      Table 4-18 Uploading a certificate

Parameter	Description
OBS Bucket	Select the OBS bucket where the certificate file has been uploaded. An IAM user must be granted the following OBS permissions to view available OBS buckets: obs:bucket:getBucketStoragePolicy obs:bucket:listAllMyBuckets
Certificate Object	Click **Select**. In the displayed dialog box, select the target certificate file, and click **OK**.

   f. Click **OK** to upload the custom certificate.
   g. After the certificate is uploaded, it is displayed in the custom certificate list. Obtain the certificate name and path.

      Usage: In your Logstash configuration file, specify <*certificate path*><*certificate name*> in the **cacert** parameter. (Example: /opt/logstash/extend/certs)

3. (Optional) Delete expired or unused certificates.

WARNING

Before deleting a certificate, ensure that the certificate is no longer referenced by any pipeline. Otherwise, related pipelines will fail to connect to data sources after the certificate is deleted.

   a. In the custom certificate list, select the certificate to be deleted and click **Delete** in the **Operation** column.
   b. In the displayed dialog box, click **OK** to delete the certificate.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

FAQ: How Do I Select a Custom CA Certificate Format?

Common options:

- **.pem**: A text-based format that is the most widely used and recommended for Linux environments.
- **.crt**/**.cer**: Certificate files that are commonly used together with **.key** files.
- **.jks**/**.keystore**: Java keystores used for Java-based connections.
- **.pfx**/**.p12**: PKCS#12 format files that contain private keys and certificate chains.

If you are not sure, **.pem** is recommended.

4.6 Configuring Migration Tasks

4.6.1 Creating a Configuration File

Enterprise data integration often involves complex requirements for cleaning and aggregating heterogeneous data from multiple sources, such as databases, logs, and message queues. Writing Logstash configuration files manually can be challenging due to complex syntax and debugging. It can also introduce security risks, for example, when database passwords or other credentials are stored in plaintext in configuration files. CSS Logstash addresses this challenges with a visual configuration center that enables you to build secure, efficient, and easy-to-manage data processing pipelines. The configuration center allows you to quickly generate ETL logic using a custom template and convenient developer-assisting features. A built-in sensitive information masking feature helps protect user credentials and other confidential data. You can also fine-tune pipeline parameters such as concurrency and buffering to meet diverse performance and throughput requirements.

How the Feature Works

A Logstash configuration file defines a pipeline consisting of three stages:

1. Input: Data is ingested from one or more sources.
2. Filter: Data is parsed, cleaned, transformed, and enriched. This is typically the most CPU-intensive stage.
3. Output: Data is sent to a destination, such as Elasticsearch.

For more information about them, see the official document **Logstash Plugins**.

Constraints

- A maximum of 50 configuration files can be created for a Logstash cluster.
- A maximum of five configuration files can be in the verifying state at the same time.
- The size of a single configuration file cannot exceed 100 KB.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Prerequisites

- You have obtained the connection information for both the data source (such as Elasticsearch, MySQL, and Kafka) and destination (such as Elasticsearch), including IP addresses, port numbers, accounts, and passwords.
- The Logstash cluster's VPC and security group rules allow communication with both the data source and destination.

Creating a Configuration File

1. Go to the **Configuration Center** page.
   a. Log in to the **CSS management console**.
   b. In the navigation pane on the left, choose **Clusters > Logstash**.
   c. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
   d. Click the **Configuration Center** tab.
2. On the Configuration Center page, click **Create** in the upper-right corner. The **Create Configuration File** page is displayed.
3. Edit the configuration file to define the data collection and processing workflow.

Table 4-19 Parameters for creating a configuration file

Parameter	Description
Name	User-defined configuration file name. It can contain only letters, digits, hyphens (-), and underscores (_), and must start with a letter. The minimum length is 4 characters. You are advised to include a description of the data's purpose in the configuration file name to facilitate management.
Configuration File Content	Define the input, filter, and output logic. To use a configuration file template, expand **System Templates** or **Custom Templates**, select a template based on its description, and click **Apply** in the **Operation** column to copy the content of the template. • System templates are preset configuration file templates of CSS. They cover various types of input sources, such as Redis, Elasticsearch, OpenSearch, JDBC, Kafka, and Beats. For details, see **Configuration File Templates**. • Custom templates are templates added by users. The size of each configuration file cannot exceed 100 KB.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Hidden Content	Specify strings to hide in the configuration file, such as passwords and access keys. Press **Enter** after each string. These strings will be displayed as ******* in the configuration file. You can configure a maximum of 20 strings to hide, each with a maximum length of 512 bytes.
Description	Add a description for the configuration file for easy identification. The value can contain 0 to 128 characters.

4. Click **Next** to configure pipeline parameters.
5. Configure pipeline runtime parameters based on the data volume and reliability requirements.

Table 4-20 Pipeline configuration parameters

Parameter	Category	Default Value	Description
pipeline.workers	Concurrency control	The number of CPU cores	Number of worker threads that execute the Filters and Outputs stages of the pipeline in parallel. • For an I/O-intensive pipeline (such as simple forwarding), set this parameter to the number of CPU cores. • For a CPU-intensive pipeline (such as complex regular expression parsing), set this parameter to the number of CPU cores or a slightly lower value. Do not exceed the number of CPU cores on a single node. Otherwise, increased context switching overhead will degrade performance.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Category	Default Value	Description
pipeline.batch.size	Throughput	125	Maximum number of events that a worker thread collects from inputs before attempting to execute its filters and outputs. When processing a large number of small documents, increasing this value (for example, 500 to 3000) can significantly enhance throughput. However, due to limited JVM heap memory, setting this value too high can lead to out of memory (OOM) errors.
pipeline.batch.delay	Latency control	50	Maximum amount of time in milliseconds that a pipeline worker waits for each new event while its current batch is not yet full. Unit: ms Reduce the value of this parameter when real-time performance takes priority.
queue.type	Data reliability	memory	Configures an internal queue for event buffering. • **memory**: Uses a traditional memory-based queue, which delivers high performance but risks data loss in case of process failures. • **persisted**: Uses a disk-based persistent queue, which prevents data loss through persistent storage and supports resumable data transfer. When selecting this mode, monitor storage usage carefully to avoid exhaustion.
queue.checkpoint.writes	Checkpoint	1024 (recommended)	Maximum number of written events before a forced checkpoint. This parameter needs to be set only when **queue.type** is set to **persisted**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Category	Default Value	Description
queue.max_bytes	Disk-based queue	1024 (recommended)	Total capacity of a persistent queue. • This parameter needs to be set only when **queue.type** is set to **persisted**. • Ensure sufficient disk space. Logstash will stop receiving new data when disk is used up. Unit: MB

6. After the configuration is complete, click **Create**.
7. Return to the configuration file list. If the **Status** of the new configuration file changes to **Available**, it is created successfully.

Managing Configuration Files

You can edit existing configuration files, set a configuration file as a custom template, and delete configuration files.

Table 4-21 Managing configuration files

Operation	Constraints	Operation Guide
Modifying a configuration file	A configuration file that has an ongoing pipeline task cannot be modified.	In the configuration file list, find the row that contains the configuration file you want to edit, and click **Edit** in the **Operation** column. Modify the file to adapt to new requirements or correct errors.
Setting a configuration file as a template	N/A	In the configuration file list, click **Add to Custom Template** in the **Operation** column. In the displayed dialog box, set the template name, description, and configuration file content, and click **OK**. It can then be used as a template for configuring configuration files subsequently.
Backing up configuration files	N/A	Click  in the upper right corner above the configuration file list to export all configuration files for local backup.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Operation	Constraints	Operation Guide
Deleting a configuration file	• A configuration file that has an ongoing pipeline task cannot be deleted. • The deletion cannot be undone.	In the configuration file list, find the row that contains the configuration file you want to delete, and click **Delete** in the **Operation** column. In the displayed dialog box, type in **DELETE** and then click **OK** to confirm the deletion.

4.6.2 Starting Migration Tasks

In large-scale data processing and real-time log ingestion scenarios, developers often need to pull data from various data sources, such as Kafka and MySQL, to Elasticsearch or OpenSearch. Key challenges in such efforts include network connectivity, task interruptions, and the complexity of multi-pipeline management. CSS provides a web-based configuration center for Logstash clusters, where you can test network connectivity between Logstash and both the data source and destination in one click, as well as start or hot-start pipelines. To ensure high availability and stability in complex scenarios, CSS also deploys a Keepalive mechanism supported by a daemon process.

- Pipeline: A core Logstash component for data processing, comprising three stages: input, filter, and output.
- Keepalive: A high availability mechanism. When enabled, a daemon process monitors Logstash; and if Logstash fails, the daemon process automatically attempts to restore it.

Comparing Pipeline Start Options

Table 4-22 Start vs. Hot Start

Option	Start	Hot Start
How It Works	When the cluster has no running pipelines, the system initializes the runtime environment and loads the configuration.	When the cluster already has running pipelines, a new pipeline is dynamically loaded without restarting the entire Logstash process, ensuring zero service interruptions.
Batch Processing	Up to 50 configuration files can be started at once.	Only one configuration file can be started at a time.
Keepalive	You can choose to enable or disable Keepalive.	The Keepalive setting is inherited from existing pipelines in the cluster. It cannot be modified.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

When to Use	The cluster has no **Running** pipelines.	The cluster already has **Running** pipelines. You use this option to add new pipelines without interrupting existing pipelines.
Constraint	N/A	• Configuration files that use the **stdin** plugin cannot use the **Hot Start** option. • If a hot start fails and the Logstash process becomes abnormal, the system will automatically try to restore it through a restart. Use it with caution.

Constraints

There can be up to 50 **Running** configuration files in the pipeline list of a Logstash cluster.

Preparations: Testing Connectivity

Before starting a migration task, test the connectivity between Logstash and both the data source and destination. This helps prevent task failures caused by connectivity issues.

How a connectivity test works:

- IP layer test: Run the **fping** command to check whether the target IP address is reachable.
- Transport layer test: Run the **telnet** command to check whether the TCP port associated with the target address is reachable.

Prerequisites for the connectivity test

Depending on the locations of the Logstash cluster, data source, and destination, the following conditions must be met:

- If they are in the same VPC: The security group used by the Logstash cluster allows outbound traffic for all protocols, and inbound traffic for the ICMP protocol as well as the TCP ports associated with the peer addresses. For details, see **Adding a Security Group Rule**.
- If they are in different VPCs: The security group used by the Logstash cluster allows outbound traffic for all protocols, and inbound traffic for the ICMP protocol as well as the TCP ports associated with the peer addresses. In addition, you must configure a route between the Logstash cluster and the peer VPCs. For details, see **Configuring Routes for a Logstash Cluster**.
- If they are in different LANs (they must communicate across the public network): Configure a public IP address for the Logstash cluster to ensure it can be accessed from the Internet. For details, see **Configuring Public Network Access for a Logstash Cluster**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

1. Go to the **Configuration Center** page.
   a. Log in to the **CSS management console**.
   b. In the navigation pane on the left, choose **Clusters > Logstash**.
   c. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
   d. Click the **Configuration Center** tab.
2. On the **Configuration Center** page, click **Test Connectivity**.
3. In the **Test Connectivity** dialog box, enter the IP addresses and port numbers of the data source and destination, and click **Test**.

   You can test a maximum of 10 IP addresses at a time. You can click **Add** to add more IP addresses and click **Test** at the bottom to test connectivity to multiple IP addresses at a time.

   If **Available** is returned, the network is connected. If **Unavailable** is returned, troubleshoot by following .

Starting a Migration Task

Start a Logstash pipeline to migrate data from the source to the destination.

1. Check whether there are pipelines whose status is **Running** in the pipeline list.
   – Yes: Perform **3** to hot-start a configuration file.
   – No: Perform **2** to start a configuration file.
2. Start a configuration file.
   a. In the configuration file list, select 1 to 50 configuration files and click **Start Logstash** above.
   b. In the displayed dialog box, choose whether to select **Keepalive**.

      Select Keepalive for long-term data streams in production environments. If the Logstash service stops due to an unexpected error, the system will automatically try to restart it.

      Deselect Keepalive for one-time offline data migration tasks. Note that if the source has no data, enabling Keepalive may cause the pipeline to restart repeatedly, which may lead to errors.
   c. Click **OK** to activate Logstash pipelines.
3. Hot-start a configuration file.
   a. In the configuration file list, select one configuration file, and click **Hot Start** above.
   b. In the displayed dialog box, confirm the Keepalive setting. The Keepalive setting is inherited from existing pipelines in the cluster. It cannot be modified.
   c. Click **OK** to activate the Logstash pipeline.
4. Verify the result. When the status of the new pipeline changes to **Running** in the pipeline list, the data migration task has started and is running properly. The **Events** column will be dynamically updated to show the number of processed documents.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Monitoring Pipelines

After starting data migration pipelines, you can monitor them in the following ways:

- Check their metrics.

  In the pipeline list, locate the target pipeline, and click **Metric Monitoring** in the **Operation** column to go to the Cloud Eye console, where you can check the metrics to evaluate its status and performance.

  When the **Events** data of a pipeline changes dynamically, the monitoring data changes accordingly. When a pipeline is being started or stopped, or the **Events** data is stable, the monitoring data remains unchanged.

  For details about the metrics supported, see **Logstash Pipeline Monitoring Metrics**. For how to configure alarms, see **Using Cloud Eye to Monitor Clusters**.
- Check run logs.

  Above the pipeline list, click **Run Logs** to check the logs of the Logstash process, which you can use to troubleshoot data parsing errors.
- Check operation records.

  Above the pipeline list, click **View** to check the start and stop records of the pipelines. This helps track historical operations and facilitate troubleshooting.

4.6.3 Stopping Migration Tasks

When using Logstash for large-scale data migration or log ingestion, you may need to suspend or terminate pipelines to allow for policy adjustment or configuration optimization, or upon task completion. However, inappropriately terminating pipelines may cause ingestion checkpoint loss or data inconsistency between the source and destination. CSS provides two methods for stopping Logstash pipelines: Stop All and Hot Stop. Each method is designed for specific task scenarios. By selecting the appropriate method, you can ensure data migration and log ingestion tasks are stopped safely.

Comparing Pipeline Stop Options

Table 4-23 Stop All vs. Hot Stop

Stop Option	Stop All	Hot Stop
Pipeline Status	The pipeline status changes from **Running** to **Stopped**.	The pipeline is removed.
How It Works	A global signal is sent to the Logstash process to block the pipeline execution logic without deregistering the configuration metadata.	A dynamic unloading operation is performed on the selected pipeline to disassociate the configuration file. The Logstash process will no longer load this pipeline.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Data integrity	Stopping a pipeline may cause brief latency for data in transit or trigger an offset reset. Ensure that the source data maintains traceability, for example, via incremental change-tracking columns in the source database. Additionally, it will immediately interrupt data migration. Exercise caution.	
When to Use	Use this option when you are preparing your cluster for a comprehensive maintenance, or when you need to release resources during peak hours. With this option, you can suspend pipelines while retaining their configuration to allow for a one-click restoration later.	Use this option when you want to permanently remove a pipeline that is no longer needed. This releases resources without interrupting other pipelines.

Prerequisites

There are **Running** pipeline tasks in the configuration center of the Logstash cluster.

Stop All

Suspend all running pipelines in the cluster in one click.

1. Go to the **Configuration Center** page.
   a. Log in to the **CSS management console**.
   b. In the navigation pane on the left, choose **Clusters > Logstash**.
   c. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
   d. Click the **Configuration Center** tab.
2. Above the pipeline list, click **Stop All**.
3. In the displayed dialog box, click **OK**.
4. Verify the result. Check the **Status** column in the pipeline list and confirm that all pipelines are now **Stopped**.

Hot Stop

Remove a specific pipeline without interrupting other pipelines.

1. Go to the **Configuration Center** page.
   a. Log in to the **CSS management console**.
   b. In the navigation pane on the left, choose **Clusters > Logstash**.
   c. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
   d. Click the **Configuration Center** tab.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. In the pipeline list, select the target pipeline, and click **Hot Stop** above.
3. In the displayed dialog box, click **OK**.
4. Verify the result. Confirm that the selected pipeline is no longer in the pipeline list.

4.6.4 Configuration File Templates

When building a real-time log processing or big data migration pipeline, writing Logstash configuration files manually is often time-consuming and error-prone due to complex syntax. Configuration errors may even lead to data loss. To address this, CSS provides prebuilt Logstash configuration templates for a variety of mainstream data sources, such as Redis, Elasticsearch, MySQL (JDBC), and Kafka. Using these verified templates, you can quickly set up automated data transmission channels from data sources to CSS Elasticsearch or OpenSearch clusters by configuring only the required connection information. This significantly reduces operational overhead and helps ensure the reliability of data ingestion pipelines.

Introduction to the Templates

Table 4-24 Logstash configuration file templates

System Template Name	Description	Details
redis	Ingests data from a Redis database into an Elasticsearch or OpenSearch cluster.	**Redis Template**
elasticsearch	Ingests data from an Elasticsearch or OpenSearch cluster into another Elasticsearch or OpenSearch cluster.	**Elasticsearch Template**
jdbc	Ingests data from a MySQL or MariaDB database via JDBC into an Elasticsearch or OpenSearch cluster.	**JDBC Template**
kafka	Ingests data from Kafka into an Elasticsearch or OpenSearch cluster.	**Kafka Template**
beats	Ingests data from Beats into an Elasticsearch or OpenSearch cluster.	**Beats Template**

Redis Template

Ingest data from a Redis database into an Elasticsearch or OpenSearch cluster.

```
# logstash-input-redis
input {
    redis {
        # Redis data type
        data_type => "pattern_channel"
        # Redis list or channel name
        key => "lgs-*"
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        host => "xxx.xxx.xxx.xxx"
        port => 6379
    }
}

filter {
    # Removes some metadata fields automatically added by Logstash
    mutate {
        remove_field => ["@timestamp", "@version"]
    }
}

# logstash-output-elasticsearch
output {
    elasticsearch {
        # Destination cluster node addresses. No need to include the protocol.
        hosts => ["xxx.xxx.xxx.xxx:9200", "xxx.xxx.xxx.xxx:9200"]
        # Name of the index that events are written into.
        index => "xxx"
        # Mandatory fields for a security-mode cluster. (Delete them for a cluster with the security mode
disabled.)
        # user => "xxx"           # Username for accessing the cluster.
        # password => "xxx"       # Password corresponding to the username.
        # If SSL is enabled for the destination cluster, additionally configure the following information:
        # ssl => true
        # cacert => "/opt/logstash/extend/certs"    # Path of the CA certificate used to verify the destination
cluster.
        # ssl_certificate_verification => false     # Whether to enable SSL certificate verification for the
destination cluster.
    }
}
```

Table 4-25 Configuration items

Configuration Item	Mandatory	Description
logstash-input-redis		
data_type	Yes	Redis data types. The value can be: • **list**: Implements a message queue by consuming data from a Redis list using the blocking pop command (BLPOP). • **channel**: Subscribes to a specific channel name using the SUBSCRIBE command. • **pattern_channel**: Subscribes to channels matching a pattern using the PSUBSCRIBE command.
key	Yes	Name of the Redis list or channel. • Single name: Enter the channel name (for example, **log**). • Wildcard: Use the wildcard (*) to match multiple channels. For example, **log-*** indicates all channels whose name starts with **log-**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuration Item	Mandatory	Description
host	Yes	Redis server address. Enter an IP address or domain name.
port	No	Listening port of the Redis server.
logstash-output-elasticsearch		
hosts	Yes	Destination cluster node addresses. You can configure multiple IP addresses. Value format: ["<Node IP address 1>:<Port number>", "<Node IP address 2>:<Port number>"]
index	Yes	Name of the index to which events are written. • Single index: Enter the index name, for example, **my_index**. • Multiple indexes: Use dynamic naming (based on event fields) or configure multiple conditional output blocks to route events to different indexes.
user	No	Username for accessing the destination cluster. Mandatory for a security-mode cluster.
password	No	Password for accessing the destination cluster. Mandatory for a security-mode cluster.
ssl	No	Whether SSL is enabled for the destination cluster. The value can be: • **true**: Uses HTTPS to transmit data. • **false**: Uses HTTP to transmit data.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuration Item	Mandatory	Description
cacert	No	Path of the CA certificate used to verify the destination cluster. Value format: <certificate path><certificate name>, for example, /opt/logstash/extend/certs. • If the destination is a CSS Elasticsearch or OpenSearch cluster, the **certificate name** and **certificate path** of the default CA certificate will be obtained. For details, see **Viewing Default CA Certificates**. • If the destination is a self-managed or third-party Elasticsearch or OpenSearch cluster, upload the destination cluster's security certificate to Logstash and obtain the **certificate name** and **certificate path**. For details, see **Uploading a Custom CA Certificate**.
ssl_certificate_verification	No	Whether SSL certificate verification is enabled for the destination cluster. The value can be: • **true** (default): Uses an SSL certificate to verify the destination cluster. • **false**: Ignores SSL certificate verification.

For more information, see **Redis input plugin** and **Elasticsearch output plugin** from the official Logstash documentation.

Elasticsearch Template

Ingest data from an Elasticsearch or OpenSearch cluster into another Elasticsearch or OpenSearch cluster.

```
# logstash-input-elasticsearch
input {
   elasticsearch {
      # Source cluster node addresses. No need to include the protocol.
      hosts => ["xxx.xxx.xxx.xxx:9200", "xxx.xxx.xxx.xxx:9200"]
      # Name of the source index to be migrated
      index => "xxx,xxx"
      docinfo => true
      # Mandatory fields for a security-mode cluster. (Delete them for a cluster with the security mode disabled.)
      # user => "xxx"          # Username for accessing the cluster.
      # password => "xxx"      # Password corresponding to the username.
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        # If SSL is enabled for the source cluster, additionally configure the following information:
        # ssl => true
        # ca_file => "/opt/logstash/extend/certs"     # Path of the CA certificate used to verify the source
cluster.
    }
}

filter {
    # Removes some metadata fields automatically added by Logstash
    mutate {
        remove_field => ["@timestamp", "@version"]
    }
}

# logstash-output-elasticsearch
output {
    elasticsearch {
        # Destination cluster node addresses. No need to include the protocol.
        hosts => ["xxx.xxx.xxx.xxx:9200", "xxx.xxx.xxx.xxx:9200"]
        # Target index for writing events. The following configuration retains the source index name,
document type, and document IDs.
        index => "%{[@metadata][_index]}"
        document_type => "%{[@metadata][_type]}"
        document_id => "%{[@metadata][_id]}"
        # Mandatory fields for a security-mode cluster. (Delete them for a cluster with the security mode
disabled.)
        # user => "xxx"           # Username for accessing the cluster.
        # password => "xxx"       # Password corresponding to the username.
        # If SSL is enabled for the destination cluster, additionally configure the following information:
        # ssl => true
        # cacert => "/opt/logstash/extend/certs"      # Path of the CA certificate used to verify the destination
cluster.
        # ssl_certificate_verification => false        # Whether to enable SSL certificate verification for the
destination cluster.
    }
}
```

Table 4-26 Configuration items

Configuration Item	Mandatory	Description
logstash-input-elasticsearch		
hosts	Yes	Source cluster node addresses. You can configure multiple IP addresses. Value format: ["<Node IP address 1>:<Port number>", "<Node IP address 2>:<Port number>"]

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuration Item	Mandatory	Description
index	Yes	Name of the source index to be migrated. • Single index: Enter the index name, for example, **my_index**. • Multiple indexes: Enter multiple index names and use a comma (,) to separate them, for example, **my_index1,my_index2**. • Wildcard: Use the wildcard (*) to match multiple indexes. For example, **myindex*** indicates all indexes whose name starts with **myindex**.
docinfo	No	Whether to retain the metadata of the source documents. The value can be: • **true**: Extracts metadata into the @metadata field of the events. • **false** (default): Only extracts the _source part of documents.
user	No	Username for accessing the source cluster. Mandatory for a security-mode cluster.
password	No	Password for accessing the source cluster. Mandatory for a security-mode cluster.
ssl	No	Whether SSL is enabled for the source cluster. The value can be: • **true**: Uses HTTPS to transmit data. • **false**: Uses HTTP to transmit data.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuration Item	Mandatory	Description
ca_file	No	Path of the CA certificate used to verify the source cluster. Value format: <certificate path><certificate name>, for example, /opt/logstash/extend/certs. • If the source is a CSS Elasticsearch or OpenSearch cluster, the **certificate name** and **certificate path** of the default CA certificate will be obtained. For details, see **Viewing Default CA Certificates**. • If the source is a self-managed or third-party Elasticsearch or OpenSearch cluster, upload the source cluster's security certificate to Logstash and obtain the **certificate name** and **certificate path**. For details, see **Uploading a Custom CA Certificate**.
logstash-output-elasticsearch		
hosts	Yes	Destination cluster node addresses. You can configure multiple IP addresses. Value format: ["<Node IP address 1>:<Port number>", "<Node IP address 2>:<Port number>"]
index	Yes	Name of the index to which events are written. • Single index: Enter the index name, for example, **my_index**. • Multiple indexes: Use dynamic naming (based on event fields) or configure multiple conditional output blocks to route events to different indexes.
document_type	No	Type of document to which events are written. Similar events should be written into documents of the same type.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuration Item	Mandatory	Description
document_id	No	ID of document to which events are written. Specifying an existing ID may cause historical data to be overwritten. For data migration tasks, you are advised to set this parameter to prevent duplicate data.
user	No	Username for accessing the destination cluster. Mandatory for a security-mode cluster.
password	No	Password for accessing the destination cluster. Mandatory for a security-mode cluster.
ssl	No	Whether SSL is enabled for the destination cluster. The value can be: • **true**: Uses HTTPS to transmit data. • **false**: Uses HTTP to transmit data.
cacert	No	Path of the CA certificate used to verify the destination cluster. Value format: <certificate path><certificate name>, for example, /opt/logstash/extend/certs. • If the destination is a CSS Elasticsearch or OpenSearch cluster, the **certificate name** and **certificate path** of the default CA certificate will be obtained. For details, see **Viewing Default CA Certificates**. • If the destination is a self-managed or third-party Elasticsearch or OpenSearch cluster, upload the destination cluster's security certificate to Logstash and obtain the **certificate name** and **certificate path**. For details, see **Uploading a Custom CA Certificate**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuration Item	Mandatory	Description
ssl_certificate_verification	No	Whether SSL certificate verification is enabled for the destination cluster. The value can be: • **true** (default): Uses an SSL certificate to verify the destination cluster. • **false**: Ignores SSL certificate verification.

For more information, see **Elasticsearch input plugin** and **Elasticsearch output plugin** from the official Logstash documentation.

JDBC Template

Ingest data from a MySQL or MariaDB database via JDBC into an Elasticsearch or OpenSearch cluster.

```
# logstash-input-jdbc
input {
    jdbc {
        # Path of the JDBC driver package
        jdbc_driver_library => "/opt/logstash/extend/jars/mariadb-java-client-2.7.0.jar"
        jdbc_driver_class => "org.mariadb.jdbc.Driver"
        jdbc_connection_string => "jdbc:mariadb://xxx.xxx.xxx.xxx:xxx/database_name"
        jdbc_user => "xxx"
        jdbc_password => "xxx"
        # SQL query statement, used to determine the scope of data synchronization
        statement => "select * from table_name"
    }
}

filter {
    # Removes some metadata fields automatically added by Logstash
    mutate {
        remove_field => ["@timestamp", "@version"]
    }
}

# logstash-output-elasticsearch
output {
    elasticsearch {
        # Destination cluster node addresses. No need to include the protocol.
        hosts => ["xxx.xxx.xxx.xxx:9200", "xxx.xxx.xxx.xxx:9200"]
        # Name of the index that events are written into.
        index => "xxx"
        # Mandatory fields for a security-mode cluster. (Delete them for a cluster with the security mode disabled.)
        # user => "xxx"          # Username for accessing the cluster.
        # password => "xxx"      # Password corresponding to the username.
        # If SSL is enabled for the destination cluster, additionally configure the following information:
        # ssl => true
        # cacert => "/opt/logstash/extend/certs"     # Path of the CA certificate used to verify the destination cluster.
        # ssl_certificate_verification => false       # Whether to enable SSL certificate verification for the destination cluster.
    }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 4-27 Configuration items

Configuration Item	Mandatory	Description
logstash-input-jdbc		
jdbc_driver_library	Yes	Path of the JDBC driver package. MariaDB and MySQL JDBC drivers are pre-installed for Logstash clusters in CSS. Value format: <Default certificate path>jars/<Driver name> (for example, /opt/logstash/extend/jars/mariadb-java-client-2.7.0.jar) • For the default certificate path, see **Viewing Default CA Certificates**. • Driver name: – MariaDB driver: **mariadb-java-client-2.7.0.jar** and **mariadb-java-client-2.4.0.jar** – MySQL driver (supported only by Logstash clusters whose image version is not earlier than x.x.x_26.1.0_xxx): **mysql-connector-j-9.4.0.jar**, **mysql-connector-java-8.0.16.jar**, and **mysql-connector-java-8.0.30.jar**
jdbc_driver_class	Yes	Driver class name. • For MariaDB, set this field to **org.mariadb.jdbc.Driver**. • For MySQL, set it to **com.mysql.cj.jdbc.Driver**.
jdbc_connection_string	Yes	Database connection string. • If the MariaDB driver is used, the value format is **jdbc:mariadb://<Database access address>:<Listening port>/<Database name>**. • If the MySQL driver is used, the value format is **jdbc:mysql://<Database access address>:<Listening port>/<Database name>**.
jdbc_user	Yes	Database username
jdbc_password	Yes	Password of the user

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuration Item	Mandatory	Description
statement	Yes	SQL query statement, used to determine the scope of data synchronization.
logstash-output-elasticsearch		
hosts	Yes	Destination cluster node addresses. You can configure multiple IP addresses. Value format: ["<Node IP address 1>:<Port number>", "<Node IP address 2>:<Port number>"]
index	Yes	Name of the index to which events are written. ● Single index: Enter the index name, for example, **my_index**. ● Multiple indexes: Use dynamic naming (based on event fields) or configure multiple conditional output blocks to route events to different indexes.
user	No	Username for accessing the destination cluster. Mandatory for a security-mode cluster.
password	No	Password for accessing the destination cluster. Mandatory for a security-mode cluster.
ssl	No	Whether SSL is enabled for the destination cluster. The value can be: ● **true**: Uses HTTPS to transmit data. ● **false**: Uses HTTP to transmit data.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuration Item	Mandatory	Description
cacert	No	Path of the CA certificate used to verify the destination cluster. Value format: <certificate path><certificate name>, for example, /opt/logstash/extend/certs. • If the destination is a CSS Elasticsearch or OpenSearch cluster, the **certificate name** and **certificate path** of the default CA certificate will be obtained. For details, see **Viewing Default CA Certificates**. • If the destination is a self-managed or third-party Elasticsearch or OpenSearch cluster, upload the destination cluster's security certificate to Logstash and obtain the **certificate name** and **certificate path**. For details, see **Uploading a Custom CA Certificate**.
ssl_certificate_verification	No	Whether SSL certificate verification is enabled for the destination cluster. The value can be: • **true** (default): Uses an SSL certificate to verify the destination cluster. • **false**: Ignores SSL certificate verification.

For more information, see **Jdbc input plugin** and **Elasticsearch output plugin** from the official Logstash documentation.

Kafka Template

Ingest data from Kafka into an Elasticsearch or OpenSearch cluster.

```
# logstash-input-kafka
input {
    kafka {
        # Kafka broker addresses
        bootstrap_servers => "xxx.xxx.xxx.xxx:xxx"
        # Kafka topics to subscribe to
        topics => ["xxx"]
        # Consumer group ID for offset tracking
        group_id => "kafka_es_test"
        # Offset reset policy
        auto_offset_reset => "earliest"
    }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
filter {
   # Removes some metadata fields automatically added by Logstash
   mutate {
      remove_field => ["@timestamp", "@version"]
   }
}

# logstash-output-elasticsearch
output {
   elasticsearch {
      # Destination cluster node addresses. No need to include the protocol.
      hosts => ["xxx.xxx.xxx.xxx:9200", "xxx.xxx.xxx.xxx:9200"]
      # Name of the index that events are written into.
      index => "xxx"
      # Mandatory fields for a security-mode cluster. (Delete them for a cluster with the security mode
disabled.)
      # user => "xxx"           # Username for accessing the cluster.
      # password => "xxx"       # Password corresponding to the username.
      # If SSL is enabled for the destination cluster, additionally configure the following information:
      # ssl => true
      # cacert => "/opt/logstash/extend/certs"       # Path of the CA certificate used to verify the destination
cluster.
      # ssl_certificate_verification => false        # Whether to enable SSL certificate verification for the
destination cluster.
   }
}
```

Table 4-28 Configuration items

Configuration Item	Mandatory	Description
logstash-input-kafka		
bootstrap_servers	Yes	Kafka broker addresses. Multiple addresses can be configured. Value format: "<Kafka broker IP address 1>:<port number>, <Kafka broker IP address 2>:<port number>"
topics	Yes	Kafka topics to subscribe to. • Single topic: Enter the topic name, for example, [**log**]. • Multiple topics: Enter multiple topic names and separate them with a comma (,), for example, [**log1**,**log2**].
group_id	Yes	Consumer group ID for offset tracking

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuration Item	Mandatory	Description
auto_offset_reset	Yes	Offset reset policy. The value can be: • **earliest**: Start consuming from the earliest available messages. • **latest**: Start consuming from the most recent messages. • **none**: Report an error if no previous offset is found for the consumer group. • **anything else**: Throws an exception to the consumer.
logstash-output-elasticsearch		
hosts	Yes	Destination cluster node addresses. You can configure multiple addresses. Value format: ["<Node IP address 1>:<port number>", "<Node IP address 2>:<port number>"].
index	Yes	Name of the index to which events are written. • Single index: Enter the index name, for example, **my_index**. • Multiple indexes: Use dynamic naming (based on event fields) or configure multiple conditional output blocks to route events to different indexes.
user	No	Username for accessing the destination cluster. Mandatory for a security-mode cluster.
password	No	Password for accessing the destination cluster. Mandatory for a security-mode cluster.
ssl	No	Whether SSL is enabled for the destination cluster. The value can be: • **true**: Uses HTTPS to transmit data. • **false**: Uses HTTP to transmit data.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuration Item	Mandatory	Description
cacert	No	Path of the CA certificate used to verify the destination cluster. Value format: <certificate path><certificate name>, for example, /opt/logstash/extend/certs. • If the destination is a CSS Elasticsearch or OpenSearch cluster, the **certificate name** and **certificate path** of the default CA certificate will be obtained. For details, see **Viewing Default CA Certificates**. • If the destination is a self-managed or third-party Elasticsearch or OpenSearch cluster, upload the destination cluster's security certificate to Logstash and obtain the **certificate name** and **certificate path**. For details, see **Uploading a Custom CA Certificate**.
ssl_certificate_verification	No	Whether SSL certificate verification is enabled for the destination cluster. The value can be: • **true** (default): Uses an SSL certificate to verify the destination cluster. • **false**: Ignores SSL certificate verification.

For more information, see **Kafka input plugin** and **Elasticsearch output plugin** from the official Logstash documentation.

Beats Template

Ingests data from Beats into an Elasticsearch or OpenSearch cluster.

```
# logstash-input-beats
input {
    beats {
        # Logstash listening port, used for receiving data from Filebeat or Metricbeat.
        port => 5044
    }
}

filter {
    # Removes some metadata fields automatically added by Logstash
    mutate {
        remove_field => ["@timestamp", "@version"]
    }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
# logstash-output-elasticsearch
output {
   elasticsearch {
      # Destination cluster node addresses. No need to include the protocol.
      hosts => ["xxx.xxx.xxx.xxx:9200", "xxx.xxx.xxx.xxx:9200"]
      # Name of the index that events are written into.
      index => "xxx"
      # Mandatory fields for a security-mode cluster. (Delete them for a cluster with the security mode disabled.)
      # user => "xxx"           # Username for accessing the cluster.
      # password => "xxx"       # Password corresponding to the username.
      # If SSL is enabled for the destination cluster, additionally configure the following information:
      # ssl => true
      # cacert => "/opt/logstash/extend/certs"      # Path of the CA certificate used to verify the destination cluster.
      # ssl_certificate_verification => false       # Whether to enable SSL certificate verification for the destination cluster.
   }
}
```

Table 4-29 Configuration items

Configuration Item	Mandatory	Description
logstash-input-beats		
port	Yes	Logstash listening port, used for receiving data from Filebeat or Metricbeat. Make sure this port (for example, 5044) is not occupied by other processes, and that it is allowed by the Logstash cluster's security group rules.
logstash-output-elasticsearch		
hosts	Yes	Destination cluster node addresses. You can configure multiple IP addresses. Value format: ["<Node IP address 1>:<Port number>", "<Node IP address 2>:<Port number>"]
index	Yes	Name of the index to which events are written. • Single index: Enter the index name, for example, **my_index**. • Multiple indexes: Use dynamic naming (based on event fields) or configure multiple conditional output blocks to route events to different indexes.
user	No	Username for accessing the destination cluster. Mandatory for a security-mode cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Configuration Item	Mandatory	Description
password	No	Password for accessing the destination cluster. Mandatory for a security-mode cluster.
ssl	No	Whether SSL is enabled for the destination cluster. The value can be: • **true**: Uses HTTPS to transmit data. • **false**: Uses HTTP to transmit data.
cacert	No	Path of the CA certificate used to verify the destination cluster. Value format: <certificate path><certificate name>, for example, /opt/logstash/extend/certs. • If the destination is a CSS Elasticsearch or OpenSearch cluster, the **certificate name** and **certificate path** of the default CA certificate will be obtained. For details, see **Viewing Default CA Certificates**. • If the destination is a self-managed or third-party Elasticsearch or OpenSearch cluster, upload the destination cluster's security certificate to Logstash and obtain the **certificate name** and **certificate path**. For details, see **Uploading a Custom CA Certificate**.
ssl_certificate_verification	No	Whether SSL certificate verification is enabled for the destination cluster. The value can be: • **true** (default): Uses an SSL certificate to verify the destination cluster. • **false**: Ignores SSL certificate verification.

For more information, see **Beats input plugin** and **Elasticsearch output plugin** from the official Logstash documentation.

4.7 Monitoring

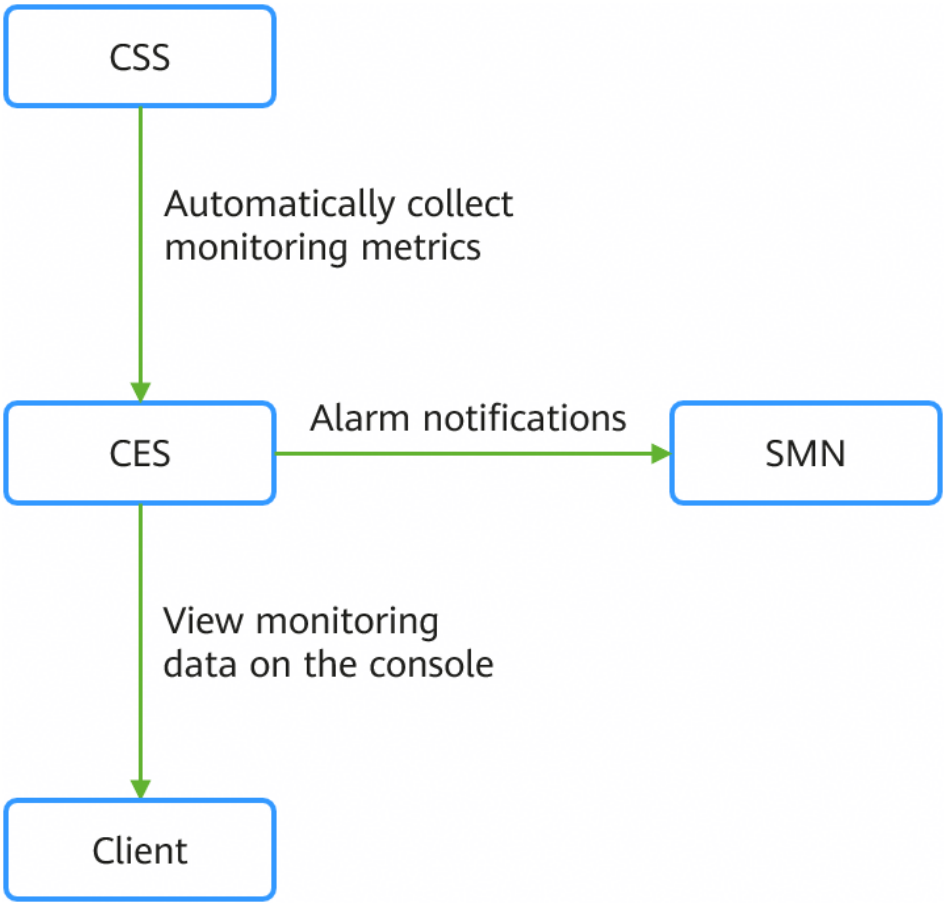
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

4.7.1 Using Cloud Eye to Monitor Clusters

Logstash clusters serve as core components in data processing pipelines, continuously handling high-load tasks such as log ingestion, format conversion, and data distribution. If a node experiences CPU overload or insufficient JVM memory, pipeline processing delays and data backlogs can occur. In severe cases, nodes may exit unexpectedly, resulting in data loss. Relying on operations personnel to periodically log in to clusters and manually check metrics is inefficient and makes it difficult to identify issues at an early stage. To address this, CSS integrates with Cloud Eye to collect and visualize key runtime metrics, such as CPU usage and JVM memory metrics, for Logstash nodes in real time. When metrics exceed configured thresholds, Cloud Eye sends alarm notifications through Simple Message Notification (SMN). This helps operations personnel shift from post-failure troubleshooting to proactive issue prevention.

How the Feature Works

Figure 4-6 Monitoring data flow



The monitoring data flow is as follows:

1. Clusters automatically report monitoring data to Cloud Eye.
2. You can set alarm rules for CSS on Cloud Eye. You can configure alarm notifications via SMN.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3. You can then view monitoring metrics on the console.

Cloud Eye provides the following four monitoring dimensions for Logstash clusters.

Table 4-30 Monitoring dimensions

Dimension	Description	Applicable Scenario
Cluster	Aggregates metrics at the Logstash cluster level to reflect overall cluster resource utilization.	Overall health checks and cluster-level alerting
Node	Collects resource metrics for individual Logstash nodes (VMs).	Identifying hot nodes and troubleshooting CPU/JVM bottlenecks on individual nodes
Pipeline	Aggregates runtime metrics for all pipelines in a cluster, such as event throughput and processing latency.	Monitoring overall pipeline performance and throughput and detecting global pipeline issues
Single-node pipeline	Collects runtime metrics for pipelines running on an individual node.	Precisely locating processing bottlenecks in specific pipelines on a specific node

Recommended Monitoring Metrics

Table 4-31 Recommended metrics

Dimension	Recommended Metrics	Description
CPU and JVM	Average JVM heap usage, maximum JVM heap usage, average CPU usage, and maximum CPU usage	Continuously high JVM heap usage (> 75%) may cause frequent GC and affect data processing performance.
Pipeline throughput	logstash_pipeline_events_in, logstash_pipeline_events_filtered, logstash_pipeline_events_out	A continuously increasing difference between input and output events usually indicates insufficient processing capacity or downstream blocking.

Constraints

- Runtime duration: The cluster must run for more than 10 minutes before Cloud Eye can collect valid monitoring data.
- Operation permissions: The account used for operations must have the required Cloud Eye permissions. Otherwise, you cannot view monitoring information or configure alarm rules.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Prerequisites

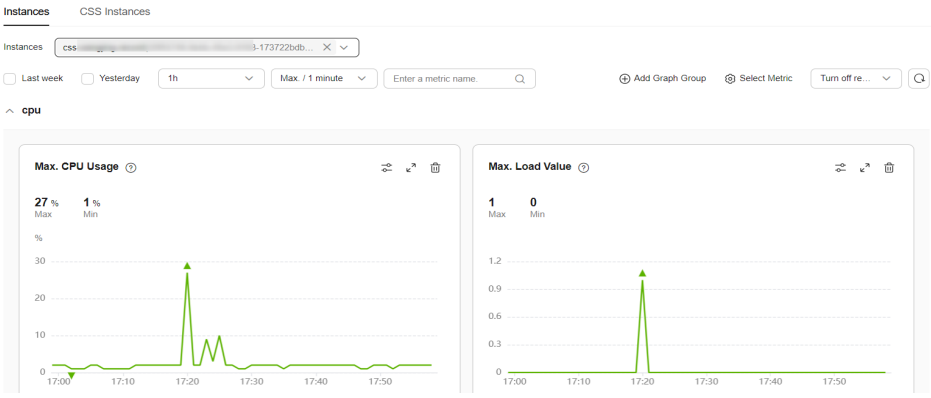
The Logstash cluster is available.

Viewing Cluster/Node Monitoring Metrics

You can view the real-time monitoring metrics of clusters and nodes on the built-in monitoring page of the CSS console. You can compare data across multiple clusters and perform period-over-period analysis.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. In the cluster list, find the target cluster, and click **Monitoring Metrics** in the **Operation** column.
4. View cluster-level or node-level monitoring data as needed.

Figure 4-7 Monitoring metrics



- **Viewing cluster-level monitoring metrics**

  On the monitoring metrics page, select the **Instances** tab.

  - **Instances**: Select the cluster whose monitoring data you want to view. Data for the current cluster is displayed by default. You can select multiple clusters for horizontal comparison.
  - You can set the monitoring time range and view period-over-period comparisons for the same cluster to identify periodic patterns or abnormal deviations.
  - **Create Graph Group**: Add custom groups on top of the default groups to group related metrics for focused analysis. When you move the pointer over a group name on the left, the icons for changing the group name and deleting the group are displayed on the right. You can modify the group.
  - **Select Metric**: In the **Select Metric** dialog box, you can select both **All Metrics** (single line charts) and **Top N Metrics** (comparison of the top N nodes by metric value).

- **Viewing node-level monitoring metrics**

  On the monitoring metrics page, select the **CSS Instances** tab.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- **CSS Instances**: Select the cluster nodes whose monitoring data you want to view. Data for the first node in the current cluster is displayed by default. You can select multiple nodes for horizontal comparison.
- You can also set monitoring periods so that you can compare metrics across different periods for the same cluster.
- **Create Graph Group**: Add custom groups on top of the default groups to group related metrics for focused analysis. When you move the pointer over a group name on the left, the icons for changing the group name and deleting the group are displayed on the right. You can modify the group.
- Select Metric: In the displayed **Select Metric** dialog box, set **All Metrics** (per-metric charts).

5. (Optional) To view monitoring dashboards, see **Method 1: Monitoring Trends on the Monitoring Page (Monitoring Dashboards)**.

   Monitoring dashboards display time-series trend charts for multiple metrics in one place, allowing you to quickly understand the overall cluster status during routine checks.

Viewing Pipeline Monitoring Metrics

You can view cluster-level or node-level pipeline runtime metrics through the Configuration Center to identify pipeline processing bottlenecks.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. In the cluster list, find the target cluster and click **Configuration Center** in the **Operation** column.
4. In the pipeline list, click **Monitoring Metrics** in the **Operation** column. On the displayed Cloud Eye monitoring metrics page, you can view the aggregated monitoring metrics for all pipelines in the current cluster.
5. Click the back button in the upper part of the page to go to the custom monitoring page.
6. View cluster-level or node-level pipeline monitoring data as needed.

   a. **Viewing cluster-level pipeline metrics**

      On the custom monitoring page, click the **cluster_id,pipeline_name** tab, select the monitoring object, and click **View Metric** in the **Operation** column to view metrics.

   b. **Viewing node-level pipeline metrics**

      On the custom monitoring page, click the **cluster_id,instance_id,pipeline_name** tab, select the monitoring object, and click **View Metric** in the **Operation** column to view metrics.

Configuring Alarm Rules

You can configure alarm thresholds and notification methods for monitoring metrics. When metrics exceed configured thresholds, notifications are

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

automatically sent to specified destinations, enabling operations personnel to shift from reactive troubleshooting to proactive monitoring.

1. Log in to the **Cloud Eye console**.
2. In the navigation pane on the left, choose **Alarm Management** > **Rule Settings**.
3. In the alarm rule list, search for alarm rules whose **Resource Type** is **Cloud Search Service** (cluster or node level) or **CSS.CUSTOM** (pipeline level), and check whether these rules meet your requirements.

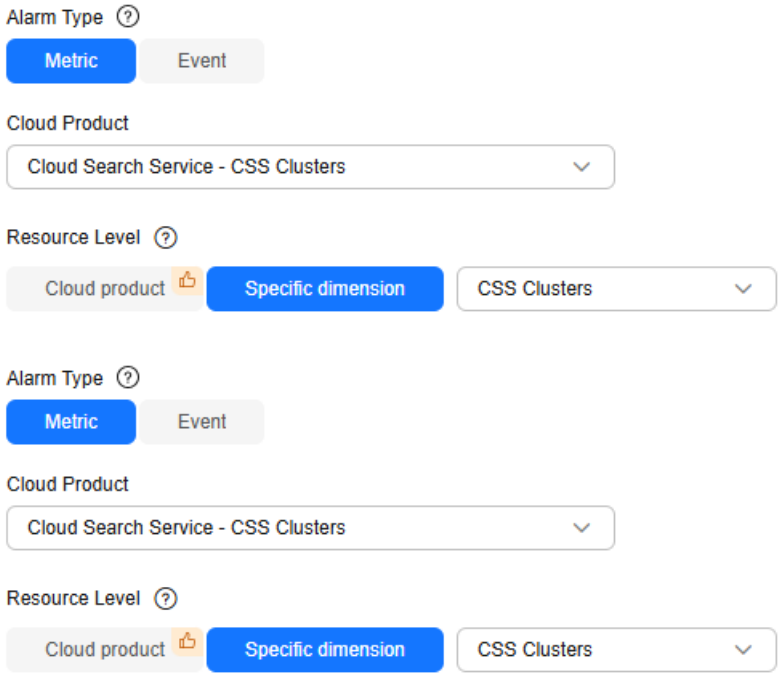
   To create new alarm rules, see **Creating an Alarm Rule and Notifications**. The following describes key parameters. Configure other parameters based on your requirements.

   – **Cluster/Node-level alarm rules**

   Table 4-32 Key parameters for configuring a cluster/node-level alarm rule

Parameter	Description
Alarm Type	Select **Metric**.
Cloud Product	Select **Cloud Search Service - CSS Clusters**.
Resource Level	Select **Specific dimension**. In the drop-down list box on the right, select a specific dimension. • **CSS Clusters**: Creates an alarm rule at the cluster level. • **CSS Clusters - CSS Instances**: Creates an alarm rule at the individual node level.

   Figure 4-8 Configuring a cluster/node-level alarm rule



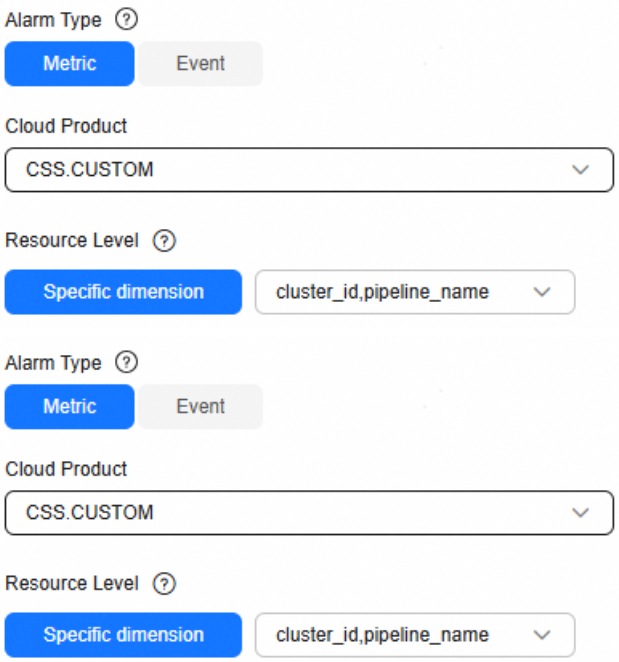
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- **Pipeline-level alarm rules**

Table 4-33 Key parameters for configuring a pipeline-level alarm rule

Parameter	Description
Alarm Type	Select **Metric**.
Cloud Product	Select **CSS.CUSTOM**.
Resource Level	Select a specific dimension to configure the alarm rule. • **cluster_id,pipeline_name**: Creates an alarm rule based on aggregated metrics of all pipelines in a cluster. • **cluster_id,instance_id,pipeline_name**: Creates an alarm rule based on pipelines running on an individual node in a cluster.

Figure 4-9 Configuring a pipeline-level alarm rule



4. After an alarm rule is created, it is displayed in the alarm rule list. The rule **Status** is **Enabled**.

4.7.2 Supported Metrics

You can use Cloud Eye to monitor CSS cluster metrics in real time and quickly handle exceptions.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Function

This section describes CSS metrics that can be monitored by Cloud Eye as well as their namespaces and dimensions. You can query metrics and alerts generated by CSS on the **Cloud Eye console** or using an **API**.

- If the configuration center of a Logstash cluster does not have any records of operations on the pipeline list, the monitoring records of this Logstash cluster are empty.
- When the **Events** data of a pipeline changes dynamically, the monitoring data changes accordingly. When a pipeline task is being started or stopped, or the **Events** data is stable, the monitoring data remains unchanged.

NOTE

Cloud Eye can monitor dimensions (or objects) nested to a maximum depth of four levels (levels 0 to 3). 3 is the deepest level. For example, if the monitored dimension of a metric is **cluster_id,instance_id**, **cluster_id** indicates level 0 and **instance_id** indicates level 1.

Namespaces

SYS.ES

CSS.CUSTOM

Cluster Monitoring Metrics

NOTE

Accumulated value: The value is accumulated from when a node is started. After the node is restarted, the value is reset to zero and starts accumulating again.

Table 4-34 Logstash cluster monitoring metrics

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
max_jvm_heap_usage	Max. JVM Heap Usage	Maximum JVM heap usage of nodes in a CSS cluster	0~100	%	N/A	cluster_id	1 minute
max_jvm_young_gc_time	Max. JVM Young GC Duration	Maximum accumulated JVM Young GC duration of nodes in a CSS cluster	≥ 0	ms	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
max_jvm_young_gc_count	Max. JVM Young GC Count	Maximum accumulated JVM Young GC count of nodes in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute
max_jvm_old_gc_time	Max. JVM Old GC Duration	Maximum accumulated JVM Old GC duration of nodes in a CSS cluster	≥ 0	ms	N/A	cluster_id	1 minute
max_jvm_old_gc_count	Max. JVM Old GC Count	Maximum accumulated JVM Old GC count of nodes in a CSS cluster.	≥ 0	Count	N/A	cluster_id	1 minute
max_cpu_usage	Max. CPU Usage	Maximum node CPU usage in a CSS cluster	0~100	%	N/A	cluster_id	1 minute
max_load_average	Max. Node Load	Maximum number of average queued tasks per minute on nodes in a cluster	≥ 0	Count	N/A	cluster_id	1 minute
avg_cpu_usage	Avg. CPU Usage	Average node CPU usage in a CSS cluster.	0~100	%	N/A	cluster_id	1 minute
avg_load_average	Avg. Node Load	Average number of queued tasks per minute on nodes in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
avg_jvm_heap_usage	Avg. JVM Heap Usage	Average node JVM heap usage in a CSS cluster	0~100	%	N/A	cluster_id	1 minute
avg_jvm_old_gc_count	Avg. GCs of Old-Generation JVM	Average number of old-generation garbage collections of nodes in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute
avg_jvm_old_gc_time	Avg. GC Duration of Old-Generation JVM	Average old-generation garbage collection duration of nodes in a CSS cluster	≥ 0	ms	N/A	cluster_id	1 minute
avg_jvm_young_gc_count	Avg. GCs of Young-Generation JVM	Average number of young-generation garbage collections of nodes in a CSS cluster	≥ 0	Count	N/A	cluster_id	1 minute
avg_jvm_young_gc_time	Avg. GC Duration of Young-Generation JVM	Average young-generation garbage collection duration of nodes in a CSS cluster	≥ 0	ms	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
sum_events_in	Total Records Passed Through the Input Plug-in	Total number of records that have passed through the **input** plugin on all the nodes in a cluster	≥ 0	Count	N/A	cluster_id	1 minute
sum_events_filtered	Total Records Passed Through the Filter Plug-in	Total number of records that have passed through the **filter** plugin on all the nodes in a cluster	≥ 0	Count	N/A	cluster_id	1 minute
sum_events_out	Total Records Passed Through the Out Plug-in	Total number of records that have passed through the **out** plugin on all the nodes in a cluster	≥ 0	Count	N/A	cluster_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Node Monitoring Metrics

Table 4-35 Logstash node monitoring metrics

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
jvm_heap_usage	JVM Heap Usage	JVM heap memory usage of a node.	0~100	%	N/A	cluster_id,instance_id	1 minute
cpu_usage	CPU Usage	CPU usage.	0~100	%	N/A	cluster_id,instance_id	1 minute
load_average	Average Load	Average number of queued tasks per minute on a node	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
jvm_old_gc_count	Total GCs of Old-Generation JVM	Number of old-generation garbage collection times	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
jvm_old_gc_time	Total GC Duration of Old-Generation JVM	Time spent on old-generation garbage collection	≥ 0	ms	N/A	cluster_id,instance_id	1 minute
jvm_young_gc_count	Total GCs of Young-Generation JVM	Number of young-generation garbage collection times	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
jvm_young_gc_time	GC Duration of Young-Generation JVM	Time spent on young-generation garbage collection	≥ 0	ms	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
events_in	Records Passed Through the Input Plug-in on Node	Number of data records that have passed through the **input** plugin on the current node	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
events_filtered	Records Passed Through the Filter Plug-in on Node	Number of records that have passed through the **filter** plugin on the current node	≥ 0	Count	N/A	cluster_id,instance_id	1 minute
events_out	Records Passed Through the Out Plug-in on Node	Number of records that have passed through the **out** plugin on the current node	≥ 0	Count	N/A	cluster_id,instance_id	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Logstash Pipeline Monitoring Metrics

Table 4-36 Logstash pipeline monitoring metrics

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
logstash_pipeline_events_in	Records Passed Through the Input Plug-in into Pipeline	Number of records that have passed through the **input** plug-in during the current pipeline monitoring period	≥ 0	Count	N/A	cluster_id,instance_id,pipeline_name or cluster_id,pipeline_name	1 minute
logstash_pipeline_events_filtered	Records Passed Through the Filter Plug-in	Number of records that have passed through the **filtered** plug-in during the current pipeline monitoring period	≥ 0	Count	N/A	cluster_id,instance_id,pipeline_name or cluster_id,pipeline_name	1 minute

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Metric ID	Metric	Description	Value Range	Unit	Number System	Dimension	Monitoring Interval (Raw)
logstash_pipeline_events_out	Records Passed Through the Out Plug-in Out of Pipeline	Number of records that have passed through the **out** plug-in during the current pipeline monitoring period	≥ 0	Count	N/A	cluster_id,instance_id,pipeline_name or cluster_id,pipeline_name	1 minute

If an object is in a hierarchical system, specify the monitored dimension in hierarchical form when you use an API to query the metrics of this object.

For example, to query the CPU usage (cpu_usage) of a Logstash cluster node in CSS, the dimension information of this metric is **cluster_id,instance_id**, where **cluster_id** indicates level 0 and **instance_id** level 1.

- To query a single metric by calling an API, the **instance_id** dimension is used as follows:
  ```
  dim.0=cluster_id,3d65c1ac-9a9f-4c5f-a054-35184a087bb2&dim.1=instance_id,6666cd76f96956469e7be39d750cc7d9
  ```
  where, **3d65c1ac-9a9f-4c5f-a054-35184a087bb2** and **6666cd76f96956469e7be39d750cc7d9** are the values of **cluster_id** and **instance_id**, respectively. For how to obtain these values, see **Dimensions**.
- To query multiple metrics by calling an API, the **instance_id** dimension is used as follows:
  ```
  "dimensions": [
    {
      "name": "cluster_id",
      "value": "3d65c1ac-9a9f-4c5f-a054-35184a087bb2"
    },
    {
      "name": "instance_id",
      "value": "6666cd76f96956469e7be39d750cc7d9"
    }
  ]
  ```
  where, **3d65c1ac-9a9f-4c5f-a054-35184a087bb2** and **6666cd76f96956469e7be39d750cc7d9** are the values of **cluster_id** and **instance_id**, respectively. For how to obtain the values, see **Dimensions**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Dimensions

Table 4-37 Dimension description

Key	Value
cluster_id	Cluster ID. You can obtain this value from the **clusters[].id** field in the response body of the **Querying the Cluster List** API.
instance_id	Cluster node ID. You can obtain this value from the **clusters[].instances[].id** field in the response body of the **Querying the Cluster List** API.
pipeline_name	Logstash pipeline name. You can obtain this value from the **pipelines[].name** field in the response body of the **Querying the Pipeline List**.

4.8 Managing Logs

4.8.1 Querying Logs

When using Logstash for large-scale data migration or real-time log ingestion, errors such as incorrect pipeline configurations, broken connections between the source and destination, or erroneous data processing logic may interrupt data flows. CSS provides a log query feature that aggregates underlying server logs to the console, where you can query them by node and keyword in seconds. By analyzing recent logs, you can gain real-time insight into each phase of the pipelines and accurately locate the root causes of exceptions such as connection timeouts or configuration syntax errors.

Querying Recent Logs

Query recently generated logs that are not yet archived.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Logs > Log Search**. The **Log Search** page is displayed.

   You can search log records by node or keyword. For a detailed description of each type of logs, see **Log Types**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

> **CAUTION**
>
> When a log file reaches 128 MB or when the time reaches 00:00 UTC, the system automatically compresses and archives it. Only unarchived logs appear on the log search page, while archived logs remain accessible through the log backup function. For more information, see **Backing Up Logs to OBS**.

Log Types

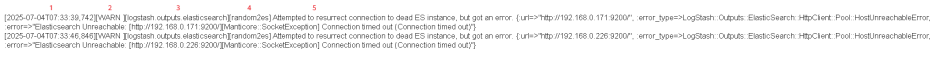
Table 4-38 Introducing different log types

Log Type	Description	Purpose
Run logs	Run logs record a cluster's node and pipeline status, such as source-destination connectivity, pipeline creation or modification, and pipeline running errors.	Check run logs to troubleshoot pipeline errors.
Deprecation logs	Deprecation logs record deprecation warnings. Deprecation warnings are written to this log when you use APIs, configurations, or functions that are marked for removal in future versions. You cannot check deprecation logs on the console. To check them, you need to back them up to an OBS bucket first.	Check for APIs or features that are about to expire in future versions.

- **Run log description**

  Run logs record a cluster's node and pipeline status. For example, the log record below indicates that the destination cluster could not be reached. You need to check whether the cluster address is correct and whether the cluster status is normal.

  Figure 4-10 A sample of run logs

  Log content:

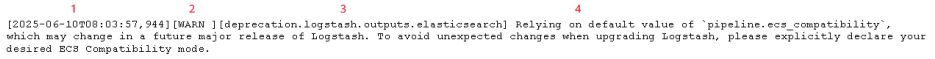
  - 1. Log generation time
  - 2. Log level, which can be DEBUG, INFO, WARN, or ERROR
  - 3. Log-generating module
  - 4. Name of the log-generating node

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

  - 5. Log content
- **Deprecation log description**

  Deprecation logs record deprecation warnings.

  Figure 4-11 A sample of deprecation logs

  Log content:

  - 1. Log generation time
  - 2. Log level, which can only be DEPRECATION.
  - 3. Log-generating module
  - 4. Log content. The log record shown in the figure above indicates that the ECS compatibility mode was not explicitly declared for a plugin when Logstash was started.

4.8.2 Backing Up Logs to OBS

During routine O&M, querying logs in real-time through the web console is a commonly used method for diagnosing issues. However, due to limited local storage capacity, the system automatically archives historical logs when they exceed 128 MB or at 00:00 UTC. Long-term log retention is essential for compliance and operational purposes, but it must also be cost-effective. To address this, CSS provides a log backup feature that incrementally synchronizes run logs and deprecation logs to OBS based on predefined policies. This solution takes advantage of OBS's vast, low-cost storage for long-term retention while also allowing users to download historical log files at any time for detailed offline analysis.

Impact on Billing

Standard storage is used for storing log backups in OBS. For pricing details, see **Billing**section "Billing" in *Object Storage Service User Guide*.

Prerequisites

You have created an OBS bucket for storing log backups. The OBS bucket must meet the following requirements:

- **Storage Class**: **Standard**.
- **Region**: the same as that of your CSS cluster.

Enabling Log Backup

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Logs > Log Backup**. The **Log Backup** page is displayed.

   Check whether log backup is enabled for the cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- Yes: No further action is required.
- No: Go to the next step.

5. On the **Log Backup** page, click **Enable Backup**. In the displayed dialog box, configure the necessary settings.

Table 4-39 Log backup settings

Parameter	Description

OBS Bucket	Select an OBS bucket for storing log backups from the drop-down list box.
	If no OBS buckets meet your requirements, click **Create Bucket** to go to the OBS console and create one. For details, see **Creating a Bucket**.
	An IAM user must be granted the following OBS permissions to view available OBS buckets:
	`obs:bucket:getBucketStoragePolicy`
	`obs:bucket:listAllMyBuckets`

Backup Path	Set the log storage location in the OBS bucket. The backup path cannot: • Contain the following characters: **\:*?"<>|'{}** • Start with a slash (/). • Start or end with a period (.). • Contain more than two consecutive slashes (/) or periods (.). • Exceed 512 characters.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
IAM Agency	To back up data to an OBS bucket, you must have the write permission to it. By configuring an IAM agency, you can authorize CSS to access its OBS resources through an associated account. • If you are configuring an agency for the first time, click **Automatically Create IAM Agency** to create **css-obs-agency**. • If there is an IAM agency automatically created earlier, you can click **One-click authorization** to have the OBS Administrator permissions deleted automatically, and have the following custom policies added automatically instead to implement more refined permissions control. "obs:bucket:GetBucketLocation", "obs:object:GetObjectVersion", "obs:object:GetObject", "obs:object:DeleteObject", "obs:bucket:HeadBucket", "obs:bucket:GetBucketStoragePolicy", "obs:object:DeleteObjectVersion", "obs:bucket:ListBucketVersions", "obs:bucket:ListBucket", "obs:object:PutObject" • When OBS buckets use SSE-KMS encryption, the IAM agency must be granted KMS permissions. You can click **Automatically Create IAM Agency** and **One-click authorization** to have the following custom policies created automatically. "kms:cmk:create", "kms:dek:create", "kms:cmk:get", "kms:dek:decrypt", "kms:cmk:list" • To use **Automatically Create IAM Agency** and **One-click authorization**, the following minimum permissions are required: "iam:agencies:listAgencies", "iam:roles:listRoles", "iam:agencies:getAgency", "iam:agencies:createAgency", "iam:permissions:listRolesForAgency", "iam:permissions:grantRoleToAgency", "iam:permissions:listRolesForAgencyOnProject", "iam:permissions:revokeRoleFromAgency", "iam:roles:createRole" • To use an IAM agency, the following minimum permissions are required: "iam:agencies:listAgencies", "iam:agencies:getAgency", "iam:permissions:listRolesForAgencyOnProject", "iam:permissions:listRolesForAgency" **WARNING** The agency name can contain only letters (case-sensitive), digits, underscores (_), and hyphens (-). Otherwise, the backup will fail.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Determine whether to enable **Automatic backup** based on service needs.

- To enable automatic, periodic log backup, select **Automatic backup**, and set the backup policy. For more, see **Configuring Automatic Log Backup**.
- If manual log backup is sufficient, deselect **Automatic backup**.

6. Click **OK** to enable log backup.

   The configuration information will be displayed on the **Log Backup** tab.

Configuring Automatic Log Backup

After enabling log backup for a cluster, you can configure an auto backup policy to periodically back up logs based on a preset schedule.

1. On the **Log Backup** page, click **Modify Settings** on the right.

   Figure 4-12 Modifying Settings

2. In the **Modify Settings** dialog box, select **Automatic backup**, and configure the backup policy.

   Table 4-40 Automatic backup settings

Parameter	Description
Time Zone	Select a time zone for the backup start time.
Backup Start Time	Specify the start time of auto backup. Select a value from the drop-down list. The value range is from **00:00** to **23:00**. The backup always happens on the hour.

3. Click **OK** to enable automatic log backup.

   Automatic log backup tasks are displayed in the task list. When **Task Status** in the task list changes to **Succeeded**, the backup is successful.

4. If automatic log backup is no longer required, click **Modify Settings**. In the displayed dialog box, deselect **Automatic backup**, and click **OK**. In the displayed confirmation dialog box, click **OK**.

   Disabling log backup does not automatically delete log backups automatically generated earlier. Instead, you need to manually delete them on the OBS console.

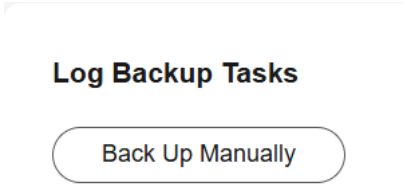
Manually Backing Up Logs

After enabling log backup for a cluster, you can manually back up its logs at any time.

1. On the **Log Backup** page, click **Back Up Manually** under **Log Backup Tasks**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 4-13 Manually backing up logs



2. In the displayed dialog box, confirm the log backup path and click **OK**.

   The log backup task is displayed in the task list. When **Task Status** in the task list changes to **Succeeded**, the backup is successful.

Checking Log Backup Files

Logs are backed up incrementally. After the backup is successful, you can access the target OBS bucket to obtain the full log files by clicking **Log Path**.

Figure 4-14 OBS bucket address

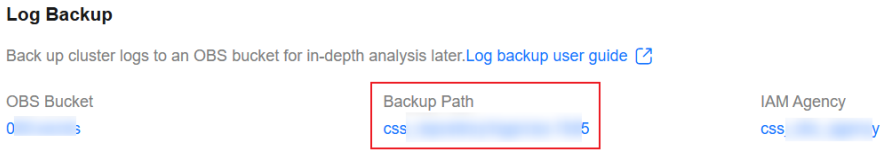


Table 4-41 lists the log types.

Table 4-41 Log types

Log Name	Description
logstash-deprecation.log	Deprecation log file
logstash-plain.log	Run log file

Disabling Log Backup

Disable log backup if it is no longer needed.

On the **Log Backup** page, click **Disable Backup**. In the displayed dialog box, click **OK**.

CAUTION

Disabling log backup does not automatically delete existing backups. Instead, you need to manually delete them on the OBS console.

4.8.3 Viewing Audit Logs

To ensure data security and compliance, enterprises must routinely record and audit operations performed on their cloud service resources. With Cloud Trace

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Service (CTS), you can record operations associated with CSS Logstash clusters for later query, auditing, and backtracking.

Prerequisites

CTS has been enabled.

Key Operations That Can Be Audited Using CTS

Table 4-42 Key operations that can be audited using CTS

Operation	Resource Type	Event Name
Querying product dynamics on the Overview page	cluster	listByGraph
Querying cluster and disk information on the Dashboards page	cluster	showDashboard
Querying supported engine types	cluster	listDatastores
Querying flavors supported by a specified engine type	cluster	getSingleDatabaseFlavors
Querying supported AZs	user	getAZInfo
Querying available disk types	cluster	listDiskTypes
Obtaining the node flavor list	cluster	listFlavors
Querying details about a specified flavor	cluster	showFlavorDetail
Querying feature availability	user	showFunctionSwitch
Querying available resource quotas	user	queryQuota
Creating a cluster	cluster	createCluster
Querying the cluster list	cluster	listClusters
Querying cluster details	cluster	showClusterDetail
Querying the storage capacity of a cluster	cluster	getClusterDiskInfos
Updating resource status	cluster	updateServiceStatus
Updating resource metadata	cluster	updateServiceMetadata
Changing the cluster name	cluster	updateClusterName
Modifying the cluster description	cluster	updateClusterDesc

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Operation	Resource Type	Event Name
Clearing the status of a specified task for a cluster	cluster	clearClusterAction
Forcibly restarting VMs in a cluster	cluster	rebootCluster
Deleting a cluster	cluster	deleteCluster
Creating a cluster (V2)	cluster	createClusterV2
Querying the cluster list	cluster	listClustersDetails
Querying the cluster list (V2)	cluster	listClustersDetailsV2
Changing the security group	cluster	changeSecurityGroup
Adding more instances	cluster	updateExtendInstanceStorage
Removing nodes of a specific type	cluster	updateShrinkCluster
Querying information about nodes to be deleted	cluster	queryNeedDeleteInstances
Changing the flavor of a specified node type	cluster	updateFlavorByType
Querying available flavors	cluster	queryResizeFlavors
Creating a configuration file	cluster	createCnf
Updating a configuration file	cluster	updateCnf
Starting a data migration pipeline	cluster	startPipeline
Stopping a data migration pipeline	cluster	stopPipeline
Adding a custom template	cluster	addFavorite
Testing connectivity	cluster	startConnectivityTest
Querying the template list	cluster	listTemplates
Querying the configuration file list	cluster	listConfs
Querying the pipeline list	cluster	listPipelines
Querying operation records	cluster	listActions
Querying the content of a configuration file	cluster	showGetConfDetail
Deleting a configuration file	cluster	deleteConf

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Operation	Resource Type	Event Name
Deleting a custom template	cluster	deleteTemplate
Hot-starting a data migration pipeline	cluster	startHotPipeline
Hot-stopping a data migration pipeline	cluster	stopHotPipeline
Updating cluster routes	cluster	updateRoute
Obtaining a cluster route	cluster	getRoutes
Uploading a certificate file	cluster	uploadCerts
Querying the certificate list	cluster	listCerts
Querying certificate information	cluster	showCertsDetail
Deleting a certificate file	cluster	deleteCerts
Enabling logging	cluster	startLogs
Disabling logging	cluster	stopLogs
Querying the log backup task list	cluster	listLogsJob
Querying basic log configurations	cluster	showGetLogSetting
Modifying basic log configurations	cluster	updateLogSetting
Enabling automatic log backup	cluster	startLogAutoBackupPolicy
Disabling automatic log backup	cluster	stopLogAutoBackupPolicy
Manually backing up logs	cluster	createLogBackup
Configuring automatic log backup	cluster	autoCreateLogBackup
Querying logs	cluster	showLogBackup
Creating an agency	user	createAgency
Querying the agency list	cluster	listAgencies
Querying available OBS buckets	user	getAvailableBuckets
Querying files in a specified OBS bucket	cluster	listObjects

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Operation	Resource Type	Event Name
Querying resources by tag	cluster	getResourceByTags

Querying Real-Time Traces

After a management tracker is created on the CTS console, the system starts recording operations performed on cloud service resources. After a data tracker is created, the system starts recording operations performed on data in OBS buckets. CTS retains operation records generated in the latest seven days.

4.9 Changing Clusters

4.9.1 Adding New Nodes

If the workloads on the data plane of a Logstash cluster increase, you can scale out the cluster by adding more nodes.

You may scale out a Logstash cluster for the following purposes:

- To ensure continuous data availability, more nodes may be added to a Logstash cluster to enhance fault tolerance.
- In the face of fast data growth, more nodes may be added to a Logstash cluster to increase data processing capacity.
- More nodes may be added to a Logstash cluster to execute more complex data processing tasks, such as data enrichment and ETL.

Scale-out process: Add more nodes of the appropriate specifications to the Logstash cluster and modify cluster configuration.

Impact on Billing

For a pay-per-use cluster, you can see its new price when confirming the scale-out or scale-up on the console. After the change is complete, the cluster will be billed based on the new price. For details, see **Cloud Search Service Price Calculator**.

Change Impact

Scaling out a Logstash cluster does not affect services. However, once started, a scale-out task cannot be stopped until it succeeds or fails.

Scale-out Duration

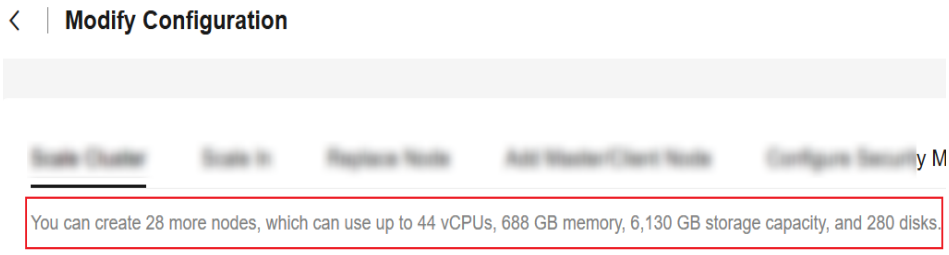
The scale-out operation takes 10 to 30 minutes, depending on the cluster's scheduling capabilities.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Prerequisites

- The cluster status is **Available**, and there are no ongoing tasks.
- Your CSS resource quotas are sufficient for the capacity expansion you are about to perform. You can check available resources on the **Modify Configuration** page.

Figure 4-15 Checking available resources



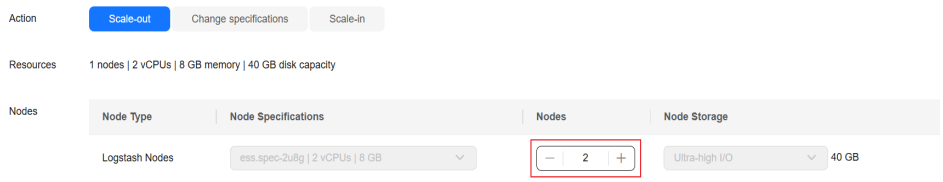
Adding New Nodes

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. Select the **Scale Cluster** tab and click **Scale out** to set parameters.

Table 4-43 Adding new nodes

Parameter	Description
Action	Select **Scale out**.
Resources	Quantities of resources added.
Nodes	Increase the number of nodes in the **Nodes** column.

Figure 4-16 Adding more nodes



5. Click **Next**.
6. Confirm the information and click **Submit**.
7. Click **Back to Cluster List** to go back to the **Clusters** page. The **Task Status** is **Scaling out**. When **Cluster Status** changes to **Available**, the cluster has been successfully scaled out.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Related Documents

You can also expand the capacity of a Logstash cluster by upgrading node specifications. For details, see **Changing Node Specifications**.

4.9.2 Reducing Nodes

If a cluster can process current traffic without fully utilizing its resources, you can scale in the cluster to cut costs.

Scaling in a Logstash cluster means to randomly remove some of its nodes, so that it is less costly to keep it running.

Scale-in process: Randomly remove cluster nodes and modify the cluster configuration.

Impact on Billing

For a pay-per-use cluster, you can see its new price when confirming the scale-in on the console. After the change is complete, the cluster will be billed based on the new price. For details, see **Cloud Search Service Price Calculator**.

Constraints

- To reduce the impact on services, you are advised to perform scale-in during off-peak hours.
- In a cross-AZ cluster, the difference between the numbers of the same-type nodes in different AZs cannot exceed 1.

Change Impact

Before the change, learn about possible impacts and operation suggestions, and develop a plan to minimize these impacts.

- Impact on cluster throughput

  Taking Logstash nodes offline usually does not interrupt services. However, jobs that are running on these will be stopped, causing the cluster throughput to decrease.
- Characteristics of this process

  Once started, a scaling task cannot be stopped until it succeeds or fails.

Scale-in Duration

The following formula can be used to estimate how long a scale-in operation will take:

Scale-in duration (min) = 5 (min) x Number of nodes to be removed

where, 5 minutes is the duration per node. It is an empirical value.

Prerequisites

The cluster status is **Available**, and there are no ongoing tasks.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

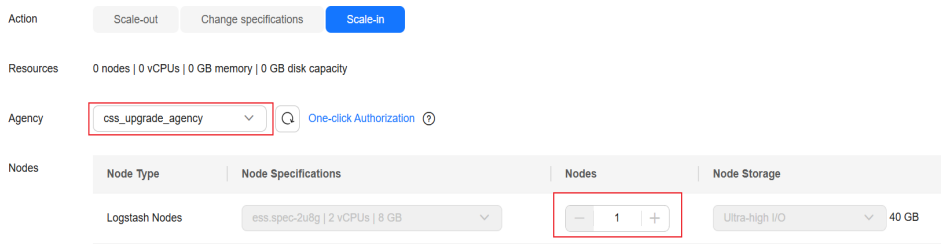
Reducing the Nodes of a Logstash Cluster

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. Select the **Scale Cluster** tab and click **Scale in** to set parameters.

Table 4-44 Removing nodes randomly

Parameter	Description
Action	Select **Scale in**.
Resources	Quantities of resources reduced.
Nodes	Reduce the number of nodes in the **Nodes** column.

Figure 4-17 Reducing nodes



5. Click **Next**.
6. Confirm the information and click **Submit**.
7. Click **Back to Cluster List** to go back to the **Clusters** page. **Task Status** is **Scaling in**. When **Cluster Status** changes to **Available**, the cluster has been successfully scaled in.

Related Documents

To optimize costs, you may also reduce the capacity of a Logstash cluster by downgrading node specifications. For details, see **Changing Node Specifications**.

4.9.3 Changing Node Specifications

If the workloads on the data plane of a Logstash cluster change, you can scale the cluster vertically by changing its node specifications.

Typically, you increase node specifications instead of decreasing them. Common scenarios include:

- In the face of fast data growth, node specifications can be increased for a Logstash cluster to increase data processing and storage capacities, thus enhancing overall performance.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- During peak hours, node specifications can be increased for a Logstash cluster to handle the extra load.

Decreasing node specifications will reduce the cluster's data processing and storage capacities. Exercise caution.

The process of changing node specifications is as follows:

1. Bring a node offline.
2. Change the node specifications.
3. Restart the node.
4. After the node recovers, proceed to another node and repeat the steps above. This goes on until all nodes are changed.

The node specifications are changed one node at a time. This is to ensure that there are sufficient resources to keep services running.

Impact on Billing

For a pay-per-use cluster, you can see its new price when confirming the node specifications or storage type change on the console. After the change is complete, the cluster will be billed based on the new price. For details, see **Cloud Search Service Price Calculator**.

Change Impact

Before changing a cluster's node specifications, it is essential to assess the potential impacts and review operational recommendations. This enables proper scheduling of the change, minimizing service interruptions.

- Impact on cluster throughput

  Taking nodes offline one at a time usually does not interrupt services. However, jobs that are running on these will be stopped, causing the cluster throughput to decrease. To minimize this impact, perform this operation when the Logstash cluster has no ongoing tasks or after temporarily suspending all its tasks.
- Characteristics of this process

  Once started, a change task cannot be stopped until it succeeds or fails.

Change Duration

The following formulas can be used to estimate how long the node specifications change will take:

Change duration (min) = 10 (min) x Number of nodes

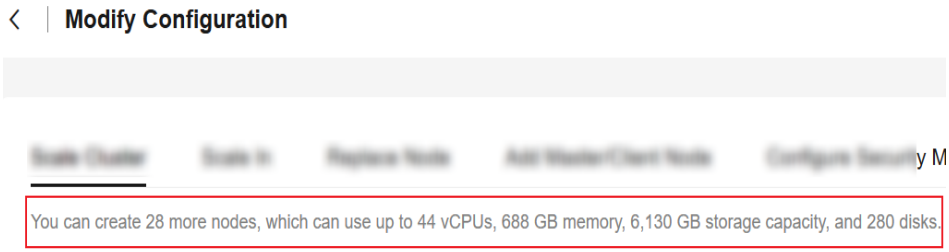
where, 10 minutes is the duration per node. It is an empirical value.

Prerequisites

- The cluster status is **Available**, and there are no ongoing tasks.
- Your CSS resource quotas are sufficient for the capacity expansion you are about to perform. You can check available resources on the **Modify Configuration** page.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 4-18 Checking available resources



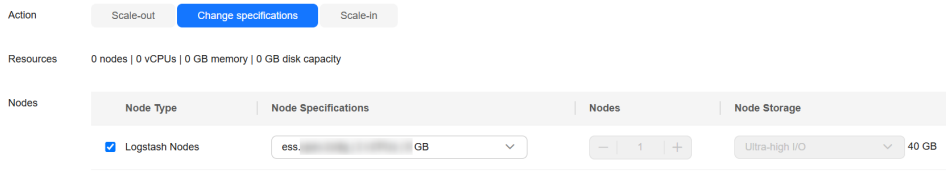
Changing Node Specifications

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters** > **Logstash**.
3. In the cluster list, find the target cluster, and choose **More** > **Modify Configuration** in the **Operation** column. The **Modify Configuration** page is displayed.
4. Select the **Scale Cluster** tab, and click **Change specifications** to set parameters.

Table 4-45 Changing node specifications

Parameter	Description
Action	Select **Change specifications**.
Resources	Shows the change of resources for this operation.
Nodes	Configure the changes you want to make. 1. Select **Logstash Nodes** in the **Node Type** column. 2. In the **Node Specifications** column, select a new node flavor.

Figure 4-19 Changing cluster specifications



5. Click **Next**.
6. Confirm the information and click **Submit**.
7. In the dialog box that is displayed, choose whether to select **Cluster status check**, and click **OK** to start the specifications change.

   When **Cluster status check** is selected: During a node specifications change, the cluster status is checked by default to improve the success rate of the change and ensure data security. The nodes are changed one at a time. For each node, the system changes its specifications, restarts it, and confirms all processes have started successfully before proceeding to the next node.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

In emergencies (for example, when a cluster is overloaded and services are faulty, which may prevent a specifications change request from being delivered), you can skip cluster status check to free up resources for cluster recovery. However, doing so may cause the cluster to become faulty and interrupt services. Exercise caution.

8. Click **Back to Cluster List** to go back to the **Clusters** page. The **Cluster Status** is **Configuration modified**. When **Cluster Status** changes to **Available**, the cluster node specifications have been successfully changed.

4.10 Managing Clusters

4.10.1 Viewing Cluster Details

On the CSS management console, you can check a Logstash cluster's status, version, nodes, configurations and settings, and resource allocation on both the cluster list and the cluster overview page.

Introduction to the Cluster List

A cluster list displays all clusters of the same type under the current account. If there are a large number of clusters, the cluster list will consist of multiple pages, showing clusters of all states.

Clusters are listed in chronological order by default in the cluster list, with the most recent clusters displayed at the top.

You can click **Export** to download the full cluster list.

Table 4-46 Cluster list parameters

Parameter	Description
Name/ID	Name and ID of a cluster. The cluster ID is automatically generated by the system and uniquely identifies a cluster.
Cluster Status	Current status of a cluster. • Available: The cluster is running properly. • Creating: The cluster is being created. • Processing: The cluster is being restarted, scaled out, scaled in, or its node specifications are being changed. • Abnormal: The cluster failed to be created or is unavailable. **WARNING** A cluster that is unavailable can be deleted, but cannot be scaled out. For such a cluster, you can check its monitoring data or restart it to try to fix the problem. If the fault persists, contact technical support.
Task Status	The status of tasks such as cluster restarting, scaling out, and scaling in, or ongoing pipeline tasks.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Version	Cluster version and image version. For example, if the version is **7.10.0(7.10.0_x.x.x_x.x.x)**, **7.10.0** is the cluster version and **7.10.0_x.x.x_x.x.x** enclosed in parentheses is the image version. **CAUTION** For a cluster created a long time ago, its image version may not be available because it is likely not recorded in the database.
Private Network Address	Private network address and port number of a cluster. You can use it to access the cluster. If the cluster has multiple nodes, the private network addresses and port numbers of all nodes are displayed.
Enterprise Project	Enterprise project that a cluster belongs to.
Tags	Displays cluster tags. You can click **Edit Tag** to add, delete, or modify cluster tags.
Cluster Description	Provides supplementary information about the cluster.
Created	Cluster creation time.
Billing Mode	Billing mode of a cluster.
Operation	Operations that can be performed on a cluster, such as checking metrics, restart, and deletion. If an operation is unavailable, the menu button is grayed out.

Viewing Basic Cluster Information

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters** > **Logstash**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, check the cluster's basic information, node configuration, and network information.

Table 4-47 Basic cluster information

Parameter	Description
Cluster ID	Unique ID of a cluster, which is automatically generated by the system. Each cluster ID is unique within the same region.
Cluster Name	Cluster name. You can click  to change the cluster name.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Cluster Version	Cluster version. For example, if the version is **7.10.0(7.10.0_x.x.x_x.x.x)**, **7.10.0** is the cluster version and **7.10.0_x.x.x_x.x.x** enclosed in parentheses is the image version. **CAUTION** For a cluster created a long time ago, its image version may not be available because it is likely not recorded in the database.
Cluster Status	Current status of a cluster
Task Status	Current task status of a cluster. If there is no ongoing task, **--** is displayed.
Region | AZ	The region and AZ where the cluster is located.
Enterprise Project	Enterprise project that a cluster belongs to. You can click a project name to check the basic project information.
Cluster Description	Provides supplementary information about the cluster. You can click to modify the description.

Table 4-48 Node information

Parameter	Description
Logstash Node Specifications	Specifications of nodes in a Logstash cluster
Logstash Nodes	Number of nodes in a Logstash cluster
Logstash Node Storage Type	The storage type of Logstash nodes in the cluster.
Logstash Node Capacity (GB)	Storage capacity per node in the cluster.
Cluster Storage Capacity (GB)	Total storage capacity of the cluster. • Quota: the upper limit of the storage capacity allocated to the cluster. • Used: storage space already used by the cluster. The cluster's disk usage can be calculated as follows: Disk usage = (Used/Quota) x 100%

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Node Details	View the details of all cluster nodes. Click **View**. In the displayed dialog box, view node details, including the node name/ID, status, and IP address.

Table 4-49 Network information

Parameter	Description
VPC	VPC to which a cluster belongs
Subnet	Subnet to which the cluster belongs
Security Group	Security group to which a cluster belongs. To modify the security group of a cluster, click **Change Security Group**. **WARNING** The selected security group must allow port 9600 in the inbound direction. Otherwise, the cluster may be inaccessible to external services.
Private IPv4 Address	Private IPv4 address and port number of a cluster. If the cluster has only one node, the IP address and port number of only one node are displayed, for example, **10.62.179.32:9600**. If the cluster has multiple nodes, the IP addresses and port numbers of all nodes are displayed, for example, **10.62.179.32:9600,10.62.179.33:9600**.

Table 4-50 Additional information

Parameter	Description
Billing Mode	Billing mode of a cluster.
Created	When a cluster was successfully created.

4.10.2 Managing Cluster Tags

As business grows, an enterprise may operate hundreds of Logstash clusters, and cluster names alone will be insufficient for categorization and identification. For example, tasks such as resource monitoring or automated operations via scripts require the ability to quickly locate clusters based on specific attributes. On CSS, you can use tags (key-value pairs) to label resources, enabling efficient filtering and batch management. When used together with Tag Management Service (TMS), tags help you implement enterprise-grade resource governance.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Constraints

- Each cluster can have a maximum of 20 tags.
- If your organization has configured tag policies for CSS, add tags to clusters based on these policies. If a tag does not comply with existing tag policies, cluster creation may fail. Contact the administrator to learn more about these policies.
- To use the same tag to identify multiple cloud resources for better resource grouping, predefine tags in Tag Management Service (TMS). For details, see **Predefined Tags Overview**.

Managing Cluster Tags in Batches

Manage tags for multiple clusters in batches, including add, modify, and delete. This feature is useful in cases such as environment refactoring and large-scale migrations.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. In the cluster list, select one or more clusters.
4. Click **Edit Tag** above the cluster list.
   - Add

     In the displayed dialog box, click **+Add**, and add new tags in the form of key-value pairs. The tags will be added for all selected clusters.
   - Modify

     In the displayed dialog box, modify existing tag values. The changes will be applied to all selected clusters, overwriting existing tags. Avoid this operation if you have custom tags that you wish to keep.
   - Delete

     In the displayed dialog box, click **Delete** next to the tag you want to delete. This deletes the tag for all selected clusters. The deletion cannot be undone.
5. Click **OK** to save the changes.
6. Verify the result: In the cluster list, hover the mouse over the icon in the **Tag** column for each cluster. Check that the tag information has been updated for all selected clusters.

Managing Tags for a Single Cluster

Add, modify, or delete tags for a single cluster.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Tags** tab.
5. Click **Edit Tag**. In the displayed dialog box, modify the cluster's tags.
   - Add

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

In the displayed dialog box, click **+Add**, and add new tags in the form of key-value pairs.

- Modify

  In the displayed dialog box, modify existing tag keys and values.

- Delete

  In the displayed dialog box, click **Delete** next to the tag you want to delete.

6. Click **OK**.
7. Verify the result: In the tag list, check that the tag keys and values have been updated.

Searching for Clusters by Tag

Locate clusters by tag from among a large number of resources.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. Click the search box above the cluster list. Select a tag key, operator, and tag value to search.
   - You can select a tag key and value from the drop-down list. The system returns a list of clusters that exactly match the tag key and value.
   - In case you use multiple tags to start a search: With the same tag key but different values, the system returns a list of clusters that match any of the tag values; If the tag keys are different, the system returns a list of clusters that match all of these keys and tags.

   The cluster list is filtered in real time to display matching clusters only.

4.10.3 Associating a Cluster with an Enterprise Project

In practice, it is common for multiple departments (such as R&D and quality management) within a large organization to share a single Huawei Cloud account. Enterprise projects let you manage Logstash clusters by department or project, enabling clearer resource visibility and more fine-grained access control across regions within the shared account.

How the Feature Works

Enterprise projects are a cloud resource management feature provided by Huawei Cloud that allow you to group cloud resources by project, department, or business. For more information, see **Enterprise Project Management Service**.

A newly created cluster belongs to the **default** project by default. To move a cluster from the default project to a custom project, you simply perform a resource migration. There is no need to re-create the cluster.

Here, resource migration is the process of moving a CSS cluster from one enterprise project (for example, **default**) to another (for example, **TEST**).

- Impact: Resource migration only changes the management metadata. It does not restart the cluster or interrupt services.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- Permission control change: After the migration, only users who have been granted permissions on the new project (or all resources) can manage the cluster.

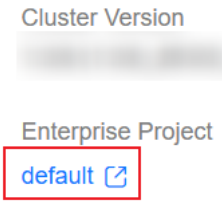
Prerequisites

- Enterprise Project Management Service (EPS) has been enabled for your account. For details, see section "Enabling Enterprise Project" in *Enterprise Management User Guide*Enabling Enterprise Project.
- You have created the destination enterprise project (for example, **TEST**). For details, see section "Creating an Enterprise Project" in *Enterprise Management User Guide***Creating an Enterprise Project**.

Migrating a Cluster Between Enterprise Projects

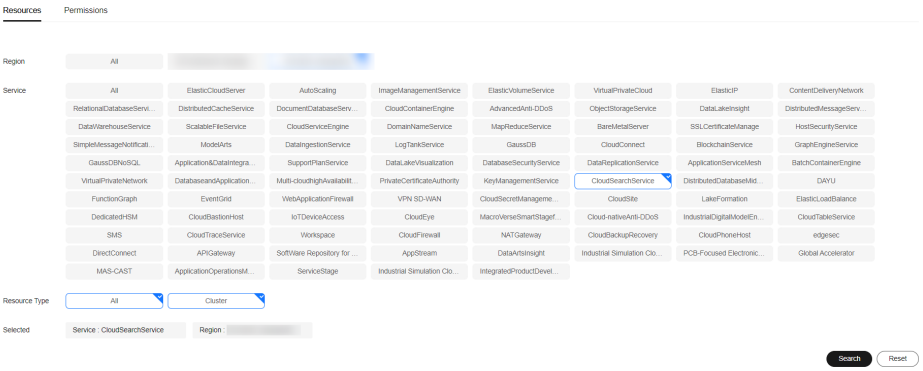
1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. On the **Overview** tab, find **Enterprise Project** in the **Basic Information** area, and click the enterprise project name under it. The **Project Management** console is displayed.

   Figure 4-20 Going to the Project Management console

5. On the **Resources** tab on the Project Management console, select the region of the current cluster, and select **CloudSearchService** as **Service**. Then click **Search**. The CSS clusters are displayed in the resource list below.

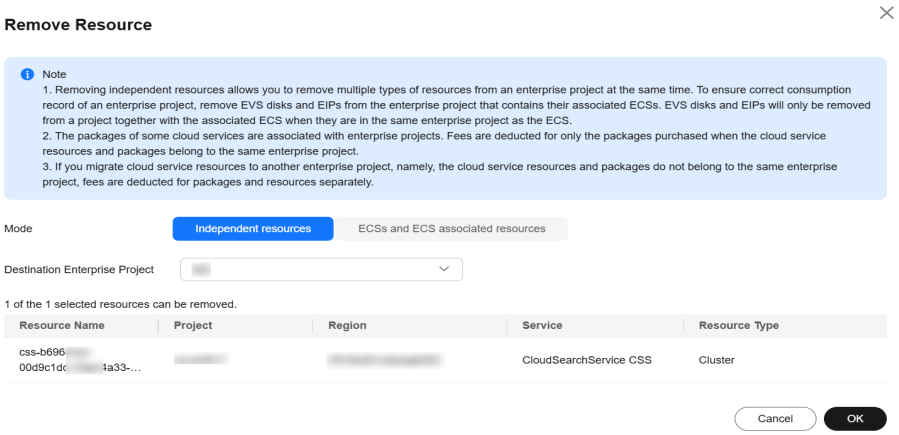
   Figure 4-21 Filtering CSS clusters

6. Select the cluster whose enterprise project you want to modify and click **Remove**.
7. In the **Remove Resource** dialog box, specify **Mode** and select **Destination Enterprise Project**. Confirm the change information, and click **OK**.

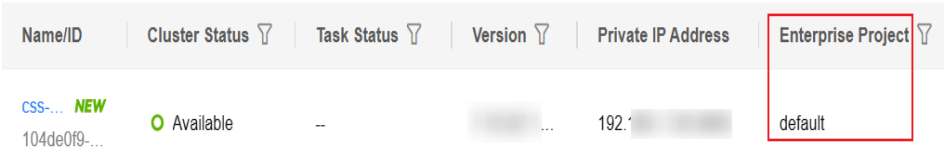
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 4-22 Remove Resource



8. After the change is complete, return to the Logstash cluster list on the CSS console and check the cluster's new enterprise project to confirm the operation result.

Figure 4-23 Enterprise Project



FAQ

- **Question 1: Does changing the enterprise project change a cluster's IP address?**

  Changing a cluster's enterprise project does not change its IP address. It merely changes the logical ownership of cloud resources. There is no change at the network layer.

- **Question 2: What should I do if, after a cluster's enterprise project is changed, the cluster is hidden from users under the original project?**

  After the change, the cluster management permissions are transferred to the new project. Check whether your account has the read permission on the new (destination) enterprise project.

4.10.4 Forcibly Restarting a Cluster

When a Logstash cluster becomes unavailable due to extended runtime or unknown faults such as memory overflow or process failures, regular configuration updates may not be sufficient to restore it. In such cases, CSS offers a forcible restart option that recovers the cluster by directly restarting its underlying VMs.

Impact

Forcibly restarting a Logstash cluster stops all Logstash processes, interrupts all ongoing data streams, and makes the cluster totally unavailable during the restart.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

The keepalive setting determines each pipeline's status after the restart completes. Table 4-51 describes how pipelines recover after the restart, depending on the keepalive setting.

Table 4-51 Pipeline recovery

Keepalive	Behavior Upon Restart	Recovery Process
No	Stopped	1. During the restart, all pipeline tasks that are **Running** are forcibly **Stopped**. 2. After the restart completes, manually start the pipelines.
Yes	Attempt to recover automatically	1. When the restart begins, the pipeline status is set to **Recovering**. 2. The system attempts to restart the Logstash processes. 3. If the Logstash processes are started within 10 minutes, the pipelines are automatically restored to **Working**. 4. Otherwise, the pipeline status changes to **Failed**, and you need to manually start them.

The total restart time depends on how fast the underlying VMs restart. Usually, it takes longer than how long the Logstash processes restart.

Constraints

Clusters whose **Task Status** is **Frozen** or that have ongoing tasks cannot be restarted.

Before You Start

- Check the target cluster's health status. Check the **Task Status** column to ensure there are no ongoing tasks.
- Perform this operation during off-peak hours, or ensure the upstream data source (such as Kafka) has sufficient buffering capacity to hold data until consumption resumes after the restart.

Forcibly Restarting a Cluster

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. In the cluster list, find the target cluster, and choose **More > Force Restart** in the **Operation** column.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

4. In the **Force Restart Cluster VMs** dialog box, confirm the restart information, type in **YES** for confirmation, and click **OK**.
5. When the cluster status changes from **Processing** to **Available**, the cluster has restarted successfully.

4.10.5 Deleting a Cluster

When a Logstash cluster is no longer required, you can delete the cluster to release compute and storage resources and stop billing. Cluster deletion is an irreversible high-risk operation that permanently removes all data in the cluster. Before deleting a cluster, ensure that all the required data has been backed up.

Before You Start

Cluster deletion is an irreversible high-risk operation. Before proceeding, confirm the following:

- **Data has been backed up**: Important cluster configuration files have been exported and backed up to a local PC. Without backups, the data will be permanently lost once deleted.
- **There are no ongoing tasks**: Check the Logstash pipeline monitoring metrics and ensure that the cluster has no ongoing tasks.

Deleting a Pay-per-Use Cluster

The cluster is deleted immediately, resources are released, and pay-per-use billing stops.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters > Logstash**.
3. In the cluster list, find the target cluster, and choose **More > Delete** in the **Operation** column.
4. In the displayed dialog box, read the risk warning carefully and enter **DELETE** in the text box.
5. Click **OK**.

   After the cluster is deleted, it is removed from the cluster list.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

5 Vector Database

5.1 CSS Vector Engine vs. OpenSearch Vector Engine

To enable efficient retrieval of massive high-dimensional vector data for AI search systems, CSS provides two vector search engines: the CSS vector engine built on proprietary technologies, and the OpenSearch vector engine, which is fully compatible with OpenSearch community standards. This topic compares the two engines in terms of features, technical capabilities, APIs, and query syntax, helping you choose the most suitable engine for your needs.

Features

Table 5-1 A comparison of features

Dimension	CSS Vector Engine	OpenSearch Vector Engine
Positioning	A proprietary vector search engine, built on a deeply customized Elasticsearch/ OpenSearch core, delivering high-performance, enterprise-grade vector search capabilities.	An open-source vector search engine provided by the OpenSearch community. Driven by the k-NN plugin, it leverages open-source libraries such as Faiss, NMSLib, and Lucene, and is fully compatible with the APIs and standards of the OpenSearch ecosystem.
Syntax	Provides proprietary indexing APIs and a custom query DSL.	Uses OpenSearch's k-NN (K-Nearest Neighbors) or ANN (Approximate Nearest Neighbor) syntax. ANN is recommended.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

When to use	To enjoy the high performance, stability, and out-of-the-box experience that CSS delivers	To maintain full compatibility with the open-source community
Supported cluster versions	• Elasticsearch: 7.6.2 and 7.10.2 • OpenSearch: 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later)	All OpenSearch versions
Details	**Procedure**	**Vector search**

Similarities

- Algorithm support: Both engines support vector indexing algorithms such as HNSW and IVF, as well as quantization algorithms such as PQ and SQ.
- Query support: Both engines support vector query, pre-filtering query, and hybrid query.
- Performance requirements: Both engines implement core computational logic in C++ and require large memory configurations to deliver optimal performance.

Differences in Core Capabilities

Table 5-2 A comparison of core capabilities

Dimension	CSS Vector Engine	OpenSearch Vector Engine
Level of integration	Deep integration with CSS: eligible for full SLA assurance and technical support.	Runs as an OpenSearch plugin (k-NN plugin) and relies on community version release cycles for updates and maintenance.
Ease of use	Proprietary APIs, not directly compatible with open-source APIs.	Open-source OpenSearch ANN/k-NN query DSL and APIs.
Flexibility	CSS provides algorithm optimization policies, with relatively centralized optimization configuration options.	Open-source libraries such as Faiss and NMSLib can be flexibly combined, enabling more fine-grained parameter tuning.
Performance optimization	Supports dedicated index memory management, with simplified optimization and tuning.	Relies on the optimization of Faiss/NMSLib parameters and OpenSearch configurations.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Differences in APIs and Query Syntax

Table 5-3 A comparison of APIs and query syntax for key operations

Dimension	CSS Vector Engine	OpenSearch Vector Engine

Creating indexes	A proprietary vector index is created, and key parameters **index.vector** and **type** are required.
	Example:
	PUT /my_index
	{
	"settings": {
	"index": {
	"vector": true
	}
	},
	"mappings": {
	"properties": {
	"image_vector": {
	"type": "vector",
	"dimension": 2,
	"indexing": true,
	"metric": "euclidean"
	},
	"price":{"type": "float"}
	}
	}
	}
		An ANN index is created, and key parameters **index.knn** and **type** are required.
		Example (OpenSearch 2.x or later):
		PUT /my_index
		{
		"settings": {
		"index": {
		"knn": true
		}
		},
		"mappings": {
		"properties": {
		"image_vector": {
		"type": "knn_vector",
		"dimension": 2,
		"method": {
		"name": "hnsw",
		"space_type": "l2",
		"engine": "faiss"}
		},
		"price":{"type": "float"}
		}
		}
		}

Query DSL	Perform a vector query, using the key parameter **query.vector**.
	Example:
	GET /my_index/_search
	{
	"query": {
	"vector": {
	"image_vector": {
	"vector": [1.0,2.0],
	"topk": 3,
	"filter": {"range": {"price": {"lte": 300}}}
	}
	}
	}
	}
		Perform a vector query, using the key parameter **query.knn**.
		Example (OpenSearch 2.x or later):
		GET /my_index/_search
		{
		"query": {
		"knn": {
		"image_vector": {
		"vector": [1.0, 2.0],
		"k": 3,
		"filter": {"range":{"price":{"lte": 300 }}}
		}
		}
		}
		}

5.2 CSS Vector Database

5.2.1 Procedure

In the age of AI, unstructured data, such as images, videos, audios, and text, is growing rapidly. Traditional keyword-based search cannot effectively handle unstructured data because it fails to capture deep semantic or visual features. The

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

CSS vector database addresses this challenge. Designed for storing, managing, and retrieving high-dimensional vectors, it transforms unstructured data into vectors and leverages techniques like approximate nearest neighbor (ANN) search to enable high-throughput, high-accuracy similarity search.

Use Cases

The CSS vector database supports a diverse range of use cases:

- Reverse image/video search: Upload an image or video clip to quickly retrieve visually similar images or videos from massive datasets.
- Similar product recommendations: Based on the item you are currently viewing, the system intelligently recommends other products with similar appearance, functionality, or attributes, improving user experience and conversion rates.
- Semantic text search: Unlike traditional keyword search, the vector database understands the semantic meaning of text and returns results more aligned with your query intent, even if the text does not contain exact keyword matches.
- Cross-modal search: The vector database enables seamless search across different data modalities. For example, you can enter a textual description to search for relevant images or videos.
- Retrieval-augmented generation (RAG): In an LLM-powered Q&A system, the vector database serves as an external knowledge base for the LLM, supporting efficient retrieval of domain-specific knowledge. This ensures the model generates answers that are more accurate, more contextually relevant, and more firmly grounded in domain-specific knowledge.

Advantages

The CSS vector database delivers the following key advantages:

- High performance: The built-in CSS vector search engine can search through tens to hundreds of millions of vectors in milliseconds. The CSS vector database supports all major indexing algorithms, including brute-force search, graph-based algorithms (such as HNSW), and Inverted File Product Quantization (IVF-PQ). This allows you to flexibly select algorithms based on precision and latency requirements.
- High reliability: Based on the native distributed architecture of Elasticsearch/ OpenSearch, CSS delivers enterprise-grade reliability guarantees, including multi-replica, snapshots, and permission control, ensuring zero data loss.
- Extensive ecosystem compatibility: The CSS vector database is fully compatible with native Elasticsearch/OpenSearch query language, so there is zero learning curve for developers. It supports mainstream client languages such as Python, Java, and Go, and can seamlessly integrate components such as Kibana, Logstash, and Dashboards.
- Security: The CSS vector database provides comprehensive security features, including network isolation, access control, and data encryption.

How It Works

The CSS vector database uses approximate nearest neighbor (ANN) search to mitigate the intensive computational load of exact k-NN search. While exact k-NN

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

guarantees recall by exhaustively comparing the query vector against all data points, it incurs prohibitive latency on large-scale datasets. ANN significantly reduces computational overhead and search latency, balancing high recall with substantial gains in throughput and efficiency. Key points include:

- Reducing the candidate set: Traditional text search filters out irrelevant documents through inverted indexes, while vector search quickly retrieves potentially relevant vectors using index structures like HNSW graphs or IVF-PQ, avoiding scanning the full dataset. For example, HNSW uses a multi-layer structure to quickly find nearest neighbors to the query vector.
- Reducing computational complexity: The funnel model first performs coarse quantization (such as IVF-PQ) on vectors to quickly obtain the candidate set. Then, it performs fine retrieval or reranking (such as cosine similarity) on the candidate set. Quantization uses the product quantization (PQ) algorithm to encode high-dimensional vectors into compact codes, reducing storage and computational overheads.
- Balancing performance and accuracy: Indexing parameters (such as the number of HNSW layers and the number of IVF clusters) can be dynamically tuned to trade off between recall and query latency.

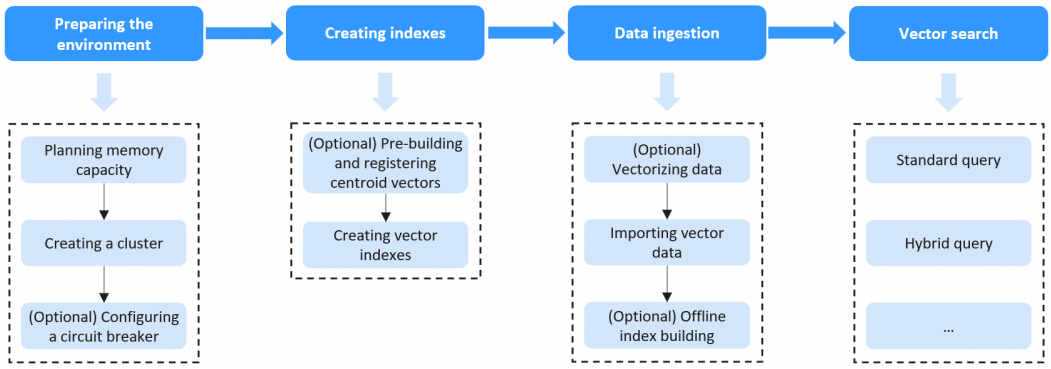
Supported Cluster Versions

Table 5-4 Clusters that support CSS Vector Database

Cluster Type	Cluster Version
Elasticsearch	7.6.2, 7.10.2
OpenSearch	1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later)

Procedure

Figure 5-1 Procedure for using the CSS vector database



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 5-5 Procedure for using the CSS vector database

Category	Operation		Details
Use	Environment preparation	Create a proper CSS vector database based on your memory planning.	**Preparing the Environment**
	Creating indexes	Create vector indexes in your cluster and define mappings that contain vector fields, including vector dimensions, indexing algorithm, and similarity measurement methods.	**Creating a Vector Index**
	Data writing	Write vectors (along with the original data or metadata) into indexes.	**Importing Vector Data**
	Vector search	Use the standard Elasticsearch query DSL to provide the query vector and specify the number (k) of nearest neighbors you expect to return as well as the similarity score.	**Performing Vector Search**
Advanced features	Nested fields	Nested fields allow multiple independent vectors to be stored within a single document. The document is returned as a match if the query vector meets the similarity threshold for any of its nested vectors.	**Implementing Nested Vector Search**
Management	Performance tuning	Provides ways to optimize the write and query performance of the CSS vector database.	**Optimizing Query and Write Performance**
	Cache management	Flexibly monitors and tunes the usage of the vector index cache to ensure stable query performance.	**Managing the Vector Search Cache**

Related Documents

- A quick start guide to CSS vector database: **Using Elasticsearch for Vector Search**.
- Performance testing for CSS vector database: **Testing the Performance of CSS's Elasticsearch Vector Search**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

5.2.2 Preparing the Environment

During environment preparation for a CSS vector database, estimate memory requirements based on vector dimensions, data volume, and the selected indexing algorithm (such as HNSW or PQ); and configure an appropriate circuit breaker threshold. They help build a vector search environment that delivers both high performance and enterprise-grade reliability.

Planning Memory Capacity

On each node of a CSS vector database, the physical memory is divided into two logical areas:

- JVM heap memory: used for service processing, such as index metadata mapping, aggregation queries, and cluster management. By default, this accounts for 50% of the physical memory, with an upper limit of 31 GB.
- Off-heap memory: used for vector computations. Graph index structures that require resident memory are directly loaded into off-heap memory to enable high-speed parallel computation.

Before selecting cluster node specifications, calculate the required off-heap memory based on the selected indexing algorithm. **Table 5-6** provides the formulas for estimating the off-heap memory required by a single index replica for each indexing algorithm.

Table 5-6 Estimating off-heap memory per indexing algorithm

Algorithm	Off-Heap Memory Formula (Unit: Bytes)	Algorithm Description
FLAT	mem_size = dim * dim_size * num + delta	Full brute-force indexing. Suitable for: Datasets with less than 10,000 records, or scenarios requiring maximum recall accuracy.
GRAPH	mem_size = (dim * dim_size + neighbors * 4) * num + delta	Hierarchical Navigable Small Worlds (HNSW), a graph-based indexing algorithm for efficient approximate nearest neighbor (ANN) search. Suitable for: Datasets with up to 100 million records, millisecond-level latency and high precision required.
GRAPH_PQ	mem_size = (fragment_num + neighbors * 4) * num + delta	A combination of HNSW with product quantization (PQ). Suitable for: Datasets with up to a billion records.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Algorithm	Off-Heap Memory Formula (Unit: Bytes)	Algorithm Description
GRAPH_SQ8	mem_size = (dim + neighbors * 4) * num + delta	A combination of HNSW with 8-bit scalar quantization. Suitable for: Datasets with up to a billion records.
GRAPH_SQ4	mem_size = (dim / 2 + neighbors * 4) * num + delta	A combination of HNSW with 4-bit scalar quantization. Suitable for: Datasets with up to a billion records.
IVF_GRAPH	Relies on a pre-built centroid vector dictionary; no resident memory required.	A combination of Inverted File Index (IVF) with HNSW. Suitable for: Use cases that demand high write performance.
IVF_GRAPH_PQ	Relies on a pre-built centroid vector dictionary; no resident memory required.	A combination of IVF, HNSW, and PQ. Suitable for: Use cases that demand high write throughput.

Parameter description:

- **mem_size**: The size of off-heap memory required by vector indexes. (This estimate assumes there are no replicas. If index replicas are involved, the required memory size multiplies with the number of replicas.)
- **dim**: Number of vector dimensions.
- **dim_size**: Number of bytes required by each dimension. By default, each dimension is a 4-byte float value.
- **num**: Total number of vectors.
- **delta**: Metadata size. This factor can be ignored.
- **neighbors**: Number of neighbors for each vector in a graph index. The default value is 64.
- **fragment_num**: Number of segments a vector is split into when product quantization is used.

  If fragment_num is not configured during index creation, the value is determined by **dim**.

```
if dim <= 256:
    fragment_num = dim / 4
elif dim <= 512:
    fragment_num = dim / 8
else:
    fragment_num = 64
```

Example

Scenario: SIFT10M dataset (128-dimensional vectors and 10 million records), GRAPH algorithm, 64 neighbors, one replica.

1. Off-heap memory for a single replica: mem_size = (128 * 4 + 64 * 4) x 10,000,000 + delta ≈ 7.7 GB

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. Off-heap memory for two replicas: 7.7 x 2 = 15.4 GB
3. Safety guarantee considering the circuit breaker (assume the default circuit breaker threshold is 80%): 15.4/80% = 19.3 GB
4. Final node specifications: The JVM heap memory accounts for 50% of the physical memory (no more than 31 GB). Therefore, you should select a node with 64 GB memory (for example, 8U64G) or two nodes each with 32 GB memory (for example, 8U32G).

Creating a Cluster

The CSS vector database is based on Elasticsearch or OpenSearch clusters. Create an Elasticsearch or OpenSearch cluster based on service requirements. **Table 5-7** describes the key parameters.

Table 5-7 Key parameters for creating a cluster

Cluster Type	Elasticsearch	OpenSearch
Cluster Version	Select **7.6.2** or **7.10.2**. Other versions do not support the CSS vector search engine. **7.10.2** is recommended, because it provides more comprehensive features.	Select **1.3.6**, **2.19.0**, or 3.4.0 (image version 3.4.0_26.6.0_xxx or later). Other versions do not support the CSS vector search engine. **2.19.0** and **3.4.0** are recommended, because they provide more comprehensive features.
CPU Architecture	**x86** is recommended.	**x86** is recommended.
Node Specifications	You are advised to select a **Memory-optimized** flavor. Flavors of this type have a high memory-to-CPU ratio, making them ideal for vector computations. The cluster's physical memory must meet the requirements described in **Planning Memory Capacity**.	You are advised to select a **Memory-optimized** flavor. Flavors of this type have a high memory-to-CPU ratio, making them ideal for vector computations. The cluster's physical memory must meet the requirements described in **Planning Memory Capacity**.
Documentation	**Creating a Cluster**.	**Creating a Cluster**.

(Optional) Configuring the Circuit Breaker

To mitigate out-of-memory (OOM) errors and maintain optimal vector query performance, a circuit breaker mechanism is employed. When the cluster's off-heap memory usage exceeds a predefined threshold, this mechanism automatically blocks vector data writes to the cluster.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

The purposes of this mechanism are as follows:

- Preventing memory overload: Write throttling lowers off-heap memory usage.
- Maintaining query performance: Optimal vector query performance can be maintained by preventing memory overload.

The off-heap memory circuit breaker is enabled by default. You can enable or disable it and adjust its threshold based on service requirements.

Run the following command on Dev Tools in Kibana or OpenSearch Dashboards:

```
PUT _cluster/settings
{
  "persistent": {
    "native.cache.circuit_breaker.enabled": "true",
    "native.cache.circuit_breaker.cpu.limit": "80%"
  }
}
```

Table 5-8 Circuit breaker parameters

Parameter	Type	Default Value	Description
native.cache.circuit_breaker.enabled	Boolean	true	Whether to enable the off-heap memory circuit breaker. The value can be: • **true**: Enable the off-heap memory circuit breaker. When the off-heap memory usage reaches the circuit breaker threshold, write requests are blocked. • **false**: Disable the off-heap memory circuit breaker. OOM errors may occur in case of excessive off-heap memory usage.
native.cache.circuit_breaker.cpu.limit	String	80%	Circuit breaker threshold in terms of maximum off-heap memory usage. This parameter is available only when **native.cache.circuit_breaker.enabled=true**. Value range: a value in percentage Assume a cluster uses 128 GB memory. The required heap memory is 31 GB, and the default circuit breaker threshold is 80%, then: (128 – 31) x 80% = 77.6 GB. This means when the off-heap memory usage exceeds 77.6 GB, the circuit breaker is triggered to block write operations.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

5.2.3 Creating a Vector Index

A vector index can be created by defining the vector dimensions, indexing algorithm (such as HNSW and IVF), and similarity measurement method (for example, cosine distance and Euclidean distance) in the mapping, establishing an optimal structure for indexing feature-rich data. Vector search mitigates the curse of dimensionality in large-scale data processing. Leveraging techniques such as quantization-based compression and multi-layer graph navigation, it delivers high recall accuracy and low latency even in datasets with tens of millions or even billions of records.

Indexing Algorithms

Table 5-9 lists the indexing algorithms supported by CSS vector databases. Select an algorithm based on your service requirements.

Table 5-9 Indexing algorithms for CSS vector databases

Algorithm	Description	When to Use	Cluster Version
FLAT	Full brute-force indexing. This method does not build a complex data structure; instead, it performs an exhaustive search by calculating the distance between the query vector and every vector in the database in a sequential manner. It achieves a 100% recall rate (zero precision loss), but the computational load increases linearly as data volume grows.	Datasets with less than 10,000 records, or scenarios requiring maximum recall accuracy	Elasticsearch: 7.6.2 or 7.10.2 OpenSearch: 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later)
GRAPH	Hierarchical Navigable Small Worlds (HNSW), a graph-based indexing algorithm for efficient approximate nearest neighbor (ANN) search. A multi-layer graph navigation structure accelerates retrieval speeds. It delivers extremely fast retrieval speeds with high recall accuracy, but it is also highly memory-intensive (resident memory is required).	Datasets with up to 100 million records, millisecond-level latency and high precision required.	Elasticsearch: 7.6.2 or 7.10.2 OpenSearch: 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later)

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Algorithm	Description	When to Use	Cluster Version
GRAPH_PQ	A combination of HNSW with product quantization (PQ). By splitting and encoding vectors, this approach significantly reduces storage and memory overheads. The compression rate can reach 1/16 or even higher. It offers a low memory footprint, though precision decreases as the compression ratio increases.	Datasets with up to a billion records	Elasticsearch: 7.6.2 or 7.10.2 OpenSearch: 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later)
GRAPH_SQ8	A combination of HNSW with 8-bit scalar quantization. 32-bit floating-point numbers are compressed into 8-bit integers. The compression rate is 1/4. The memory footprint is significantly reduced, with a small precision loss.	Datasets with up to a billion records	Elasticsearch: 7.10.2 OpenSearch: 2.19.0 and 3.4.0 (image version 3.4.0_26.6.0_xxx or later)
GRAPH_SQ4	A combination of HNSW with 4-bit scalar quantization. 32-bit floating-point numbers are compressed into 4-bit integers. The compression rate is 1/8. The memory footprint is significantly reduced, but recall also decreases significantly. GRAPH_SQ4 is more computationally efficient than GRAPH_SQ8.	Datasets with up to a billion records	Elasticsearch: 7.10.2 OpenSearch: 2.19.0 and 3.4.0 (image version 3.4.0_26.6.0_xxx or later)
IVF_GRAPH	A combination of Inverted File Index (IVF) with HNSW. The vector space is partitioned into multiple clustering subspaces, each represented by a centroid. During retrieval, only relevant subspaces are scanned. This significantly accelerates retrieval speeds but leads to a slight precision loss.	High write throughput is required, and the operational complexity associated with centroid pre-building is acceptable.	Elasticsearch: 7.6.2 or 7.10.2 OpenSearch: 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later)

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Algorithm	Description	When to Use	Cluster Version
IVF_GRAPH_PQ	A combination of IVF, HNSW, and PQ. System capacity is further increased through compression, and system overheads are reduced.	High write throughput is required, and the operational complexity associated with centroid pre-building is acceptable.	Elasticsearch: 7.6.2 or 7.10.2 OpenSearch: 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later)

Constraints

- When creating a vector index, do not use special characters **\/*?"<>|,** or spaces in the vector field name. Use a valid name such as **my_vector** instead. This helps prevent failures when backing up and restoring the vector index.
- If you select **IVF_GRAPH** or **IVF_GRAPH_PQ**, you need to perform **(Optional) Pre-Building and Registering Centroid Vectors** before creating a vector index. When restoring an IVF index, you must first restore and register its dict index (centroid vectors) before restoring the IVF index itself.

Logging in to Dev Tools

Log into Dev Tools to run DSL commands.

- **For an Elasticsearch cluster, log in to Kibana**
  a. Log in to the **CSS management console**.
  b. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
  c. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
  d. In the left navigation pane, choose **Dev Tools**.

     The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.
- **For an OpenSearch cluster, log in to Dashboards**
  a. Log in to the **CSS management console**.
  b. In the navigation pane on the left, choose **Clusters > OpenSearch**.
  c. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
  d. In the left navigation pane, choose **Dev Tools**.

     The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Creating a Vector Index

Define the index mappings and specify the indexing algorithm parameters.

For example, create an index named **my_index**. This index contains a vector field named **my_vector** and a text field named **my_label**. A graph-based index is created for the vector field, and Euclidean distance is used for similarity measurement.

```
PUT my_index
{
  "settings": {
    "index": {
      "vector": true,
      "number_of_shards": 1,
      "number_of_replicas": 1
    }
  },
  "mappings": {
    "properties": {
      "my_vector": {
        "type": "vector",
        "dimension": 2,
        "indexing": true,
        "algorithm": "GRAPH",
        "metric": "euclidean"
      },
      "my_label": {
        "type": "keyword"
      }
    }
  }
}
```

Table 5-10 settings parameters

Parameter	Mandatory	Type	Default Value	Description
index.vector	Yes	Boolean	N/A	Whether to enable vector indexes. Set this parameter to **true**. Otherwise, vector indexes cannot be created.
index.number_of_shards	No	Integer	1	Number of index shards. This value should be divisible by the number of cluster nodes. Value range: 1–1024
index.number_of_replicas	No	Integer	1	Number of index replicas. Replicas improve data availability. Value range: 0 to the number of nodes minus 1

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
index.vector.exact_search_threshold	No	Integer	null (no switching)	Threshold for automatically switching from pre-filtering search to brute-force search. When the size of the intermediate result set in a segment is lower than this threshold, a brute-force search is performed. Value range: null (disables auto switchover from pre-filtering search to brute-force search) or a positive integer
index.vector.search.concurrency.enabled	No	Boolean	false	Whether to enable concurrent vector searches across segments. In an Elasticsearch cluster, each index shard consists of multiple segments. By default, each segment is searched in sequence. Enabling concurrent searches across segments reduces query latency but does not increase the cluster's maximum query throughput. Furthermore, it may increase the average CPU usage of cluster nodes. Constraints: This parameter is available only for Elasticsearch clusters whose image version is no earlier than 7.10.2_25.3.0_xxx. The value can be: • **true**: Enables concurrent search. • **false**: Performs serial search.

Table 5-11 mappings parameters

Parameter	Mandatory	Type	Default Value	Description
type	Yes	String	N/A	Data type of a field. Set this parameter to **vector** to indicate vector fields.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
dimension	Yes	Integer	N/A	Number of vector dimensions. Value range: 1–4096
indexing	No	Boolean	false	Whether to enable vector index acceleration. The value can be: • **true**: Enables vector index acceleration. When this parameter is set to **true**, an extra vector index is created. The indexing algorithm is specified by the **algorithm** field and the index supports vector search. • **false**: Disables vector index acceleration. If this parameter is set to **false**, vector data is written only to docvalues, and only **ScriptScore** and **Rescore** can be used for vector query.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
lazy_indexing	No	Boolean	false	Whether to enable lazy vector indexing. When enabled, the system prioritizes data ingestion speed by delaying the construction of vector indices. Instead of building the index in real-time, the system simply persists the raw data. After the ingestion process is complete, **offline index building** needs to be performed manually. This parameter is designed to balance write throughput against indexing overhead. Use this option for large-scale offline data migrations where high ingestion speed is critical and real-time search capabilities are not required during the ingestion phase. Constraints: • This parameter takes effect only when **indexing** is set to **true** in the mapping. • For Elasticsearch clusters, the image version must be 7.10.2_24.3.3_xxx or later. • For OpenSearch clusters, the version must be 2.19.0 or 3.4.0 (image version 3.4.0_26.6.0_xxx or later). The value can be: • **true**: Enables lazy indexing. Write throughput is improved. However, before offline index building is manually triggered and completed, data cannot be queried via VectorQuery. • **false**: Enables real-time index building. Indexing happens immediately upon ingestion, so data can be queried immediately after being written in.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
algorithm	No	String	GRAPH	Vector indexing algorithm. Constraints: • This parameter takes effect only when **indexing** is set to **true** in the mapping. • If you select **IVF_GRAPH** or **IVF_GRAPH_PQ**, you must perform **(Optional) Pre-Building and Registering Centroid Vectors** before creating a vector index. Value range: FLAT, GRAPH, GRAPH_PQ, GRAPH_SQ8, GRAPH_SQ4, IVF_GRAPH, and IVF_GRAPH_PQ. For how to select an algorithm and the cluster version needed to run it, see **Indexing Algorithms**. • When the GRAPH algorithm (GRAPH, GRAPH_PQ, GRAPH_SQ8, or GRAPH_SQ4) is selected, you can adjust the graph index structure and construction quality via **Table 5-12**. • When the GRAPH_PQ algorithm is selected, precision can be controlled via **Table 5-13**.
dim_type	No	String	float	Vector data type. The value can be: • binary: binary value • float: floating-point number

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
metric	No	String	euclidean	Vector distance metric, which measures the similarity or distance between vectors. The value can be: • **euclidean**: Euclidean distance • **inner_product**: inner product distance • **cosine**: cosine distance • **hamming**: Hamming distance, which can be used only when **dim_type** is set to **binary**.

Table 5-12 Optional parameters for the GRAPH indexing algorithm

Parameter	Mandatory	Type	Default Value	Description
neighbors	No	Integer	64	Maximum number of neighbors for each vector in the graph index. A larger value indicates denser connectivity of the graph and leads to higher retrieval precision (recall), but it also increases the index file size and slows down index building and query speeds. Constraints: This parameter takes effect only when **indexing** is set to **true** and **algorithm** is a GRAPH variant (GRAPH, GRAPH_PQ, GRAPH_SQ8, or GRAPH_SQ4) in the mapping. Value range: 20–255

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
shrink	No	Float	1	The aggressiveness of redundant edge removal (pruning) during graph construction. This setting directly controls the final graph density. A smaller value indicates more aggressive pruning, a sparser graph, and faster retrieval speeds, but may increase precision loss. Constraints: This parameter takes effect only when **indexing** is set to **true** and **algorithm** is a GRAPH variant (GRAPH, GRAPH_PQ, GRAPH_SQ8, or GRAPH_SQ4) in the mapping. Value range: 0-10
scaling	No	Integer	50	Scaling ratio for the number of nodes in the upper layers of the HNSW graph. This setting affects the layers and node distribution per layer in the HNSW graph. A proper scaling ratio ensures optimal cross-layer navigation efficiency during retrieval. Constraints: This parameter takes effect only when **indexing** is set to **true** and **algorithm** is a GRAPH variant (GRAPH, GRAPH_PQ, GRAPH_SQ8, or GRAPH_SQ4) in the mapping. Value range: 1–10000

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
efc	No	Integer	200	How many nearest neighbors to explore when inserting a new vector into the HNSW graph. This parameter controls the search depth during index building. A larger value results in higher-quality graph structure and query accuracy but slower index building. Constraints: This parameter takes effect only when **indexing** is set to **true** and **algorithm** is a GRAPH variant (GRAPH, GRAPH_PQ, GRAPH_SQ8, or GRAPH_SQ4) in the mapping. Value range: 1–100000 You are advised to increase the value for large-scale datasets.
max_scan_num	No	Integer	10000	Maximum number of nodes to scan during a single query. This parameter limits the search depth. A larger value results in higher query accuracy but increased latency. Constraints: This parameter takes effect only when **indexing** is set to **true** and **algorithm** is a GRAPH variant (GRAPH, GRAPH_PQ, GRAPH_SQ8, or GRAPH_SQ4) in the mapping. Value range: 1–1000000

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 5-13 Optional parameters for the GRAPH_PQ indexing algorithm

Parameter	Mandatory	Type	Default Value	Description
centroid_num	No	Integer	255 (corresponding to 8-bit quantization)	Number of centroids within each subspace for the PQ algorithm. This parameter determines the encoding precision of the vectors after quantization. A larger value indicates more precise representation of the original vectors and higher recall, but also a slight increase in computational overhead and memory footprint. Constraints: This parameter takes effect only when **indexing** is set to **true** and **algorithm** is set to **GRAPH_PQ** in the mapping. Value range: 1–65535

fragment_num	No	Integer	**0**: The plug-in automatically sets the number of fragments based on the vector dimensions.	Number of fragments (M) each high-dimensional vector is split into. It affects the quantization granularity. A larger value results in compressed vectors that more closely approximate the original vectors and hence higher retrieval precision. However, it also leads to higher storage consumption.
				Constraints: This parameter takes effect only when **indexing** is set to **true** and **algorithm** is set to **GRAPH_PQ** in the mapping.
				Value range: 0–4096
				When this parameter is set to **0**, the system automatically calculates the optimal number of segments based on the vector dimensions **dim**.
				`if dim <= 256:`
				`  fragment_num = dim / 4`
				`elif dim <= 512:`
				`  fragment_num = dim / 8`
				`else:`
				`  fragment_num = 64`

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

(Optional) Pre-Building and Registering Centroid Vectors

When the indexing algorithm is set to **IVF_GRAPH** or **IVF_GRAPH_PQ**, you must first pre-build and register the centroid vectors before creating a vector index.

IVF_GRAPH and IVF_GRAPH_PQ help to accelerate indexing and queries in ultra-large-scale datasets with more than 1 billion records. They allow you to narrow down the query scope by dividing a vector space into subspaces through clustering or random sampling. Before pre-build, you need to obtain all centroid vectors through clustering or random sampling. Centroid vectors are pre-built into a GRAPH or GRAPH_PQ index and then registered with the CSS vector database. Multiple nodes can share this index. Reuse of the centroid index among shards can effectively reduce the training overhead and the number of centroid index queries, improving write and query performance.

1. Create a centroid index.

   For example, run the following command to create a centroid index named **my_dict**:

   ```
   PUT my_dict
   {
     "settings": {
       "index": {
         "vector": true
       },
       "number_of_shards": 1,
       "number_of_replicas": 0
     },
     "mappings": {
       "properties": {
         "my_vector": {
           "type": "vector",
           "dimension": 2,
           "indexing": true,
           "algorithm": "GRAPH",
           "metric": "euclidean"
         }
       }
     }
   }
   ```

   For detailed parameter configuration, see **Creating a Vector Index**. Pay attention to the following mandatory parameters:

   - **index.number_of_shards**: The number of index shards must be set to **1**. Otherwise, the centroid index cannot be registered.
   - **indexing**: This parameter must be set to **true** to enable vector index acceleration.
   - **algorithm**: Set the indexing algorithm. Set it to **GRAPH** for the IVF_GRAPH algorithm, and GRAPH_PQ if the IVF_GRAPH_PQ algorithm is used.

2. Write centroid vectors to the created index. Write the centroid vectors obtained through sampling or clustering into the newly created index **my_dict**.

3. Call the registration API.

   For example, run the following command to register the centroid index as a Dict object with a globally unique name (**dict_name**):

   ```
   PUT _vector/register/my_dict
   {
   ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  "dict_name": "my_dict"
}
```

4. Create an IVF_GRAPH or IVF_GRAPH_PQ vector index.

   When creating the vector index, you do not need to specify **dimension** or **metric**. Rather, you specify the registered Dict object. **Table 5-14** describes key parameters for specifying a Dict object.

   For example, run the following command to create an IVF_GRAPH vector index:

```
PUT my_index
{
  "settings": {
    "index": {
      "vector": true,
      "sort.field": "my_vector.centroid" # Set the centroid subfield of each vector field as a sorting field.
    }
  },
  "mappings": {
    "properties": {
      "my_vector": {
        "type": "vector",
        "indexing": true,
        "algorithm": "IVF_GRAPH",
        "dict_name": "my_dict",
        "offload_ivf": true
      }
    }
  }
}
```

Table 5-14 Key parameters for specifying a Dict object

Parameter	Mandatory	Type	Default Value	Description
dict_name	Yes	String	N/A	Name of the centroid Dict object, for example, **my_dict**. The vector dimensions and similarity measurement method of the index are the same as those of the Dict object. There is no need to configure them again.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
offload_ivf	Yes	Boolean	false	Whether to offload the IVF inverted index to the Elasticsearch/OpenSearch engine layer. The value can be: • **true**: Offloads the IVF inverted index to Elasticsearch/OpenSearch for physical storage management. This significantly reduces the off-heap memory usage of the vector search engine. It also reduces CPU and memory overhead when writing and merging data in large throughput. • **false**: Keep the IVF inverted index entirely within the dedicated memory buffer of the vector search engine. When processing datasets of hundreds of millions of records or more, you should set this parameter to **true** to optimize the cluster's memory ratio while ensuring retrieval performance.

5.2.4 Importing Vector Data

In large-scale search and analytics scenarios, efficiently ingesting tens to hundreds of millions of high-dimensional vectors is a critical challenge. Because vector data is significantly larger than standard text, real-time construction of complex index structures (such as HNSW) consumes extensive CPU resources, which can bottleneck write throughput. To maximize ingestion performance, the CSS vector database supports two formats: directly readable floating-point arrays, and Base64 encoding optimized for higher network transmission efficiency. Additionally, where real-time retrieval is not required during the ingestion phase, the CSS vector database supports offline index building. This allows you to use the Bulk API to rapidly ingest massive datasets and later trigger offline index building during off-peak hours. This decoupled approach ensures high ingestion efficiency even with limited hardware resources.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

How the Feature Works

The CSS vector database supports two primary formats:

- Floating-point arrays: Standard JSON format (for example, [1.0, 2.0]), which is directly readable and easy to debug. Use this format for small datasets.
- Base64 encoding: Transforms single-precision floating-point numbers (little-endian byte order) into strings. For vectors of the same dimensionality, Base64-encoded data is approximately one-third the size of JSON arrays, significantly reducing network transmission overhead and accelerating the processing of high-dimensional vectors.

Choose an appropriate ingestion method based on your service requirements.

- Single-record import: Use for small-scale applications or testing.
- Bulk import: Use for large-scale applications, where write requests are batched to reduce network overhead.

The vector database utilizes an LSM (Log-Structured Merge) tree-like model for data persistence. When data is written in, it is first stored in a memory buffer and then periodically flushed to disk as index segments.

- Real-time index building (default): For each segment generated, the system immediately builds an HNSW graph index. This can cause write jitter, and the subsequent merging of small segments leads to redundant vector index computations.
- Offline index building: When **lazy_indexing** is set to **true**, the system delays indexing until data ingestion is complete. You then manually trigger offline index building, which merges all segments and constructs the index at once. This significantly accelerates end-to-end performance for ingestion and indexing. Use this option for large-scale offline migrations where high ingestion speed is critical and real-time search capabilities are not required during the ingestion phase.

Constraints

- The dimensions of the ingested vectors must strictly match the **dimension** defined during index creation.
- Base64 encoding must use the little-endian byte order. Otherwise, parsing errors may occur.
- Use offline index building only when there are no real-time query requirements; and for Elasticsearch clusters, the image version is 7.10.2_24.3.3_x.x.x or later, or for OpenSearch clusters, the version is 2.19.0 or 3.4.0 (image version 3.4.0_26.6.0_xxx or later).

Importing a Single Record

- Floating-point array

```
POST my_index/_doc
{
  "my_vector": [1.0, 2.0]
}
```

- Base64

```
POST my_index/_doc
{
```

```
  "my_vector": "AACAPwAAAEA="
}
```

Bulk Import

When importing data in bulk, we recommend keeping the size of each batch between 5 MB and 15 MB (approximately 100 to 1000 records) to balance between CPU load and network latency.

- Floating-point array

```
POST my_index/_bulk
{"index": {}}
{"my_vector": [1.0, 2.0], "my_label": "red"}
{"index": {}}
{"my_vector": [2.0, 2.0], "my_label": "green"}
{"index": {}}
{"my_vector": [2.0, 3.0], "my_label": "red"}
```

- Base64

```
POST my_index/_bulk
{"index":{}}
{"my_vector":"AACAPwAAAEA=", "my_label": "red"}
{"index":{}}
{"my_vector":"AAAAQAAAAEA=", "my_label": "green"}
{"index":{}}
{"my_vector":"AAAAQAAAQEA=", "my_label": "red"}
```

For details about how to use the Bulk API, see **Bulk API**.

(Optional) Triggering Offline Index Building

If lazy_indexing is enabled, offline index building must be performed after data ingestion. Otherwise, the system will return error code 500 for standard vector query, with the error message "Load native index failed exception." To solve this, perform offline index building before vector queries.

CAUTION

Use offline index building only when there are no real-time query requirements; and for Elasticsearch clusters, the image version is 7.10.2_24.3.3_x.x.x or later, or for OpenSearch clusters, the version is 2.19.0 or 3.4.0 (image version 3.4.0_26.6.0_xxx or later).

Offline index building consists of two steps:

1. Merge index segments.
2. Create the final vector index based on the final index segments.

Run the following command to trigger offline index building:

```
POST _vector/indexing/{index_name}
{
  "field": "my_vector"
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 5-15 Parameters for offline index building

Parameter	Mandatory	Type	Description
index_name	Yes	String	Name of the index to be built offline.
field	Yes	String	Vector field name. Constraints: **lazy_indexing** must have been set to **true** for this field in the mapping.

5.2.5 Performing Vector Search

In modern AI search applications, high-dimensional vectors capture the semantic essence of data, while business-specific tags (such as category, price, and region) enable precise filtering. While standard vector search is fast, it often struggles to accommodate complex business logic or custom ranking requirements. To bridge the gap between performance and flexibility, the CSS vector database supports a wide range of query methods, from ultra-fast approximate nearest neighbor (ANN) search to intensive re-scoring and fully customizable, script-based ranking. By strategically implementing pre-filtering to narrow the search space, or leveraging re-scoring to compensate for precision loss from vector quantization, developers can build intelligent search engines capable of efficiently processing datasets with hundreds of millions of records while meeting service requirements for both low latency and high recall.

Comparing Different Query Methods

To balance search accuracy and performance, select an appropriate query method based on the size of your dataset.

Table 5-16 A comparison of different query methods

Query Method	How It Works	When to Use	Cluster Version
Standard vector search	Executes ANN searches using graph-based structures like HNSW. Navigating pre-built hierarchical layers enables millisecond-level latency. This method maintains both ultra-high throughput and high recall precision (with only a slight loss).	Large-scale general search: scenarios that prioritize low latency, such as searching through hundreds of millions of images and recommending short videos.	Elasticsearch: 7.6.2 or 7.10.2 OpenSearch: 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later)

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Query Method	How It Works	When to Use	Cluster Version
Hybrid query: pre-filtering	Applies metadata filtering (such as category and region) first, and then executes full vector search on the filtered candidate set. By drastically pruning the search space, this method ensures excellent search performance and high recall. The returned results match the applied filters.	Conditional search: precise metadata filtering, for example, searching for similar products under a specific brand.	Elasticsearch: 7.10.2 OpenSearch: 2.19.0 and 3.4.0 (image version 3.4.0_26.6.0_xxx or later)
Hybrid query: Boolean query	Boolean query is in fact a post-filtering query method, where vector search and tag-based filtering are performed separately and then their results are merged using Boolean logic.	Simple logic combination: where fusion scoring is required or a legacy version does not support pre-filtering.	Elasticsearch: 7.6.2 or 7.10.2 OpenSearch: 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later)
Script score query	This method does not deal with vector indexes. It performs brute-force search on an intermediate result set obtained through pre-filtering. A 100% recall rate can be achieved on the filtered subset. However, latency increases linearly with the data volume, leading to performance bottlenecks on large datasets.	Small-scale ranking: high-precision matching over small datasets (less than 10,000 records) or second-pass ranking after precise metadata filtering.	Elasticsearch: 7.6.2 or 7.10.2 OpenSearch: 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later)
Rescore query	This method uses a two-stage retrieval process: pre-filtering and reranking. First, an initial query quickly retrieves a relatively large candidate set from the vector index, and then the top N more relevant results are retrieved through a precise reranking. This method significantly improves retrieval precision with only a slight increase in latency.	Large-scale quantization: Mitigates the precision loss of PQ quantization and ensures high recall while keeping a low memory footprint.	Elasticsearch: 7.6.2 or 7.10.2 OpenSearch: 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later)

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Query Method	How It Works	When to Use	Cluster Version
Painless syntax extension	Calls specialized vector functions via a custom script to implement complex, multi-weighted scoring formulas. Performance and precision can be tuned flexibly, with support for high-dimensional custom service logic.	Dynamic, multi-factor ranking, for example, based on "similarity + price offset + promotion weight"	Elasticsearch: 7.6.2 or 7.10.2 OpenSearch: 1.3.6, 2.19.0, and 3.4.0 (image version 3.4.0_26.6.0_xxx or later)

Logging in to Dev Tools

Log into Dev Tools to run DSL commands.

- **For an Elasticsearch cluster, log in to Kibana**
  a. Log in to the **CSS management console**.
  b. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
  c. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
  d. In the left navigation pane, choose **Dev Tools**.
     The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.
- **For an OpenSearch cluster, log in to Dashboards**
  a. Log in to the **CSS management console**.
  b. In the navigation pane on the left, choose **Clusters > OpenSearch**.
  c. In the cluster list, find the target cluster, and click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
  d. In the left navigation pane, choose **Dev Tools**.
     The left part of the console is the command input box, and the triangle icon in its upper-right corner is the execution button. The right part shows the execution result.

Standard Vector Query

Run the following command to retrieve the top K records that are the closest matches to the query vector.

```
POST my_index/_search
{
  "size":2,
  "_source": false,
  "query": {
    "vector": {
      "my_vector": {
        "vector": [1, 1],
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
      "topk":2
    }
   }
  }
}
```

Table 5-17 Parameters for standard query

Parameter	Mandatory	Type	Default Value	Description
size	No	Integer	10	Number of final search results to return.
_source	No	Boolean	true	Whether to return the source text in documents. To reduce data transmission and improve query performance, set this parameter to **false**. The value can be: • **true**: Returns the source text. • **false**: Not to return the source text.
query	Yes	Map	N/A	Specifies the query vector. Parameter description: **vector** (mandatory): indicates a vector query (vector similarity-based search), including the vector field and query vector value. **my_vector** (mandatory): queried vector field (for example, **my_vector**).
vector (sub-parameter)	Yes	Array/ String	N/A	Query vector value. It is used to calculate the similarity between indexed vectors and the query vector. The value can be an array (for example, [1, 1]) or Base64-encoded value (for example, **AAABAAACAAAD**).
topk	Yes	Integer	N/A	The number of the most similar or relevant results to be returned. Keep the value consistent with **size**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
ef	No	Integer	200	The number of neighbor nodes to explore within the HNSW graph during a query. A larger value indicates a higher query accuracy yet slower query speed. Constraints: This parameter takes effect only when **algorithm** is set to **GRAPH**, **GRAPH_PQ**, **GRAPH_SQ8**, or **GRAPH_SQ4** in the index mapping. Value range: 1–1000000
max_scan_num	No	Integer	10000	Maximum number of graph nodes to scan within the HNSW graph during a query. A larger value indicates a higher query accuracy yet slower query speed. Constraints: This parameter takes effect only when **algorithm** is set to **GRAPH**, **GRAPH_PQ**, **GRAPH_SQ8**, or **GRAPH_SQ4** in the index mapping. Value range: 0–1000000
rescore	No	Boolean	false	Whether to enable rescore query. During PQ or SQ index queries, this parameter determines whether to enable rescore query. The effect is the same as **Rescore Query**, but the configuration is simpler. Constraints: This parameter is available only for Elasticsearch 7.10.2 (image version 7.10.2_26.1.0_xxx or later), OpenSearch 2.19.0 (image version 2.19.0_26.1.0_xxx or later), and OpenSearch 3.4.0 (image version 3.4.0_26.6.0_xxx or later). The value can be: • **true**: Enable rescore query. • **false**: Disable rescore query.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
nprobe	No	Integer	100	Number of centroids to explore during an IVF index query. A larger value indicates a higher query accuracy yet slower query speed. Constraints: This parameter takes effect only when **algorithm** is set to **IVF_GRAPH** or **IVF_GRAPH_PQ** in the index mapping. Value range: 0–100000

Hybrid Query: Pre-filtering

A hybrid query means combining vector search with other DSL queries. With pre-filtering, documents are first filtered using metadata tags (such as colors and categories), and then vector similarity search is performed on the filtered result set to return the most similar results.

Run the following command to retrieve the top 10 results whose **my_label** is **red**:

```
POST my_index/_search
{
  "size": 10,
  "query": {
    "vector": {
      "my_vector": {
        "vector": [1, 2],
        "topk": 10,
        "filter": {
          "term": { "my_label": "red" }
        }
      }
    }
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 5-18 Parameters of pre-filtering query

Parameter	Mandatory	Type	Description
filter	Yes	Map	Vector query filters. The filters are in key-value pairs. The key can be a standard filter such as term and range, and the value is the specific filtering condition. If **filter** is too restrictive, leading to a small intermediate result set, you can set the **index.vector.exact_search_threshold** parameter, so that when the intermediate result set is smaller than this threshold, pre-filtering query automatically switches over to brute-force query (FLAT algorithm), which ensures a high recall rate. For more information, see **Creating a Vector Index**.
term	No	Map	Term query is a type of exact query. In the key-value pair, the key is the field name, and the value is the exact value to match. For example, **{"term": {"my_label": "red"}}** means only to return documents whose **my_label** value is **red**.

Hybrid Query: Boolean Query

A hybrid query means combining vector search with other DSL queries. A Boolean query is in fact a post-filtering query method. Filtering and vector similarity-based search are performed separately. Then, the results of the two are combined using Boolean logic defined by clauses like must, should, and filter.

Run the following command to retrieve the top 10 results whose **my_label** is **red**:

```
POST my_index/_search
{
  "size": 10,
  "query": {
    "bool": {
      "must": {
        "vector": {
          "my_vector": {
            "vector": [1, 2],
            "topk": 10
          }
        }
      },
      "filter": {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
      "term": { "my_label": "red" }
    }
   }
  }
}
```

Table 5-19 Boolean query parameters

Parameter	Mandatory	Type	Description
bool	Yes	Map	A compound query clause that combines subqueries using configured Boolean logic. Parameter description: • **must**: Clauses that must match for documents to be included in the results. • **filter**: It is similar to **must**, but does not contribute to the relevance score. • **should**: Clauses that should match, but are not required. They are like nice-to-haves. • **must_not**: Clauses that must not match for documents to be included in the results.
bool.must	Yes	Map	Clauses that must match for documents to be included in the results. Parameter description: • **vector**: query vector • **my_vector**: vector field • **topk**: number of results to return
bool.filter	Yes	Map	Clauses that must match, but do not contribute to the relevance score.

Script Score Query

This method calculates vector similarity on pre-filtered results by executing a user-defined script. It does not use vector indexes. Its performance depends on the size of the pre-filtered result set. If the pre-filtering condition is set to **match_all**, a brute-force search is performed on all data.

Run the following command to perform a brute-force search to retrieve two vector records:

```
POST my_index/_search
 {
  "size":2,
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
 "query": {
 "script_score": {
    "query": {
     "match_all": {}
    },
    "script": {
     "source": "vector_score",
     "lang": "vector",
     "params": {
      "field": "my_vector",
      "vector": [1.0, 2.0],
      "metric": "euclidean"
     }
    }
   }
  }
}
```

Table 5-20 script_score query parameters

Parameter	Mandatory	Type	Default Value	Description
script_score	Yes	Map	N/A	A root parameter for the script_score query. Parameter description: • **query**: pre-filtering criteria. When it is set to **match_all**, a brute-force search is performed on all data. • **script**: custom script that calculates similarity scores.
source	Yes	String	N/A	Script name. The value is fixed to **vector_score**, indicating that a built-in script is used for calculating similarity.
lang	Yes	String	N/A	Script language type. The value is fixed to **vector**.
field	Yes	String	N/A	Queried vector field, for example, **my_vector**.
vector	Yes	Array/ String	N/A	Query vector value. It is used to calculate the similarity between indexed vectors and the query vector. The value can be an array (for example, [1, 1]) or Base64-encoded value (for example, **AAABAAACAAAD**).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Mandatory	Type	Default Value	Description
metric	No	String	euclidean	Vector distance metric, which measures the similarity or distance between vectors. The value can be: • **euclidean**: Euclidean distance • **inner_product**: inner product distance • **cosine**: cosine distance • **hamming**: Hamming distance, which can be used only when **dim_type** is set to **binary**.

Rescore Query

When the index uses the GRAPH_PQ or IVF_GRAPH_PQ algorithm, the query consists of two stages: first, a quick filtering is performed on the compressed vectors; then, the original vectors are retrieved to accurately rank the top K results to improve recall.

Assume that **my_index** is an index that utilizes product quantization. Run the following command to perform a rescore query to return 10 results:

```
GET my_index/_search
{
  "size": 10,
  "query": {
    "vector": {
      "my_vector": {
        "vector": [1.0, 2.0],
        "topk": 100
      }
    }
  },
  "rescore": {
    "window_size": 100,
    "vector_rescore": {
      "field": "my_vector",
      "vector": [1.0, 2.0],
      "metric": "euclidean"
    }
  }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 5-21 Rescore query parameters

Parameter	Mandatory	Type	Default Value	Description
rescore	Yes	Map	N/A	Defines rescoring parameters. Parameter description: • **window_size**: rescoring/reranking window size. • **vector_rescore**: other vector rescoring settings. **CAUTION** To implement rescore query for nested fields, perform **Standard Vector Query**.
window_size	No	Integer	100	Rescoring/reranking window size. The vector search returns the top k results, but only the first *window_size* results are rescored and reranked. A larger value indicates a larger reranking scope and hence a higher recall rate, but it also leads to higher computational overhead.
field	Yes	String	N/A	Queried vector field, for example, **my_vector**.
vector	Yes	Array/String	N/A	Query vector value. It is used to calculate the similarity between indexed vectors and the query vector. The value can be an array (for example, [1, 1]) or Base64-encoded value (for example, **AAABAAACAAAD**).
metric	No	String	euclidean	Vector distance metric, which measures the similarity or distance between vectors. The value can be: • **euclidean**: Euclidean distance • **inner_product**: inner product distance • **cosine**: cosine distance • **hamming**: Hamming distance, which can be used only when **dim_type** is set to **binary**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Painless Syntax Extension

This method allows you to embed vector distance or similarity calculation functions in custom scripts to implement complex logic.

Run the following command to perform a Painless syntax extension-based query to retrieve 10 vector records:

```
POST my_index/_search
{
  "size": 10,
  "query": {
    "script_score": {
      "query": {
        "match_all": {}
      },
      "script": {
        "source": "1 / (1 + euclidean(params.vector, doc[params.field]))",
        "params": {
          "field": "my_vector",
          "vector": [1, 2]
        }
      }
    }
  }
}
```

As shown in **Table 5-22**, CSS supports several vector distance/similarity calculation functions, which users can use readily in custom Painless scripts to build flexible rescoring formulas.

Table 5-22 Supported vector distance/similarity calculation functions

Function Signature	Description
euclidean(Float[], DocValues)	Euclidean distance
cosine(Float[], DocValues)	Cosine similarity
innerproduct(Float[], DocValues)	Inner product
hamming(String, DocValues)	Hamming distance Constraints: • Only vectors whose **dim_type** is **binary** are supported. The input query vector must be a Base64-encoded string. • Only Elasticsearch 7.6.2, Elasticsearch 7.10.2, and OpenSearch 1.3.6 clusters support this function.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Function Signature	Description
hammings(String, DocValues)	Hamming distance Constraints: • Only vectors whose **dim_type** is **binary** are supported. The input query vector must be a Base64-encoded string. • This function is supported only in OpenSearch 2.19.0 and OpenSearch 3.4.0 (image version 3.4.0_26.6.0_xxx or later).

5.2.6 Implementing Nested Vector Search

When processing long documents, such as technical manuals or legal provisions, documents are often partitioned into semantic chunks based on paragraph boundaries or fixed token lengths, with vector embeddings generated for each chunk. Traditional flat storage structures make it difficult to maintain the mapping between document-level metadata (such as IDs and titles) and per-chunk vectors, so it can be difficult to accurately locate the chunk that triggered a match during retrieval. To address this, the CSS vector database supports vector search with nested fields. This allows multiple sub-documents to be nested within a parent document. At query time, the search engine traversed all nested vectors in a single request, and the most relevant chunk can be retrieved based on similarity score. This greatly simplifies architectural complexity for long-text retrieval.

How the Feature Works

In standard Elasticsearch and OpenSearch configurations, object arrays are flattened into a single multi-value field at the storage layer. This causes the association between different sub-fields to get lost. By using nested fields, each nested object is stored as a separate, hidden sub-document linked to a single parent document, ensuring that each chunk in a nested object remains associated with its corresponding vector during retrieval.

When you perform vector search on a nested field, the following process happens:

1. Sub-document scanning: Traverses the vectors of all sub-documents within each parent document.
2. Partial scoring: Calculate the similarity score between each sub-vector and the query vector.
3. Score aggregation: Determine the final relevance score of each parent document based on the defined score_mode. Typically, score_mode is set to **max**, indicating that a document should get a high score if any of its chunks is most relevant to the query.

Constraints

- Nested vector search is supported only in Elasticsearch 7.10.2, OpenSearch 2.19.0, and OpenSearch 3.4.0 (image version 3.4.0_26.6.0_xxx or later).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- Nested fields increase the total number of documents. For a large-scale dataset, you should carefully evaluate this operational overhead.
- Rescore query cannot be implemented for nested fields through the rescore statement. Only the query.vector statement (see **Standard Vector Query**) can be used to do that.

Creating a Vector Index with Nested Fields

Run the following command to create a vector index with nested fields. The index contains an **id** field whose type is **keyword**, and an **embedding** field whose type is **nested**. The embedding field contains two subfields: **chunk** and **emb**. The **chunk** subfield is of the **keyword** type, and the **emb** subfield is of the **vector** type.

```
PUT my_index
{
  "settings": {
    "index.vector": true
  },
  "mappings": {
    "properties": {
      "id": {
        "type": "keyword"
      },
      "embedding": {
        "type": "nested",        // Declares the nested type.
        "properties": {
          "chunk": {
            "type": "keyword"
          },
          "emb": {               // Vector embedding in the subfield
            "type": "vector",
            "dimension": 2,
            "indexing": true,
            "algorithm": "GRAPH",
            "metric": "euclidean"
          }
        }
      }
    }
  }
}
```

Importing Vector Data

Run the following command to bulk-import parent documents along with their corresponding chunk embeddings as nested objects. Each parent document contains two vector embeddings.

```
POST my_index/_bulk
{"index":{}}
{"id": 1, "embedding": [{"chunk":1,"emb": [1, 1]}, {"chunk":2,"emb": [2, 2]}]}
{"index":{}}
{"id": 2, "embedding": [{"chunk":1,"emb": [2, 2]}, {"chunk":2,"emb": [3, 3]}]}
{"index":{}}
{"id": 3, "embedding": [{"chunk":1,"emb": [3, 3]}, {"chunk":2,"emb": [4, 4]}]}
```

Performing a Nested Vector Search

Retrieve and rank parent documents based on the similarity scores of their most relevant internal chunks. Because the data is stored in a nested structure, you must use a nested query to retrieve nested fields. To perform such a query, you need to set the path parameter to specify the nested path, and set **score_mode** to

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

max, indicating the maximum similarity between all vectors in the document and the query vector.

- Standard query

  Query the top 10 documents that are most similar to vector [1, 1].

```
GET my_index/_search
{
  "_source": {"excludes": ["embedding"]}, // Returns only the parent document ID, and the complex vector array remains hidden.
  "query": {
    "nested": {
      "path": "embedding",                // The nested field path must be specified.
      "score_mode": "max",                // Policy for consolidating the scores of sub-documents for the parent document: Use the score of the most relevant chunk.
      "query": {
        "vector": {
          "embedding.emb": {
            "vector": [1, 1],
            "topk": 10
          }
        }
      }
    }
  }
}
```

  An example of the query result:

```
{
  "took" : 2,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 3,
      "relation" : "eq"
    },
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "my_index",
        "_type" : "_doc",
        "_id" : "Hc4Vc5QBSxCnghau22AE",
        "_score" : 1.0,
        "_source" : {
          "id" : 1
        }
      },
      {
        "_index" : "my_index",
        "_type" : "_doc",
        "_id" : "Hs4Vc5QBSxCnghau22AE",
        "_score" : 0.33333334,
        "_source" : {
          "id" : 2
        }
      },
      {
        "_index" : "my_index",
        "_type" : "_doc",
        "_id" : "H84Vc5QBSxCnghau22AE",
        "_score" : 0.11111111,
        "_source" : {
          "id" : 3
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
      }
    }
   ]
  }
}
```

- Pre-filtering query

  First retrieve documents whose ID is ["2", "3"], and then return the top 10 documents that are most similar to the query vector [1, 1].

```
GET my_index/_search
{
  "query": {
    "nested": {
      "path": "embedding",               // The nested field path must be specified.
      "score_mode": "max",               // Policy for consolidating the scores of sub-documents for the parent document: Use the score of the most relevant chunk.
      "query": {
        "vector": {
          "embedding.emb": {
            "vector": [1, 1],
            "topk": 10,
            "filter": {
              "terms": {"id": ["2", "3"]}
            }
          }
        }
      }
    }
  }
}
```

  An example of the query result:

```
{
  "took" : 3,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 2,
      "relation" : "eq"
    },
    "max_score" : 0.33333334,
    "hits" : [
      {
        "_index" : "my_index",
        "_type" : "_doc",
        "_id" : "3t0ZypcB-Tff59gMTZO2",
        "_score" : 0.33333334,
        "_source" : {
          "id" : 2,
          "embedding" : [
            {
              "chunk" : 1,
              "emb" : [
                2,
                2
              ]
            },
            {
              "chunk" : 2,
              "emb" : [
                3,
                3
              ]
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        }
       ]
      }
     },
     {
      "_index" : "my_index",
      "_type" : "_doc",
      "_id" : "390ZypcB-Tff59gMTZO2",
      "_score" : 0.11111111,
      "_source" : {
       "id" : 3,
       "embedding" : [
        {
         "chunk" : 1,
         "emb" : [
          3,
          3
         ]
        },
        {
         "chunk" : 2,
         "emb" : [
          4,
          4
         ]
        }
       ]
      }
     }
    ]
   }
}
```


5.2.7 Optimizing Query and Write Performance

In large-scale vector search scenarios, as vector dimensionality and data volume grow, clusters may face performance bottlenecks such as low write throughput and severe jitter in query latency (P99). Unlike traditional keyword-based search, vector search is defined by compute-intensive index building and memory-intensive retrieval. Standard Elasticsearch and OpenSearch configurations may struggle with the complex topology computations involved. To address these challenges, the CSS vector database supports full-stack performance tuning for clusters, covering both write and query. It helps you achieve the optimal balance between system performance and costs while ensuring high recall.

Optimizing Write Performance

Vector data ingestion involves three major overheads: replica synchronization, index refresh, and segment merging. During real-time index data ingestion, frequent index refresh operations generate a large number of small segments. This triggers frequent vector index build and merge operations, which consume excessive CPU/IO resources. You can try the following solutions to optimize write performance.

Solution 1: Temporarily disable replicas

- Description

  Temporarily disable replicas during data ingestion and enable them after data ingestion is complete. Use this solution when importing historical data in batches or performing a full update (for example, when initializing a vector database).

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- Operation

  Set the number of replicas:

  ```
  PUT {index_name}/_settings
  {
     "number_of_replicas": 0
  }
  ```

- Result

  Write performance is enhanced by avoiding real-time vector index building on replica nodes.

Solution 2: Adjust the refresh interval

- Description

  Set the index refresh interval to 120s or longer to reduce the number of small segments generated during frequent index refreshes and also reduce the vector index building overhead caused by segment merges. You can also disable automatic index refresh by setting the refresh interval to –1. Use this solution in high-throughput write scenarios (for example, when writing vectorized log data).

- Operation

  Set **refresh_interval**.

  ```
  PUT {index_name}/_settings
  {
     "refresh_interval": "120s"
  }
  ```

- Result

  Index refreshes occur less frequently. This reduces the number of small segments and also the overhead of segment merges, leading to enhanced write performance.

Solution 3: Increase indexing threads

- Description

  Increasing the number of threads for vector index building accelerates the indexing process. However, too many such threads will compete for query resources. Use this solution when there are sufficient CPU resources but the write latency is high—such as in GPU-accelerated environments.

- Operation

  The default value of **native.vector.index_threads** is 4. Change this value as needed.

  ```
  PUT _cluster/settings
  {
    "persistent": {
      "native.vector.index_threads": 8
    }
  }
  ```

- Result

  Vector index building is accelerated, and concurrent write performance is enhanced.

Optimizing Query Performance

Query performance is affected by the following factors: the number of segments, the memory circuit breaker mechanism, and field recall. An excessively large

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

number of segments impacts search efficiency; when off-heap memory becomes insufficient, vector index data is frequently swapped in and out of the memory; recalling all fields increases the load during the fetch phase. You can optimize query performance by addressing these factors.

Solution 1: Perform force merge

- Description

  After batch data ingestion, perform the force merge operation to forcibly merge segments, thus reducing the number of segments. Typically, you should perform this operation after data ingestion and before data query (for example, after a scheduled batch ingestion).

- Operation

  Perform the force merge operation:

  ```
  POST {index_name}/_forcemerge?max_num_segments=1
  ```

- Result

  Multiple segments are merged into a single segment. This reduces the file scanning overhead and accelerates the query speed.

Solution 2: Adjust the automatic merge policy

- Description

  In real-time ingestion and update scenarios, many small, fragmented segments may be created. You can tune the automatic merge policy to accelerate the merging of these segments into larger ones. This reduces the total number of segments and improves query performance.

- Operation

  Adjust the automatic merge policy for an index.

  ```
  PUT {index_name}/_settings
  {
    "index": {
      "merge": {
        "policy": {
          "max_merged_segment": "10gb",
          "max_merge_at_once": 10,
          "segments_per_tier": 5,
          "floor_segment": "200mb"
        }
      }
    }
  }
  ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 5-23 Parameters for adjusting the automatic merge policy

Parameter	Type	Default Value	Description
max_merged_segment	String	5GB	Maximum segment size after merging. Segments exceeding this size are excluded from automatic merging, but they remain eligible for forcible merge. This prevents segments from growing indefinitely, as they may block I/O links. For vector indexes, increasing this value helps reduce the total number of segments. Value format: positive integer + unit. Supported units: B, KB, MB, and GB (case-insensitive).
max_merge_at_once	Integer	10	Maximum number of segments that can be processed in a single merge task. Increasing this value can reduce the total number of segments, but will lead to high disk I/O and CPU spikes during the merge. Minimum value: 2
segments_per_tier	Integer	10	Maximum number of segments per tier. A smaller value indicates more frequent merging and faster query, but higher I/O pressure. A larger value enhances write throughput but may compromise query performance. You can decrease this value to trigger automatic merging more frequently and thus reduce the number of fragmented segments. Value range: ≥ max_merge_at_once

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Default Value	Description
floor_segment	String	2MB	Minimum segment size. Segments below this threshold will be merged first. You can increase this value to trigger automatic merging more frequently and thus reduce the number of fragmented segments. Value format: positive integer + unit. Supported units: B, KB, MB, and GB (case-insensitive).

- Result

  Increasing the maximum segment size helps to reduce the number of segments and thus accelerate query performance.

Solution 3: Adjust the circuit breaker limit for off-heap memory

- Description

  When the off-heap memory required by vector indexes exceeds the circuit breaker limit, the index cache manager frequently swaps in and out index data from the cache, which slows down queries. You can raise the circuit breaker limit to reduce circuit breaking (indicated by CircuitBreakingException recorded in the log) resulted from insufficient memory.

- Operation

  The default circuit breaker limit for off-heap memory is 80%. You can adjust this limit as required.

  ```
  PUT _cluster/settings
  {
    "persistent": {
      "native.cache.circuit_breaker.cpu.limit": "85%"
    }
  }
  ```

- Result

  It is less likely for vector index data to be swapped out from the memory, and query jitter is reduced.

Solution 4: Optimize field recall

- Description

  If the query result needs to return only a few fields that are either keywords or numeric values, you can use the docvalue_fields parameter to fetch them. Use this method if only numeric or enumerated metadata (such as product IDs and class labels) needs to be fetched. It can significantly reduce overhead during the fetch phase.

- Operation

  Use the docvalue_fields parameter to fetch only specific fields:

  ```
  POST {index_name}/_search
  {
  ```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  "size": 2,
  "stored_fields": ["_none_"],
  "docvalue_fields": ["my_label"],
  "query": {
    "vector": {
      "my_vector": {
        "vector": [1, 1],
        "topk": 2
      }
    }
  }
}
```

- Result

  There is no need to parse the entire _source document. Column-oriented storage (docvalues) reduces the overhead during the fetch phase and improves query performance.

5.2.8 Managing the Vector Search Cache

For vector search systems processing hundreds of millions of vectors, maintaining millisecond-level latency requires storing a large number of high-dimensional vector indexes in memory. Unlike traditional Elasticsearch or OpenSearch implementations that rely heavily on JVM heap memory, the CSS vector search engine is built on C++ and uses off-heap memory, which delivers superior performance. Without effective lifecycle management, large-scale deployments may experience: out of memory (OOM) if inactive (or cold) indexes accumulate and occupy too much memory; or unstable query latency due to frequent "swap-in and swap-out" of cached index segments. To address this problem, the CSS vector database implements a comprehensive set of off-heap memory management policies to ensure stable search performance under heavy loads. These policies include: real-time monitoring of memory usage watermarks; index preloading to mitigate high first-query latency; and dynamic cache reclamation through automatic cache clearing based on predefined idle timeout periods or usage thresholds.

How the Feature Works

The CSS vector database divides cluster physical memory into JVM heap memory and off-heap memory. The management policies for off-heap memory are as follows:

- Upon first hits, vector segments are loaded from the disk into off-heap memory. In this case, queries may experience high latency.
- Once the data is resident in off-heap memory, all subsequent queries are served directly from the cache, enabling millisecond-level response times.
- When off-heap memory is full or the predefined timeout period expires, inactive segments are evicted from off-heap memory to help maintain stable query performance under heavy loads.

Monitoring Cache Status

To troubleshoot performance bottlenecks, check each cluster node's off-heap memory utilization and cache hit rate.

Run the following command to monitor the cache status:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
GET /_vector/stats
```

Example response:

```
{
  "_nodes" : {          # Node information
    "total" : 1,        # Total number of nodes
    "successful" : 1,             # Number of successful nodes
    "failed" : 0          # Number of failed nodes
  },
  "cluster_name" : "css-d3a7",            # Cluster name
  "cpu_circuit_breaker_triggered" : false,     # Whether circuit breaking is triggered
  "nodes" : {
    "cAHmVUZTR9ON7t6jxcDCkg" : {          # Node UUID
      "cpu_cache_capacity_reached" : false,     # Whether the off-heap memory usage of the current node
reaches the upper limit
      "cpu_eviction_count" : 0,          # Number of segment-level cache swap-outs on the current node
      "cpu_hit_count" : 0,             # Number of segment-level cache hits on the current node
      "cpu_load_exception_count" : 0,          # Number of segment-level index loading failures on the current
node
      "cpu_load_success_count" : 0,          # Number of segment-level index loading successes on the current
node
      "cpu_miss_count" : 0,              # Number of segment-level cache misses on the current node
      "cpu_query_memory_usage" : 0,          # Off-heap memory usage on the current node, in KB
      "cpu_total_load_time" : 0          # Total time loading segments to the off-heap memory on the current
node, in ms
    }
  }
}
```

Preloading Frequently Accessed Indexes

Preload frequently accessed or newly ingested indexes to the off-heap memory to mitigate high first-query latency (because data needs to be loaded from disk).

Run the following command to preload a specified index:

```
PUT /_vector/warmup/{index_name}
```

Table 5-24 Parameter description

Parameter	Type	Default Value	Description
index_name	String	N/A	Specifies one or more vector indexes. • Single index: Enter the index name, for example, **my_index**. • Multiple indexes: Enter multiple index names and use a comma (,) to separate them, for example, **my_index1,my_index2**. • Wildcard: Use the wildcard (*) to match multiple indexes. For example, **myindex*** indicates all indexes whose name starts with **myindex**.

Example response:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
{
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  }
}
```

Configuring an Automatic Cache Clearing Policy

When data is frequently updated or memory resources become constrained, you can enable automatic cache clearing to automatically evict inactive segments and reclaim off-heap memory, ensuring stable query performance under heavy loads.

Run the following command to enable automatic eviction of segments that have exceeded their idle timeout period:

```
PUT _cluster/settings
{
  "persistent": {
    "native.cache.expiry.enabled": "true",
    "native.cache.expiry.time": "30m"
  }
}
```

Table 5-25 Parameter description

Parameter	Type	Default Value	Description
native.cache.expiry.enabled	Boolean	false	Whether to enable automatic cache clearing. When enabled, inactive segments are automatically evicted when their idle timeout period expires. The value can be: • **true**: Enables automatic cache clearing. • **false**: Disables automatic cache clearing.
native.cache.expiry.time	String	24h	Idle timeout period for evicting inactive segments. Value format: number + unit • Number: a natural number • Unit: s (second), m (minute), h (hour), or d (day) Example: **24h** (24 hours) and **30m** (30 minutes).

Manually Clearing the Cache

When off-heap memory reaches its capacity, the system automatically manages data through a "swap-in and swap-out" process. However, frequent, high-volume

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

cache churn can impact query performance. After deleting indexes or switching workloads, you can manually reclaim off-heap memory occupied by inactive index segments to ensure query performance for hot data indexes.

- Clear the full cache:
  ```
  PUT /_vector/clear/cache
  ```
- Clear specified indexes from the cache:
  ```
  PUT /_vector/clear/cache/{index_name}
  ```

Example response:

```
{
  "acknowledged" : "true"
}
```

5.2.9 Client Code Examples

5.2.9.1 Elasticsearch Vector Search Sample Code (Java)

The CSS vector database is fully compatible with the native Elasticsearch Java client. With this SDK, you can use Java to easily manage the vector index lifecycle for Elasticsearch clusters, from creating vector indexes and importing high-dimensional vectors at high speed, to performing millisecond-level approximate nearest neighbor (ANN) searches. The sample code in this topic is modular and ready to run. In just a few minutes, you can execute the complete workflow from data ingestion to vector search, helping you accelerate your service deployment.

Prerequisites

Java client dependencies have been installed on the server. Sample code in Maven mode:

```
<dependency>
   <groupId>org.elasticsearch.client</groupId>
   <artifactId>elasticsearch-rest-high-level-client</artifactId>
   <version>7.10.2</version>
</dependency>
```

Sample Code

```
package org.example;

import org.apache.http.HttpEntity;
import org.apache.http.HttpHost;
import org.apache.http.HttpStatus;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.elasticsearch.action.admin.indices.delete.DeleteIndexRequest;
import org.elasticsearch.action.admin.indices.refresh.RefreshRequest;
import org.elasticsearch.action.bulk.BulkRequest;
import org.elasticsearch.action.bulk.BulkResponse;
import org.elasticsearch.action.index.IndexRequest;
import org.elasticsearch.action.search.SearchResponse;
import org.elasticsearch.action.support.master.AcknowledgedResponse;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.RequestOptions;
import org.elasticsearch.client.Response;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.client.indices.CreateIndexRequest;
import org.elasticsearch.client.indices.CreateIndexResponse;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.xcontent.DeprecationHandler;
import org.elasticsearch.common.xcontent.NamedXContentRegistry;
import org.elasticsearch.common.xcontent.XContentParser;
import org.elasticsearch.common.xcontent.XContentType;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
import java.io.IOException;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;

public class ClientExample {
    private final RestHighLevelClient client;

    public ClientExample(RestHighLevelClient client) {
        this.client = client;
    }

    //Create a client for accessing non-security clusters.
    public static RestHighLevelClient getClient(List<String> hosts, int port, String scheme) {
        HttpHost[] httpHosts = hosts.stream().map(p -> new HttpHost(p, port,
scheme)).toArray(HttpHost[]::new);
        return new RestHighLevelClient(RestClient.builder(httpHosts));
    }

    //Create a client for accessing security clusters.
    public static RestHighLevelClient getClient(List<String> hosts, int port, String scheme, String user, String
password) {
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials(user,
password));
        SSLContext sc = null;
        try {
            sc = SSLContext.getInstance("SSL");
            sc.init(null, trustAllCerts, new SecureRandom());
        } catch (KeyManagementException | NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        HttpHost[] httpHosts = hosts.stream().map(p -> new HttpHost(p, port,
scheme)).toArray(HttpHost[]::new);
        final SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc,
NoopHostnameVerifier.INSTANCE);
        RestClientBuilder builder = RestClient.builder(httpHosts).setHttpClientConfigCallback(httpClientBuilder
-> {
            httpClientBuilder.disableAuthCaching();
            httpClientBuilder.setSSLStrategy(sessionStrategy);
            return httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
        });
        return new RestHighLevelClient(builder);
    }

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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            }
            @Override
            public X509Certificate[] getAcceptedIssuers() {
                return null;
            }
        }
};

# Create an index.
public void create(String index) throws IOException {
    CreateIndexRequest request = new CreateIndexRequest(index);
    request.settings(Settings.builder()
            .put("index.vector", true) // Enable the vector feature.
            .put("index.number_of_shards", 1) //Set the number of index shards as needed.
            .put("index.number_of_replicas", 0) //Set the number of index replicas as needed.
    );
    String mapping =
            "{\n" +
            "  \"properties\": {\n" +
            "    \"my_vector\": {\n" +
            "      \"type\": \"vector\",\n" +       // Set this field as the vector type.
            "      \"indexing\": \"true\",\n" +     // Enable index acceleration.
            "      \"dimension\": \"2\",\n" +       // Vector dimension
            "      \"metric\": \"euclidean\",\n" + // Similarity metric
            "      \"algorithm\": \"GRAPH\"\n" +    // Index algorithm
            "    }\n" +
            "  }\n" +
            "}";
    request.mapping(mapping, XContentType.JSON);
    CreateIndexResponse response = client.indices().create(request, RequestOptions.DEFAULT);
    if (response.isAcknowledged()) {
        System.out.println("create " + index + " success");
    }
}

// Write data. You are advised to keep the batch size under 500 records.
public void write(String index, List<float[]> vectors) throws IOException {
    BulkRequest request = new BulkRequest();
    for (float[] vec : vectors) {
        request.add(new IndexRequest(index).source("my_vector", vec));
    }

    BulkResponse response = client.bulk(request, RequestOptions.DEFAULT);
    if (response.hasFailures()) {
        System.out.println(response.buildFailureMessage());
    } else {
        System.out.println("write bulk to " + index + " success");
    }

    // Optional. Elasticsearch will refresh it by default.
    client.indices().refresh(new RefreshRequest(index), RequestOptions.DEFAULT);
}

// Query vectors.
public void search(String index, float[] query, int size) throws IOException {
    String queryFormat =
        "{\n" +
        "  \"size\":%d,\n" +
        "  \"query\": {\n" +
        "    \"vector\": {\n" +
        "      \"my_vector\": {\n" +  // Query the vector field name.
        "        \"vector\": %s,\n" +
        "        \"topk\":%d\n" +
        "      }\n" +
        "    }\n" +
        "  }\n" +
        "}";
    String body = String.format(queryFormat, size, Arrays.toString(query), size);
    Request request = new Request("POST", index + "/_search");
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        request.setJsonEntity(body);
        Response response = client.getLowLevelClient().performRequest(request);
        if (response.getStatusLine().getStatusCode() != HttpStatus.SC_OK) {
            System.out.println(response.getEntity()); //Handle the error based on service requirements.
            return;
        }
        // Process the normally returned result based on service requirements.
        HttpEntity entity = response.getEntity();
        XContentType xContentType = XContentType.fromMediaTypeOrFormat("application/json");
        XContentParser parser = xContentType.xContent().createParser(NamedXContentRegistry.EMPTY,
                DeprecationHandler.IGNORE_DEPRECATIONS, entity.getContent());
        SearchResponse searchResponse = SearchResponse.fromXContent(parser);
        System.out.println(searchResponse);
    }

    // Delete an index.
    public void delete(String index) throws IOException {
        DeleteIndexRequest request = new DeleteIndexRequest(index);
        AcknowledgedResponse response = client.indices().delete(request, RequestOptions.DEFAULT);
        if (response.isAcknowledged()) {
            System.out.println("delete " + index + " success");
        }
    }

    public void close() throws IOException {
        client.close();
    }

    public static void main(String[] args) throws IOException {
         // For a non-security cluster, run the following:
         RestHighLevelClient client = getClient(Arrays.asList("xx.xx.xx.xx"), 9200, "http");

        /*
         *For a security cluster with HTTPS enabled, run the following:
         *  RestHighLevelClient client = getClient(Arrays.asList("xx.xx.xx.xx", "xx.xx.xx.xx"), 9200, "https",
"user_name", "password");
         *For a security cluster with HTTPS disabled, run the following:
         *  RestHighLevelClient client = getClient(Arrays.asList("xx.xx.xx.xx", "xx.xx.xx.xx"), 9200, "http",
"user_name", "password");
         */

        ClientExample example = new ClientExample(client);
        String indexName = "my_index";

        //Create an index.
        example.create(indexName);

        //Write data.
        List<float[]> data = Arrays.asList(new float[]{1.0f, 1.0f}, new float[]{2.0f, 2.0f}, new float[]{3.0f, 3.0f});
        example.write(indexName, data);

        //Query an index.
        float[] queryVector = new float[]{1.0f, 1.0f};
        example.search(indexName, queryVector, 3);

        //Delete an index.
        example.delete(indexName);

        //Close the client.
        example.close();
    }
}
```

5.2.9.2 Elasticsearch Vector Search Sample Code (Python)

The CSS vector database is fully compatible with the native Elasticsearch Python client. With this SDK, you can use Python to easily manage the vector index lifecycle for Elasticsearch clusters, from creating vector indexes and importing

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

high-dimensional vectors at high speed, to performing millisecond-level approximate nearest neighbor (ANN) searches. The sample code in this topic is modular and ready to run. In just a few minutes, you can execute the complete workflow from data ingestion to vector search, helping you accelerate your service deployment.

Prerequisites

The Python dependency package has been installed on the client. If it is not installed, run the following commands to install it:

```
# Set the actual cluster version. 7.10 is used in this example.
pip install elasticsearch==7.10
```

Sample Code

```
from elasticsearch import Elasticsearch
from elasticsearch import helpers

# Create the Elasticsearch client.
def get_client(hosts: list, user: str = None, password: str = None):
    if user and password:
        return Elasticsearch(hosts, http_auth=(user, password), verify_certs=False, ssl_show_warn=False)
    else:
        return Elasticsearch(hosts)

# Create an index table.
def create(client: Elasticsearch, index: str):
    # Index mapping information
    index_mapping = {
        "settings": {
            "index": {
                "vector": "true",  # Enable the vector feature.
                "number_of_shards": 1,  # Set the number of index shards as needed.
                "number_of_replicas": 0  # Set the number of index replicas as needed.
            }
        },
        "mappings": {
            "properties": {
                "my_vector": {
                    "type": "vector",
                    "dimension": 2,
                    "indexing": True,
                    "algorithm": "GRAPH",
                    "metric": "euclidean"
                }
                # Other fields can be added if necessary.
            }
        }
    }
    res = client.indices.create(index=index, body=index_mapping)
    print("create index result: ", res)

# Write data.
def write(client: Elasticsearch, index: str, vecs: list, bulk_size=500):
    for i in range(0, len(vecs), bulk_size):
        actions = [
            {
                "_index": index,
                "my_vector": vec,
                # Other fields can be added if necessary.
            }
            for vec in vecs[i: i+bulk_size]
        ]
        success, errors = helpers.bulk(client, actions, request_timeout=3600)
        if errors:
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            print("write bulk failed with errors: ", errors)  # Handle the error as needed.
        else:
            print("write bulk {} docs success".format(success))
    client.indices.refresh(index=index, request_timeout=3600)

# Query a vector index.
def search(client: Elasticsearch, index: str, query: list, size: int):
    # Query statement. Select an appropriate query method.
    query_body = {
        "size": size,
        "query": {
            "vector": {
                "my_vector": {
                    "vector": query,
                    "topk": size
                }
            }
        }
    }
    res = client.search(index=index, body=query_body)
    print("search index result: ", res)

# Delete an index.
def delete(client: Elasticsearch, index: str):
    res = client.indices.delete(index=index)
    print("delete index result: ", res)

if __name__ == '__main__':
    # For a non-security cluster, run the following:
    es_client = get_client(hosts=['http://xx.xx.xx.xx:9200'])

    # For a security-mode cluster that uses HTTPS, run the following:
    # es_client = get_client(hosts=['https://xx.xx.xx.xx:9200', 'https://xx.xx.xx.xx:9200'], user='xxxxx',
password='xxxxx')

    # For a security-mode cluster with HTTPS disabled, run the following:
    # es_client = get_client(hosts=['http://xx.xx.xx.xx:9200', 'http://xx.xx.xx.xx:9200'], user='xxxxx',
password='xxxxx')

    # Test the index name.
    index_name = "my_index"

    # Create an index.
    create(es_client, index=index_name)

    # Write data.
    data = [[1.0, 1.0], [2.0, 2.0], [3.0, 3.0]]
    write(es_client, index=index_name, vecs=data)

    # Query an index.
    query_vector = [1.0, 1.0]
    search(es_client, index=index_name, query=query_vector, size=3)

    # Delete an index.
    delete(es_client, index=index_name)
```

5.2.9.3 OpenSearch Vector Search Sample Code (Java)

The CSS vector database is fully compatible with the native Python Java client. With this SDK, you can use Java to easily manage the vector index lifecycle for OpenSearch clusters, from creating vector indexes and importing high-dimensional vectors at high speed, to performing millisecond-level approximate nearest neighbor (ANN) searches. The sample code in this topic is modular and ready to run. In just a few minutes, you can execute the complete workflow from data ingestion to vector search, helping you accelerate your service deployment.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Prerequisites

Java client dependencies have been installed on the server. Sample code in Maven mode:

```
// OpenSearch 2.19.0 and OpenSearch 3.4.0 (image version 3.4.0_26.6.0_xxx or later)
<dependency>
    <groupId>org.opensearch.client</groupId>
    <artifactId>opensearch-rest-high-level-client</artifactId>
    <version>2.19.0</version>
</dependency>

// OpenSearch 1.3.6
<dependency>
    <groupId>org.opensearch.client</groupId>
    <artifactId>opensearch-rest-high-level-client</artifactId>
    <version>1.3.6</version>
</dependency>
```

Sample Code (2.19.0)

Applicable version: OpenSearch 2.19.0 and OpenSearch 3.4.0 (image version 3.4.0_26.6.0_xxx or later)

```
package org.example;

import org.apache.http.HttpEntity;
import org.apache.http.HttpHost;
import org.apache.http.HttpStatus;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.opensearch.action.admin.indices.delete.DeleteIndexRequest;
import org.opensearch.action.admin.indices.refresh.RefreshRequest;
import org.opensearch.action.bulk.BulkRequest;
import org.opensearch.action.bulk.BulkResponse;
import org.opensearch.action.index.IndexRequest;
import org.opensearch.action.search.SearchResponse;
import org.opensearch.action.support.master.AcknowledgedResponse;
import org.opensearch.client.Request;
import org.opensearch.client.RequestOptions;
import org.opensearch.client.Response;
import org.opensearch.client.RestClient;
import org.opensearch.client.RestClientBuilder;
import org.opensearch.client.RestHighLevelClient;
import org.opensearch.client.indices.CreateIndexRequest;
import org.opensearch.client.indices.CreateIndexResponse;
import org.opensearch.common.settings.Settings;
import org.opensearch.common.xcontent.XContentType;
import org.opensearch.core.xcontent.DeprecationHandler;
import org.opensearch.core.xcontent.NamedXContentRegistry;
import org.opensearch.core.xcontent.XContentParser;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
import java.io.IOException;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;

public class ClientExampleOS {
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
    private final RestHighLevelClient client;

    public ClientExampleOS(RestHighLevelClient client) {
        this.client = client;
    }

    //Create a client for accessing non-security clusters.
    public static RestHighLevelClient getClient(List<String> hosts, int port, String scheme) {
        HttpHost[] httpHosts = hosts.stream().map(p -> new HttpHost(p, port,
scheme)).toArray(HttpHost[]::new);
        return new RestHighLevelClient(RestClient.builder(httpHosts));
    }

    //Create a client for accessing security clusters.
    public static RestHighLevelClient getClient(List<String> hosts, int port, String scheme, String user, String
password) {
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials(user,
password));
        SSLContext sc = null;
        try {
            sc = SSLContext.getInstance("SSL");
            sc.init(null, trustAllCerts, new SecureRandom());
        } catch (KeyManagementException | NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        HttpHost[] httpHosts = hosts.stream().map(p -> new HttpHost(p, port,
scheme)).toArray(HttpHost[]::new);
        final SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc,
NoopHostnameVerifier.INSTANCE);
        RestClientBuilder builder = RestClient.builder(httpHosts).setHttpClientConfigCallback(httpClientBuilder
-> {
            httpClientBuilder.disableAuthCaching();
            httpClientBuilder.setSSLStrategy(sessionStrategy);
            return httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
        });
        return new RestHighLevelClient(builder);
    }

    public static TrustManager[] trustAllCerts = new TrustManager[] {
            new X509TrustManager() {
                @Override
                public void checkClientTrusted(X509Certificate[] chain, String authType) {
                }
                @Override
                public void checkServerTrusted(X509Certificate[] chain, String authType) {
                }
                @Override
                public X509Certificate[] getAcceptedIssuers() {
                    return null;
                }
            }
    };

    # Create an index.
    public void create(String index) throws IOException {
        CreateIndexRequest request = new CreateIndexRequest(index);
        request.settings(Settings.builder()
                .put("index.vector", true) // Enable the vector feature.
                .put("index.number_of_shards", 1) //Set the number of index shards as needed.
                .put("index.number_of_replicas", 0) //Set the number of index replicas as needed.
        );
        String mapping =
                "{" +
                "  \"properties\": {" +
                "    \"my_vector\": {" +
                "      \"type\": \"vector\"," +      // Set this field as the vector type.
                "      \"indexing\": \"true\"," +    // Enable index acceleration.
                "      \"dimension\": \"2\"," +      // Vector dimensions
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            "      \"metric\": \"euclidean\"," + // Similarity metric
            "      \"algorithm\": \"GRAPH\"" +   // Index algorithm
            "    }" +
            "  }" +
            "}";
    request.mapping(mapping, XContentType.JSON);
    CreateIndexResponse response = client.indices().create(request, RequestOptions.DEFAULT);
    if (response.isAcknowledged()) {
        System.out.println("create " + index + " success");
    }
}

// Write data. You are advised to keep the batch size under 500 vectors.
public void write(String index, List<float[]> vectors) throws IOException {
    BulkRequest request = new BulkRequest();
    for (float[] vec : vectors) {
        request.add(new IndexRequest(index).source("my_vector", vec));
    }

    BulkResponse response = client.bulk(request, RequestOptions.DEFAULT);
    if (response.hasFailures()) {
        System.out.println(response.buildFailureMessage());
    } else {
        System.out.println("write bulk to " + index + " success");
    }

    // Optional. OpenSearch will refresh it by default.
    client.indices().refresh(new RefreshRequest(index), RequestOptions.DEFAULT);
}

// Query vectors.
public void search(String index, float[] query, int size) throws IOException {
    String queryFormat =
            "{\n" +
            "  \"size\":%d,\n" +
            "  \"query\": {\n" +
            "    \"vector\": {\n" +
            "      \"my_vector\": {\n" +  // Query the vector field name.
            "        \"vector\": %s,\n" +
            "        \"topk\":%d\n" +
            "      }\n" +
            "    }\n" +
            "  }\n" +
            "}";
    String body = String.format(queryFormat, size, Arrays.toString(query), size);
    Request request = new Request("POST", index + "/_search");
    request.setJsonEntity(body);
    Response response = client.getLowLevelClient().performRequest(request);
    if (response.getStatusLine().getStatusCode() != HttpStatus.SC_OK) {
        System.out.println(response.getEntity()); //Handle the error based on service requirements.
        return;
    }
    // Process the normally returned result based on service requirements.
    HttpEntity entity = response.getEntity();
    XContentType xContentType = XContentType.JSON;
    XContentParser parser = xContentType.xContent().createParser(NamedXContentRegistry.EMPTY,
            DeprecationHandler.IGNORE_DEPRECATIONS, entity.getContent());
    SearchResponse searchResponse = SearchResponse.fromXContent(parser);
    System.out.println(searchResponse);
}

// Delete an index.
public void delete(String index) throws IOException {
    DeleteIndexRequest request = new DeleteIndexRequest(index);
    AcknowledgedResponse response = client.indices().delete(request, RequestOptions.DEFAULT);
    if (response.isAcknowledged()) {
        System.out.println("delete " + index + " success");
    }
}
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
    public void close() throws IOException {
        client.close();
    }

    public static void main(String[] args) throws IOException {
         // For a non-security mode cluster, run the following:
        RestHighLevelClient client = getClient(Arrays.asList("xx.xx.xx.xx"), 9200, "http");

        /*
         *For a security-mode cluster with HTTPS enabled, run the following:
         *  RestHighLevelClient client = getClient(Arrays.asList("xx.xx.xx.xx", "xx.xx.xx.xx"), 9200, "https",
"user_name", "password");
         *For a security-mode cluster with HTTPS disabled, run the following:
         *  RestHighLevelClient client = getClient(Arrays.asList("xx.xx.xx.xx", "xx.xx.xx.xx"), 9200, "http",
"user_name", "password");
         */

        ClientExampleOS example = new ClientExampleOS(client);
        String indexName = "my_index";

        //Create an index.
        example.create(indexName);

        //Write data.
        List<float[]> data = Arrays.asList(new float[]{1.0f, 1.0f}, new float[]{2.0f, 2.0f}, new float[]{3.0f, 3.0f});
        example.write(indexName, data);

        //Query an index.
        float[] queryVector = new float[]{1.0f, 1.0f};
        example.search(indexName, queryVector, 3);

        //Delete an index.
        example.delete(indexName);

        //Close the client.
        example.close();
    }
}
```

Sample Code (1.3.6)

Applicable version: OpenSearch 1.3.6

```
package org.example;

import org.apache.http.HttpEntity;
import org.apache.http.HttpHost;
import org.apache.http.HttpStatus;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.opensearch.action.admin.indices.delete.DeleteIndexRequest;
import org.opensearch.action.admin.indices.refresh.RefreshRequest;
import org.opensearch.action.bulk.BulkRequest;
import org.opensearch.action.bulk.BulkResponse;
import org.opensearch.action.index.IndexRequest;
import org.opensearch.action.search.SearchResponse;
import org.opensearch.action.support.master.AcknowledgedResponse;
import org.opensearch.client.Request;
import org.opensearch.client.RequestOptions;
import org.opensearch.client.Response;
import org.opensearch.client.RestClient;
import org.opensearch.client.RestClientBuilder;
import org.opensearch.client.RestHighLevelClient;
import org.opensearch.client.indices.CreateIndexRequest;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
import org.opensearch.client.indices.CreateIndexResponse;
import org.opensearch.common.settings.Settings;
import org.opensearch.common.xcontent.DeprecationHandler;
import org.opensearch.common.xcontent.NamedXContentRegistry;
import org.opensearch.common.xcontent.XContentParser;
import org.opensearch.common.xcontent.XContentType;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
import java.io.IOException;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;

public class ClientExampleOS {
    private final RestHighLevelClient client;

    public ClientExampleOS(RestHighLevelClient client) {
        this.client = client;
    }

    //Create a client for accessing non-security clusters.
    public static RestHighLevelClient getClient(List<String> hosts, int port, String scheme) {
        HttpHost[] httpHosts = hosts.stream().map(p -> new HttpHost(p, port,
scheme)).toArray(HttpHost[]::new);
        return new RestHighLevelClient(RestClient.builder(httpHosts));
    }

    //Create a client for accessing security clusters.
    public static RestHighLevelClient getClient(List<String> hosts, int port, String scheme, String user, String
password) {
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials(user,
password));
        SSLContext sc = null;
        try {
            sc = SSLContext.getInstance("SSL");
            sc.init(null, trustAllCerts, new SecureRandom());
        } catch (KeyManagementException | NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        HttpHost[] httpHosts = hosts.stream().map(p -> new HttpHost(p, port,
scheme)).toArray(HttpHost[]::new);
        final SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc,
NoopHostnameVerifier.INSTANCE);
        RestClientBuilder builder = RestClient.builder(httpHosts).setHttpClientConfigCallback(httpClientBuilder
-> {
            httpClientBuilder.disableAuthCaching();
            httpClientBuilder.setSSLStrategy(sessionStrategy);
            return httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
        });
        return new RestHighLevelClient(builder);
    }

    public static TrustManager[] trustAllCerts = new TrustManager[] {
            new X509TrustManager() {
                @Override
                public void checkClientTrusted(X509Certificate[] chain, String authType) {
                }
                @Override
                public void checkServerTrusted(X509Certificate[] chain, String authType) {
                }
                @Override
                public X509Certificate[] getAcceptedIssuers() {
                    return null;
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
        }
    }
};

# Create an index.
public void create(String index) throws IOException {
    CreateIndexRequest request = new CreateIndexRequest(index);
    request.settings(Settings.builder()
            .put("index.vector", true) // Enable the vector feature.
            .put("index.number_of_shards", 1) //Set the number of index shards as needed.
            .put("index.number_of_replicas", 0) //Set the number of index replicas as needed.
    );
    String mapping =
            "{" +
            "  \"properties\": {" +
            "    \"my_vector\": {" +
            "      \"type\": \"vector\"," +      // Set this field as the vector type.
            "      \"indexing\": \"true\"," +    // Enable index acceleration.
            "      \"dimension\": \"2\"," +      // Vector dimensions
            "      \"metric\": \"euclidean\"," + // Similarity metric
            "      \"algorithm\": \"GRAPH\"" +   // Index algorithm
            "    }" +
            "  }" +
            "}";
    request.mapping(mapping, XContentType.JSON);
    CreateIndexResponse response = client.indices().create(request, RequestOptions.DEFAULT);
    if (response.isAcknowledged()) {
        System.out.println("create " + index + " success");
    }
}

// Write data. You are advised to keep the batch size under 500 vectors.
public void write(String index, List<float[]> vectors) throws IOException {
    BulkRequest request = new BulkRequest();
    for (float[] vec : vectors) {
        request.add(new IndexRequest(index).source("my_vector", vec));
    }

    BulkResponse response = client.bulk(request, RequestOptions.DEFAULT);
    if (response.hasFailures()) {
        System.out.println(response.buildFailureMessage());
    } else {
        System.out.println("write bulk to " + index + " success");
    }

    // Optional. OpenSearch will refresh it by default.
    client.indices().refresh(new RefreshRequest(index), RequestOptions.DEFAULT);
}

// Query vectors.
public void search(String index, float[] query, int size) throws IOException {
    String queryFormat =
            "{\n" +
            "  \"size\":%d,\n" +
            "  \"query\": {\n" +
            "    \"vector\": {\n" +
            "      \"my_vector\": {\n" +  // Query the vector field name.
            "        \"vector\": %s,\n" +
            "        \"topk\":%d\n" +
            "      }\n" +
            "    }\n" +
            "  }\n" +
            "}";
    String body = String.format(queryFormat, size, Arrays.toString(query), size);
    Request request = new Request("POST", index + "/_search");
    request.setJsonEntity(body);
    Response response = client.getLowLevelClient().performRequest(request);
    if (response.getStatusLine().getStatusCode() != HttpStatus.SC_OK) {
        System.out.println(response.getEntity()); //Handle the error based on service requirements.
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
            return;
        }
        // Process the normally returned result based on service requirements.
        HttpEntity entity = response.getEntity();
        XContentType xContentType = XContentType.fromMediaTypeOrFormat("application/json");
        XContentParser parser = xContentType.xContent().createParser(NamedXContentRegistry.EMPTY,
                DeprecationHandler.IGNORE_DEPRECATIONS, entity.getContent());
        SearchResponse searchResponse = SearchResponse.fromXContent(parser);
        System.out.println(searchResponse);
    }

    // Delete an index.
    public void delete(String index) throws IOException {
        DeleteIndexRequest request = new DeleteIndexRequest(index);
        AcknowledgedResponse response = client.indices().delete(request, RequestOptions.DEFAULT);
        if (response.isAcknowledged()) {
            System.out.println("delete " + index + " success");
        }
    }

    public void close() throws IOException {
        client.close();
    }

    public static void main(String[] args) throws IOException {
         // For a non-security mode cluster, run the following:
        RestHighLevelClient client = getClient(Arrays.asList("xx.xx.xx.xx"), 9200, "http");

        /*
         *For a security-mode cluster with HTTPS enabled, run the following:
         *  RestHighLevelClient client = getClient(Arrays.asList("xx.xx.xx.xx", "xx.xx.xx.xx"), 9200, "https",
"user_name", "password");
         *For a security-mode cluster with HTTPS disabled, run the following:
         *  RestHighLevelClient client = getClient(Arrays.asList("xx.xx.xx.xx", "xx.xx.xx.xx"), 9200, "http",
"user_name", "password");
         */

        ClientExampleOS example = new ClientExampleOS(client);
        String indexName = "my_index";

        //Create an index.
        example.create(indexName);

        //Write data.
        List<float[]> data = Arrays.asList(new float[]{1.0f, 1.0f}, new float[]{2.0f, 2.0f}, new float[]{3.0f, 3.0f});
        example.write(indexName, data);

        //Query an index.
        float[] queryVector = new float[]{1.0f, 1.0f};
        example.search(indexName, queryVector, 3);

        //Delete an index.
        example.delete(indexName);

        //Close the client.
        example.close();
    }
}
```

5.2.9.4 OpenSearch Vector Search Sample Code (Python)

The CSS vector database is fully compatible with the native OpenSearch Python client. With this SDK, you can use Python to easily manage the vector index lifecycle for OpenSearch clusters, from creating vector indexes and importing high-dimensional vectors at high speed, to performing millisecond-level approximate nearest neighbor (ANN) searches. The sample code in this topic is modular and

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

ready to run. In just a few minutes, you can execute the complete workflow from data ingestion to vector search, helping you accelerate your service deployment.

Prerequisites

The Python dependency package has been installed on the client. If it is not installed, run the following command to install it:

```
# OpenSearch 3.4.0 (image version 3.4.0_26.6.0_xxx or later)
pip install opensearch-py==3.2.0

# OpenSearch 2.19.0
pip install opensearch-py==2.8.0

# OpenSearch 1.3.6
pip install opensearch-py==1.1.0
```

Sample Code

```
from opensearchpy import OpenSearch

# Create a client.
def get_client(hosts: list, user: str = None, password: str = None):
    if user and password:
        return OpenSearch(hosts, http_auth=(user, password), verify_certs=False, ssl_show_warn=False)
    else:
        return OpenSearch(hosts)

# Create an index table.
def create(client: OpenSearch, index: str):
    # Index mapping information
    index_mapping = {
        "settings": {
            "index": {
                "vector": "true",  # Enable the vector feature.
                "number_of_shards": 1,  # Set the number of index shards as needed.
                "number_of_replicas": 0  # Set the number of index replicas as needed.
            }
        },
        "mappings": {
            "properties": {
                "my_vector": {
                    "type": "vector",
                    "dimension": 2,
                    "indexing": True,
                    "algorithm": "GRAPH",
                    "metric": "euclidean"
                }
                # Other fields can be added if necessary.
            }
        }
    }
    res = client.indices.create(index=index, body=index_mapping)
    print("create index result: ", res)

# Write data.
def write(client: OpenSearch, index: str, vecs: list, bulk_size=500):
    for i in range(0, len(vecs), bulk_size):
        actions = ""
        for vec in vecs[i: i + bulk_size]:
            actions += '{"index": {"_index": "%s"}}\n' % index
            actions += '{"my_vector": %s}\n' % str(vec)
        client.bulk(body=actions, request_timeout=3600)
    client.indices.refresh(index=index, request_timeout=3600)
    print("write index success!")

# Query a vector index.
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
def search(client: OpenSearch, index: str, query: list, size: int):
    # Query statement. Select an appropriate query method.
    query_body = {
        "size": size,
        "query": {
            "vector": {
                "my_vector": {
                    "vector": query,
                    "topk": size
                }
            }
        }
    }
    res = client.search(index=index, body=query_body)
    print("search index result: ", res)

# Delete an index.
def delete(client: OpenSearch, index: str):
    res = client.indices.delete(index=index)
    print("delete index result: ", res)

if __name__ == '__main__':
    os_client = get_client(hosts=['http://xx.xx.xx.xx:9200'])

    # For a security-mode cluster that uses HTTPS, run the following:
    # os_client = get_client(hosts=['https://xx.xx.xx.xx:9200', 'https://xx.xx.xx.xx:9200'], user='xxxxx',
password='xxxxx')

    # For a security-mode cluster with HTTPS disabled, run the following:
    # os_client = get_client(hosts=['http://xx.xx.xx.xx:9200', 'http://xx.xx.xx.xx:9200'], user='xxxxx',
password='xxxxx')

    # Test the index name.
    index_name = "my_index"

    # Create an index.
    create(os_client, index=index_name)

    # Write data.
    data = [[1.0, 1.0], [2.0, 2.0], [3.0, 3.0]]
    write(os_client, index=index_name, vecs=data)

    # Query an index.
    query_vector = [1.0, 1.0]
    search(os_client, index=index_name, query=query_vector, size=3)

    # Delete an index.
    delete(os_client, index=index_name)
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

6 CSS Resource Monitoring and Alarms

For enterprises using CSS for data search and analytics, service continuity and system stability are critical. Sudden surges in data volumes or query requests can lead to performance bottlenecks and availability risks, including high CPU usage, disk exhaustion, and JVM heap memory exhaustion. To enable you to intuitively monitor cluster health, identify performance trends, and receive notifications even before issues occur, CSS integrates Cloud Eye to provide monitoring dashboards (**Monitoring**) as well as metrics and alarms on the **Dashboard** page. With intuitive visualizations and proactive alerting, you can track resource usage and cluster performance in real time, ensuring greater operational resilience.

Method 1: Monitoring Trends on the Monitoring Page (Monitoring Dashboards)

CSS's out-of-the-box monitoring dashboards offer a one-stop resource monitoring solution. By integrating cloud resource usage data and core service metrics, the monitoring dashboards quickly build unified, intuitive monitoring views for the service, facilitating cluster health inspection and trend analysis.

The monitoring dashboards depend on the following:

- Metric collection: Collects key resource metrics for the cloud service.
- Preset dashboards: Displays prebuilt monitoring dashboards for the cloud service. There is no need to manually configure metrics.
- Data visualization: Intuitively shows resource usage, service status, and trends using configurable charts.

Constraints:

- Region: The monitoring dashboards are available only in some regions. If they are unavailable in your region, the console will show a message informing you of that.
- Configuration: The **Monitoring** page on the CSS console provides pre-built dashboards only. It does not support custom metrics. To use a custom dashboard, click **More** in the upper-right corner to go to the Cloud Eye console to configure the dashboard.
- Monitoring scope: Only resources in the current region can be monitored. Cross-region monitoring is not supported.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Procedure

1. Log in to the **CSS management console**.
2. In the navigation pane, choose **Monitoring** to view the preset monitoring dashboards.

   Monitoring dashboards cover the following core metrics:

   - Resource metrics: cluster health status, average indexing speed, average query speed, maximum CPU usage, maximum JVM heap usage, maximum node loads, number of shards, maximum disk usage, total number of tasks in the write queue, and total number of tasks in the search queue.
   - Data metrics: current, maximum, minimum, average, and sum values.
   - Trend analysis: week-on-week (this week vs. last week) and day-on-day (today vs. yesterday) comparisons.

Method 2: Checking Key Metrics on the Dashboard Page

The **Dashboard** page on the CSS console provides a lightweight monitoring view where you can quickly check cluster health.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Dashboard**.
3. In the **Monitoring** area, check the metrics there.

   This area shows the thumbnails of selected metrics.

   Figure 6-1 Metrics

   NOTE

   The metrics of deleted resources are retained for one hour to allow traceability even after deletion.

4. (Optional) Create a custom view.

   a. Click **Configure**. In the displayed dialog box, select the metrics you want to display (for example, **Cluster Health Status**), and set **Rollup** and **Chart Type**.

   b. Click **OK** to save the custom view.

5. Click **More** to go to the Cloud Eye console and see more information about CSS.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Managing Resource Alarms

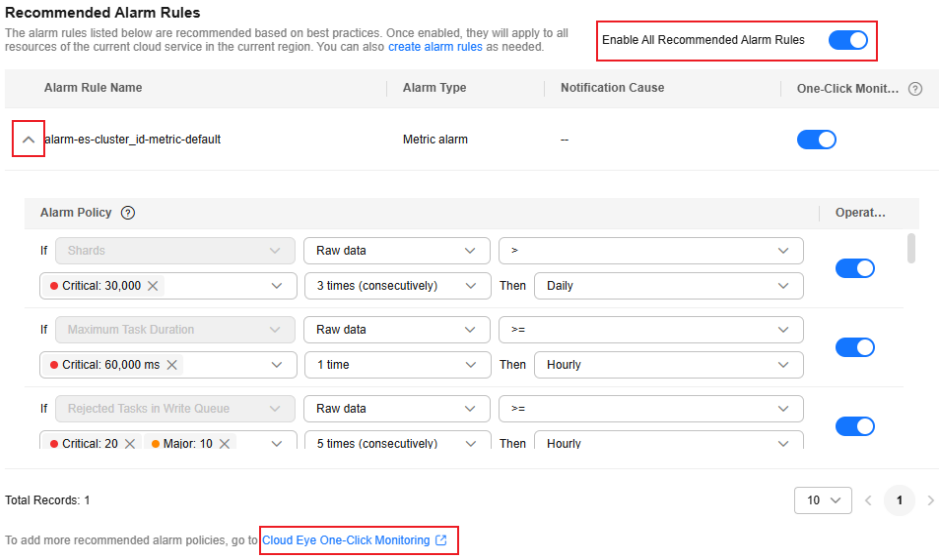
The **Dashboard** page on the CSS console provides a lightweight alarm management view where you quickly check generated alarms. You can set custom alarm thresholds (for example, maximum CPU usage > 80%). When a preset threshold is crossed, the system sends an SMS or email to notify the O&M team promptly.

Configure alarm rules:

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Dashboard**.
3. In the upper right corner of the **Alarm Management** area, click **Create Alarm Rule**.
4. In the displayed dialog box, configure alarm rules.
   a. Enable recommended alarm rules. CSS provides recommended alarm rules that, when enabled, automatically apply to all resources in the current region.

      Select **Enable All Recommended Alarm Rules** to enable the recommended alarm rules. Then you can check and modify alarm rules in the list below.
   b. (Optional) Modify recommended alarm rules. If the recommended alarm rules do not meet your needs, click **Cloud Eye One-Click Monitoring** to go to the Cloud Eye console and modify the recommended alarm rules or add new ones. For details, see **Creating an Alarm Rule and Notifications**.

Figure 6-2 Configuring alarm rules



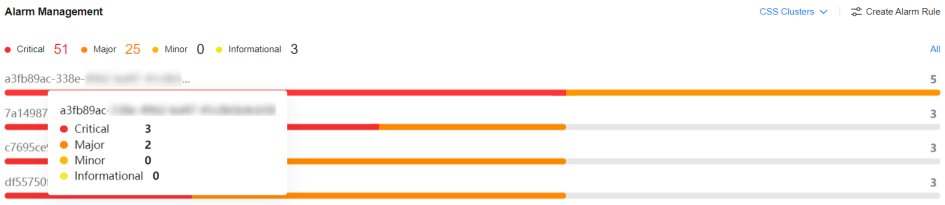
5. (Optional) Configure a notification policy. To have notifications automatically sent via SMS or email, turn on **Batch Configure Notification Policies**, and configure a notification policy.
6. Click **OK** to save the alarm rules.

Check alarm information:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Dashboard**.
3. Check alarm information in the **Alarm Management** area.

Figure 6-3 Alarm Management



- Hover the mouse pointer over a resource name to check alarm information in a floating tooltip.
- Click **All**. On the displayed **Alarm Records** page, check detailed alarm information.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.