ALIBABA CLOUD

Alibaba Cloud

Elasticsearch

Best Practices

Document Version: 20220614

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	**Danger:** Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	**Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	**Notice:** If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	**Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Overview of best practices

This topic provides an overview of the best practices of Alibaba Cloud Elasticsearch in various scenarios. You can view the best practices based on your business requirements.

Scenario	References

Elasticsearch data migration	• Data migration between Alibaba Cloud Elasticsearch clusters
	  ◦ Use the reindex API to migrate data
	  ◦ Use the reindex operation to migrate data in a multi-type index of an earlier version
	• Data migration from a self-managed Elasticsearch cluster
	  ◦ Use OSS to migrate data from a user-created Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster
	  ◦ Use Alibaba Cloud Logstash to migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster
	  ◦ Use the reindex API to migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster
	  ◦ Migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster deployed in the new network architecture
	• Data migration from third-party Elasticsearch services
	  ◦ Migrate data from an Amazon ES domain to an Alibaba Cloud Elasticsearch cluster

Data synchronization from databases	• Data synchronization from ApsaraDB RDS for MySQL
	  ◦ Select a synchronization method
	  ◦ Use DataWorks to synchronize data from a MySQL database to an Alibaba Cloud Elasticsearch cluster
	  ◦ Use DTS to synchronize MySQL data to an Alibaba Cloud Elasticsearch cluster in real time
	  ◦ Use Canal to synchronize data to an Alibaba Cloud Elasticsearch cluster
	• Data synchronization from DRDS
	  ◦ Use DataWorks to synchronize data from a DRDS database to an Elasticsearch cluster in offline mode
	• Use DTS to synchronize data from a PolarDB for MySQL database to an Alibaba Cloud Elasticsearch cluster
	• Use Monstache to synchronize data from a MongoDB database to an Alibaba Cloud Elasticsearch cluster in real time

Scenario	References
Data synchronization from big data cloud services	• Use DataWorks to synchronize data from MaxCompute to an Alibaba Cloud Elasticsearch cluster • Use Realtime Compute to process and synchronize data to an Alibaba Cloud Elasticsearch cluster • Use DataWorks to synchronize data from a Hadoop cluster to an Alibaba Cloud Elasticsearch cluster
Data migration from storage services	• Migrate documents from a Solr cluster to an Alibaba Cloud Elasticsearch cluster
Use of ES-Hadoop for data synchronization	• Use ES-Hadoop to enable Hive to write data to and read data from Alibaba Cloud Elasticsearch • Use ES-Hadoop to write HDFS data to Elasticsearch • Use ES-Hadoop to enable Apache Spark to write data to and read data from Alibaba Cloud Elasticsearch
Log data collection and analysis	• Overview of log synchronization and analysis • Use user-created Filebeat to collect MySQL logs • Use Alibaba Cloud Elasticsearch and Rsbeat to analyze Redis slow logs in real time
Sever data collection	• Overview of best practices for server data collection • Data collection for Alibaba Cloud Elasticsearch • Use user-created Metricbeat to collect system metrics • Use SkyWalking to implement end-to-end monitoring on Alibaba Cloud Elasticsearch • Use Uptime to monitor Alibaba Cloud Elasticsearch clusters in real time

Scenario	References

Cluster management	• Overview of cluster management
	• Hot and cold data separation and lifecycle management
	  ◦ Use ILM to manage Heartbeat indexes
	  ◦ Use ILM to separate hot data from cold data
	• Application of X-Pack advanced features
	  ◦ Use the CCR feature to migrate data
	  ◦ Use X-Pack to configure LDAP authentication
	  ◦ Use the RBAC mechanism provided by Elasticsearch X-Pack to implement access control
	  ◦ Configure AD user authentication
	• Integrated monitoring
	  ◦ Use Elastic Stack to implement integrated monitoring for containers in Kubernetes
	• Data management and visualization
	  ◦ Use Terraform to manage Alibaba Cloud Elasticsearch clusters
	  ◦ Use Curator
	  ◦ Use the rollup mechanism to summarize traffic data
	  ◦ Use Cerebro to access an Alibaba Cloud Elasticsearch cluster
	• Notification of alerts for clusters
	  ◦ Configure a DingTalk chatbot to receive alert notifications from X-Pack Watcher

2.Elasticsearch migration

2.1. Migrate data between Alibaba Cloud Elasticsearch clusters

2.1.1. Use the reindex API to migrate data

You can use the reindex API to migrate data between Elasticsearch clusters. This topic describes the migration procedure in detail.

Background information

You can use the reindex API to perform the following operations:

- Migrate data between Elasticsearch clusters.
- Reindex the data in an index whose shards are inappropriately configured. For example, the data volume is large, but only a few shards are configured for the index. This slows down data write operations.
- Replicate the data in an index if the index stores large volumes of data and you want to modify the mapping configuration of the index. This operation requires only a short period of time. You can also insert the data into a new index. However, this operation is time-consuming.

> **Note** After you define the mapping configuration for an index in an Elasticsearch cluster and insert data into the index, you cannot modify the mapping configuration.

Prerequisites

- Two Alibaba Cloud Elasticsearch clusters are created. One is used as a local cluster, and the other is used as a remote cluster.

  For more information, see Create an Alibaba Cloud Elasticsearch cluster. The two clusters must belong to the same virtual private cloud (VPC) and vSwitch. In this example, an Elasticsearch V6.7.0 cluster is used as the local cluster, and an Elasticsearch V6.3.2 cluster is used as the remote cluster.

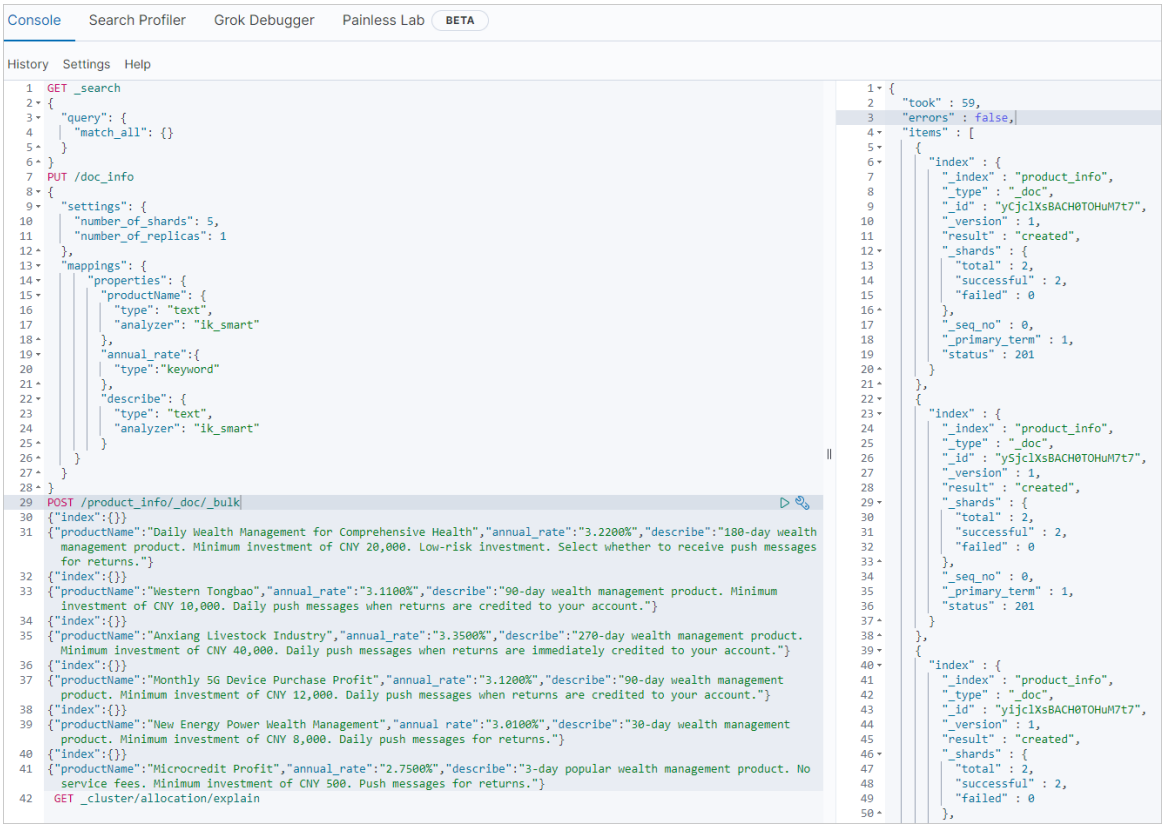
- Test data is prepared.
  - Local cluster

    Create a destination index in the local cluster.

```
PUT dest
{
  "settings": {
    "number_of_shards": 5,
    "number_of_replicas": 1
  }
}
```

- Remote cluster

  Prepare the data that you want to migrate. In this example, the data in the "Quick start" topic is used. For more information, see Quick start.



> **Notice** If you want to use a cluster that runs Elasticsearch V7.0 or later as a remote cluster, you must set the index type to _doc.

Limits

The network architecture of Alibaba Cloud Elasticsearch was adjusted in October 2020. Alibaba Cloud Elasticsearch clusters created before October 2020 are deployed in the original network architecture. Alibaba Cloud Elasticsearch cluster created in October 2020 or later are deployed in the new network architecture. Due to the adjustment of the network architecture, you cannot use the reindex API to migrate data between clusters in some scenarios. The following table describes the scenarios and provides data migration solutions in these scenarios.

Scenario	Network architecture	Support for the reindex API	Solution
	Both clusters are deployed in the original network architecture.	Yes	For more information, see Use the reindex API to migrate data.

Migrate data between Alibaba Cloud Elasticsearch clusters	Both clusters are deployed in the new network architecture.	No	Use OSS or Logstash to migrate data between the clusters. For more information, see Use OSS to migrate data from a user-created Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster and Use Alibaba Cloud Logstash to migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster.
	One is deployed in the original network architecture, and the other is deployed in the new network architecture.	No	
Migrate data from a self-managed Elasticsearch cluster that runs on ECS instances to an Alibaba Cloud Elasticsearch cluster	The Alibaba Cloud Elasticsearch cluster is deployed in the original network architecture.	Yes	For more information, see Use the reindex API to migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster.
	The Alibaba Cloud Elasticsearch cluster is deployed in the new network architecture.	Yes	Use the PrivateLink service to establish a private connection between the VPC where the Alibaba Cloud Elasticsearch cluster resides and the VPC where the self-managed Elasticsearch cluster resides. Then, use the domain name of the endpoint you obtained and the reindex API to migrate data between the clusters. For more information, see Migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster deployed in the new network architecture. **Note** Only some regions support PrivateLink. For more information, see Regions and zones that support PrivateLink. If the zone where your Alibaba Cloud Elasticsearch cluster resides does not support PrivateLink, you cannot use the reindex API to migrate data between the clusters.

Procedure

1.
2.
3.
4. Configure a reindex whitelist for the local cluster.
   i. In the left-side navigation pane of the page that appears, click **Cluster Configuration**.
   ii. On the page that appears, click **Modify Configuration** on the right side of **YML Configuration**.
   iii. In the **Other Configurations** field of the **YML File Configuration** panel, configure a reindex whitelist.

   For more information about how to configure a reindex whitelist, see Configure a remote reindex whitelist.

   - If the remote cluster is a single-zone cluster, specify the reindex whitelist in the format of <Domain name of the cluster>:9200.

   ```
   1  reindex.remote.whitelist:
      ["es-cn-09k1rgid9000g    .elasticsearch.
      aliyuncs.com:9200"]
   ```

   ```
   reindex.remote.whitelist: ["es-cn-09k1rgid9000g****.elasticsearch.aliyuncs.com:92
   00"]
   ```

   - If the remote cluster is a multi-zone cluster, the reindex whitelist must contain the IP addresses of all the data nodes in the cluster and the port number of the cluster.

   ```
   1  reindex.remote.whitelist:
      ["es-cn-09k1rgid9000g    .elasticsearch.
      aliyuncs.com:9200"]
   ```

   ```
   reindex.remote.whitelist: ["10.0.xx.xx:9200","10.0.xx.xx:9200","10.0.xx.xx:9200",
   "10.15.xx.xx:9200","10.15.xx.xx:9200","10.15.xx.xx:9200"]
   ```

   > **Note** You can obtain the IP addresses of all the data nodes in a cluster from the **Node Visualization** tab on the **Basic Information** page of the cluster. For more information, see View the basic information of nodes.

   iv. Select **This operation will restart the cluster. Continue?** and click **OK**.
5. In the local cluster, call the reindex API to reindex data.

   Log on to the Kibana console of the local cluster and run the following command to reindex data:

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "http://es-cn-09k1rgid9000g****.elasticsearch.aliyuncs.com:9200",
      "username": "elastic",
      "password": "your_password"
    },
    "index": "product_info",
    "query": {
      "match": {
        "productName": "Wealth management"
      }
    }
  },
  "dest": {
    "index": "dest"
  }
}
```

Part	Parameter	Description
	host	The URL that is used to connect to the remote cluster. The URL must contain the protocol, domain name, and port number. Example: https://otherhost:9200. ◦ If the remote cluster is a single-zone cluster, the value of the host parameter must be in the format of http://<Domain name of the cluster>:9200. **Note** You can obtain the domain name from the Basic Information page of the cluster. For more information, see View the basic information of a cluster. ◦ If the remote cluster is a multi-zone cluster, the value of the host parameter must be in the format of http://<IP address of a data node in the cluster>:9200.

Part	Parameter	Description
source	username	The username that is used to connect to the remote cluster. This parameter is optional. It is required only if basic authentication needs to be performed on requests that are sent to the remote cluster. The default username that is used to connect to Alibaba Cloud Elasticsearch clusters is elastic. **Notice** ◦ For security purposes, we recommend that you use HTTPS to send requests if basic authentication needs to be performed. Otherwise, the required password is transmitted in plaintext. ◦ For Alibaba Cloud Elasticsearch clusters, you can use HTTPS in host only after you enable the protocol for the clusters.
	password	The password that is used to connect to the remote cluster. The password is specified when you create the cluster. If you forget the password, you can reset it. For more information about the procedure and precautions for resetting the password, see Reset the access password for an Elasticsearch cluster.
	index	The source index in the remote cluster.
	query	Specifies the data that you want to migrate. For more information, see Reindex API.
dest	index	The destination index in the local cluster.

> **Note** When you reindex data from a remote cluster, manual slicing and automatic slicing are not supported for the data. For more information, see Manual slicing and Automatic slicing.

If the command is successfully run, the following result is returned:

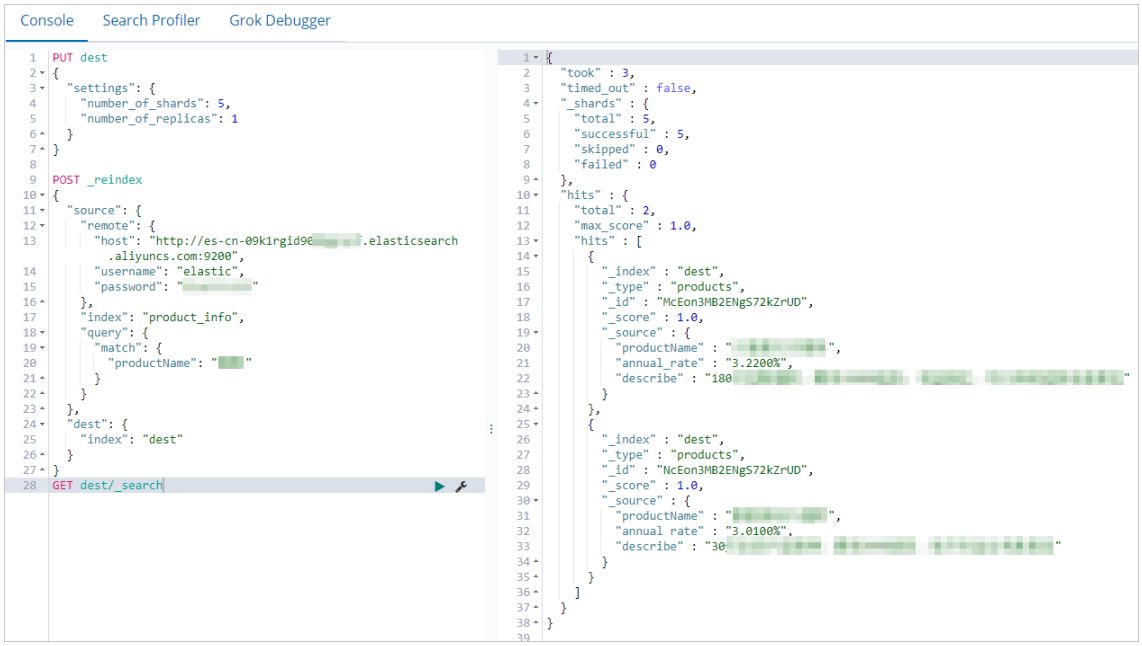
```
{
  "took" : 51,
  "timed_out" : false,
  "total" : 2,
  "updated" : 2,
  "created" : 0,
  "deleted" : 0,
  "batches" : 1,
  "version_conflicts" : 0,
  "noops" : 0,
  "retries" : {
    "bulk" : 0,
    "search" : 0
  },
  "throttled_millis" : 0,
  "requests_per_second" : -1.0,
  "throttled_until_millis" : 0,
  "failures" : [ ]
}
```

6. Run the following command to view the migrated data:

```
GET dest/_search
```

The following figures show the command outputs.

- Single-zone cluster



- Multi-zone cluster

Summary

The configurations that are required to migrate data from a single-zone cluster are similar to the configurations that are required to migrate data from a multi-zone cluster. The following table lists differences.

Cluster type	Configuration of the reindex whitelist	Configuration of the host parameter
Single-zone cluster	Domain name of the cluster:9200	https://Domain name of the cluster:9200
Multi-zone cluster	Combination of the IP addresses of all the data nodes in the cluster and the port number of the cluster	https://IP address of a data node in the cluster:9200

Additional information

When you use the reindex API to reindex data, you can specify a batch size and timeout periods.

- Batch size

  A remote Elasticsearch cluster uses a heap to cache index data. The default batch size is 100 MB. If an index in the remote cluster contains large documents, you must change the batch size to a smaller value.

  In the following example, size is set to 10.

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "http://otherhost:9200"
    },
    "index": "source",
    "size": 10,
    "query": {
      "match": {
        "test": "data"
      }
    }
  },
  "dest": {
    "index": "dest"
  }
}
```

- Timeout periods

  Use socket_timeout to specify a timeout period for socket reads. The default value of socket_timeout is 30s. Use connect_timeout to specify a timeout period for connections. The default value of connect_timeout is 1s.

  In the following example, socket_timeout is set to 1m, and connect_timeout is set to 10s.

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "http://otherhost:9200",
      "socket_timeout": "1m",
      "connect_timeout": "10s"
    },
    "index": "source",
    "query": {
      "match": {
        "test": "data"
      }
    }
  },
  "dest": {
    "index": "dest"
  }
}
```

2.1.2. Use the reindex operation to migrate data in a multi-type index of an earlier version

This topic describes how to use the reindex operation to migrate data from a multi-type index to a single-type index. The multi-type index is on an Alibaba Cloud Elasticsearch V5.X cluster. The single-type index is on an Alibaba Cloud Elasticsearch V6.X cluster.

Limits

The network architecture of Alibaba Cloud Elasticsearch was adjusted in October 2020. Alibaba Cloud Elasticsearch clusters created before October 2020 are deployed in the original network architecture. Alibaba Cloud Elasticsearch cluster created in October 2020 or later are deployed in the new network architecture. Due to the adjustment of the network architecture, you cannot use the reindex API to migrate data between clusters in some scenarios. The following table describes the scenarios and provides data migration solutions in these scenarios.

Scenario	Network architecture	Support for the reindex API	Solution
Migrate data between Alibaba Cloud Elasticsearch clusters	Both clusters are deployed in the original network architecture.	Yes	For more information, see Use the reindex API to migrate data.
	Both clusters are deployed in the new network architecture.	No	Use OSS or Logstash to migrate data between the clusters. For more information, see Use OSS to migrate data from a user-created Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster and Use Alibaba Cloud Logstash to migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster.
	One is deployed in the original network architecture, and the other is deployed in the new network architecture.	No	
Migrate data	The Alibaba Cloud Elasticsearch cluster is deployed in the original network architecture.	Yes	For more information, see Use the reindex API to migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster.

Scenario	Network architecture	Support for the reindex API	Solution
Migrate data from a self-managed Elasticsearch cluster that runs on ECS instances to an Alibaba Cloud Elasticsearch cluster	The Alibaba Cloud Elasticsearch cluster is deployed in the new network architecture.	Yes	Use the PrivateLink service to establish a private connection between the VPC where the Alibaba Cloud Elasticsearch cluster resides and the VPC where the self-managed Elasticsearch cluster resides. Then, use the domain name of the endpoint you obtained and the reindex API to migrate data between the clusters. For more information, see Migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster deployed in the new network architecture. **Note** Only some regions support PrivateLink. For more information, see Regions and zones that support PrivateLink. If the zone where your Alibaba Cloud Elasticsearch cluster resides does not support PrivateLink, you cannot use the reindex API to migrate data between the clusters.

Procedure

1. Preparations

   Create an Elasticsearch V5.X cluster, an Elasticsearch V6.X cluster, and a Logstash cluster in the same virtual private cloud (VPC).

   - The Elasticsearch clusters are used to store index data.
   - The Logstash cluster is used to migrate processed data based on pipelines.

2. Step 1: Convert the multi-type index into one or more single-type indexes

   Use the reindex operation to convert the multi-type index on the Elasticsearch V5.X cluster into one or more single-type indexes. You can use one of the following methods to implement the conversion:

   - Combine types: Call the reindex operation with the script condition specified to combine the types of the index.
   - Split the index: Call the reindex operation to split the index into multiple indexes. Each of these indexes has only one type.

3. Step 2: Use Logstash to migrate data

Use the Logstash cluster to migrate the processed index data to the Elasticsearch V6.X cluster.

4. Step 3: View the data migration results

   View the migrated data in the Kibana console.

Preparations

1. Create an Elasticsearch V5.5.3 cluster and an Elasticsearch V6.7.0 cluster. Then, create a multi-type index on the Elasticsearch V5.5.3 cluster and insert data into the index.

   For more information about how to create an Elasticsearch cluster, see Create an Alibaba Cloud Elasticsearch cluster.

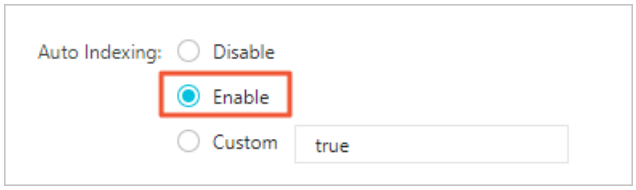
2. Create an Logstash cluster in the VPC where the Elasticsearch clusters reside.

   For more information, see Step 1: Create a Logstash cluster.

Step 1: Convert the multi-type index into one or more single-type indexes

In the following steps, the types of the index are combined to convert the index into one single-type index.

1. Enable the Auto Indexing feature for the Elasticsearch V5.5.3 cluster.
   i. Log on to the Elasticsearch console.
   ii. In the left-side navigation pane, click **Elasticsearch Clusters**.
   iii. In the top navigation bar, select a resource group and a region.
   iv. On the **Clusters** page, find the Elasticsearch V5.5.3 cluster and click its ID.
   v. In the left-side navigation pane of the page that appears, click **Cluster Configuration**.
   vi. Click **Modify Configuration** on the right side of **YML Configuration**.
   vii. In the **YML File Configuration** panel, set **Auto Indexing** to **Enable**.



> **Warning** This operation will restart the cluster. Therefore, before you change the value of **Auto Indexing**, make sure that the restart does not affect your services.

   viii. Select **This operation will restart the cluster. Continue?** and click **OK**.

2. Log on to the Kibana console of the Elasticsearch cluster.

   For more information, see Log on to the Kibana console.

3. In the left-side navigation pane, click **Dev Tools**.
4. On the **Console** tab of the page that appears, run the following command to combine the types of the index:

```
POST _reindex
{
  "source": {
    "index": "twitter"
  },
  "dest": {
    "index": "new1"
  },
  "script": {
    "inline": """
    ctx._id = ctx._type + "-" + ctx._id;
    ctx._source.type = ctx._type;
    ctx._type = "doc";
    """,
    "lang": "painless"
  }
}
```

In this example, a custom type field is added for the new1 index. ctx._source.type specifies the custom type field, and this field is set to the value of the original _type parameter. In addition, _id of the new1 index includes _type-_id. This prevents documents of different types from having the same ID.

5. Run the `GET new1/_mapping` command to view the mapping after the combination.
6. Run the following command to view data in the new index with types combined:

```
GET new1/_search
{
   "query":{
     "match_all":{
      }
  }
}
```

In the following steps, the multi-type index is split into multiple single-type indexes.

1. On the **Console** tab, run the following command to split the multi-type index into multiple single-type indexes:

```
POST _reindex
{
  "source": {
    "index": "twitter",
    "type": "tweet",
    "size": 10000
  },
  "dest": {
    "index": "twitter_tweet"
  }
}
POST _reindex
{
  "source": {
    "index": "twitter",
    "type": "user",
    "size": 10000
  },
  "dest": {
    "index": "twitter_user"
  }
}
```

In this example, the twitter index is split into the twitter_tweet and twitter_user indexes based on types.

2. Run the following command to view data in the new indexes:

```
GET twitter_tweet/_search
{
   "query":{
     "match_all":{
      }
  }
}
```

```
GET twitter_user/_search
{
   "query":{
     "match_all":{
      }
  }
}
```

Step 2: Use Logstash to migrate data

1.
2.
3.
4. In the left-side navigation pane of the page that appears, click **Pipelines**.
5. In the **Pipelines** section, click **Create Pipeline**.

   For more information about how to create and configure a pipeline, see Use configuration files to

manage pipelines. The following example configurations are used in this topic:

```
input {
    elasticsearch {
    hosts => ["http://es-cn-0pp1f1y5g000h****.elasticsearch.aliyuncs.com:9200"]
    user => "elastic"
    index => "*"
    password => "your_password"
    docinfo => true
  }
}
filter {
}
output {
  elasticsearch {
    hosts => ["http://es-cn-mp91cbxsm000c****.elasticsearch.aliyuncs.com:9200"]
    user => "elastic"
    password => "your_password"
    index => "test"
  }
}
```

6. Click Save and Deploy to start data migration.

   For more information, see Use configuration files to manage pipelines.

Step 3: View the data migration results

1. Log on to the Kibana console of the Elasticsearch V6.7.0 cluster.

   For more information, see Log on to the Kibana console.

2. In the left-side navigation pane, click **Dev Tools**.
3. On the **Console** tab of the page that appears, run the following command to view the index that stores the migrated data:

```
GET _cat/indices?v
```

2.2. Migrate data from a user-created Elasticsearch cluster

2.2.1. Use OSS to migrate data from a user-created Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster

Use OSS to migrate data from a user-created Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster

You can use the snapshots that are stored in Object Storage Service (OSS) to migrate data from a user-created Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster. To migrate data, call the snapshot operation to create a snapshot for the user-created Elasticsearch cluster. Then, store the snapshot in OSS and restore data from the snapshot to your Alibaba Cloud Elasticsearch cluster. This topic describes the detailed procedure.

Context

OSS is used to migrate large volumes of data.

Procedure

1. Preparations

   Prepare a user-created Elasticsearch cluster, and create an OSS bucket and an Alibaba Cloud Elasticsearch cluster.

2. Step 1: Install the elasticsearch-repository-oss plug-in

   Install the elasticsearch-repository-oss plug-in on each node of the user-created Elasticsearch cluster. You can create an OSS repository for the user-created Elasticsearch cluster only after the plug-in is installed.

3. Step 2: Create a repository for the user-created Elasticsearch cluster

   Call the snapshot operation to create an OSS repository for the user-created Elasticsearch cluster.

4. Step 3: Create a snapshot for specified indexes

   Create a snapshot for the indexes whose data you want to migrate and store the snapshot in the OSS repository.

5. Step 4: Create the same repository for the Alibaba Cloud Elasticsearch cluster

   In the Kibana console of your Alibaba Cloud Elasticsearch cluster, call the snapshot operation to create a repository that has the same name as the repository for the user-created Elasticsearch cluster.

6. Step 5: Restore data from the created snapshot

   Restore data from the snapshot in the repository of the user-created Elasticsearch cluster to your Alibaba Cloud Elasticsearch cluster.

7. Step 6: View restoration results

   View the restored indexes and their data.

Preparations

1. Prepare a user-created Elasticsearch cluster.

   We recommend that you deploy an Elasticsearch cluster on an Alibaba Cloud Elastic Compute Service (ECS) instance. For more information, see Installing and Running Elasticsearch.

   A single-node Elasticsearch V6.7.0 cluster is used in this topic. In actual production, you can purchase multiple ECS instances that reside in the same virtual private cloud (VPC) to deploy an Elasticsearch cluster. For more information about how to purchase an ECS instance, see Create an instance by using the wizard.

2. Activate OSS, and create a bucket in the region where the ECS instance that hosts the user-created Elasticsearch cluster resides.

   For more information, see 开通OSS服务 and Create buckets.

> **Notice** The storage class of the OSS bucket must be Standard. Elasticsearch does not support the Archive storage class.

3. Create an Alibaba Cloud Elasticsearch cluster in the region where the OSS bucket you created resides.

   For more information, see Create an Alibaba Cloud Elasticsearch cluster.

Step 1: Install the elasticsearch-repository-oss plug-in

1. Connect to the ECS instance that hosts the user-created Elasticsearch cluster.

   > **Note** For more information, see Connect to a Linux instance.

2. Download the installation package of the elasticsearch-repository-oss plug-in.

   In this topic, the version of the plug-in is V6.7.0, which requires JDK 8.0 or later.

   ```
   wget https://github.com/aliyun/elasticsearch-repository-oss/releases/download/v6.7.0/elasticsearch-repository-oss-6.7.0.zip
   ```

3. Decompress the installation package to the plugins folder in the installation path for the user-created Elasticsearch cluster on the ECS instance.

   ```
   unzip -d /usr/local/elasticsearch-6.7.0/plugins/elasticsearch-repository-oss elasticsearch-repository-oss-6.7.0.zip
   ```

   You can also use the command to install plugins.

   ```
   ./bin/elasticsearch-plugin install file:///usr/local/elasticsearch-repository-oss-6.7.0.zip
   ```

4. Start the ECS instance that hosts the user-created Elasticsearch cluster.

   ```
   cd /usr/local/elasticsearch-6.7.0/bin
   ./elasticsearch -d
   ```

Step 2: Create a repository for the user-created Elasticsearch cluster

Connect to the ECS instance that hosts the user-created Elasticsearch cluster and run the following command to create a repository:

```
curl -H "Content-Type: application/json" -XPUT localhost:9200/_snapshot/es_backup -d' {"type": "oss", "settings": { "endpoint": "http://oss-cn-hangzhou-internal.aliyuncs.com",  "access_key_id": "your_accesskeyid",  "secret_access_key":"your_accesskeysecret", "bucket": "es-backup-es", "compress": true }}'
```

Parameter	Description
`es_backup`	The name of the repository, which can be customized.
`type`	The type of the repository. Set the value to `oss`.

Parameter	Description
endpoint	The endpoint of your OSS bucket. For more information, see Regions and endpoints. **Note** If the ECS instance that hosts the user-created Elasticsearch cluster resides in the same region as your OSS bucket, use the internal endpoint of the OSS bucket. Otherwise, use the public endpoint.
access_key_id	The AccessKey ID of the Alibaba Cloud account that is used to create the OSS bucket. For more information about how to obtain the AccessKey ID, see How can I obtain an AccessKey pair?
secret_access_key	The AccessKey secret of the Alibaba Cloud account that is used to create the OSS bucket. For more information about how to obtain the AccessKey secret, see How can I obtain an AccessKey pair?
bucket	The name of your OSS bucket.
compress	Specifies whether to enable compression.

If the repository is created, `"acknowledge":true` is returned.

Step 3: Create a snapshot for specified indexes

Create a snapshot for indexes whose data you want to migrate. By default, all enabled indexes are backed up in the snapshot. If you do not want to back up system indexes, such as indexes whose names start with `.kibana`, `.security`, or `.monitoring`, you can specify the indexes you want to back up.

Notice We recommend that you do not back up system indexes because they occupy large storage space.

```
curl -H "Content-Type: application/json" -XPUT localhost:9200/_snapshot/es_backup/snapshot_
1?pretty -d'
{
"indices": "index1,index2"
}'
```

`index1` and `index2` indicate the names of the indexes that you want to back up. If the snapshot is created, `"accepted" : true` is returned.

Step 4: Create the same repository for the Alibaba Cloud Elasticsearch cluster

1. Log on to the Kibana console of the Alibaba Cloud Elasticsearch cluster.

   For more information, see Log on to the Kibana console.
2. In the left-side navigation pane, click **Dev Tools**.

3. On the **Console** tab of the page that appears, run the following command to create a repository that has the same name as the repository for the user-created Elasticsearch cluster.

```
PUT _snapshot/es_backup
{
    "type": "oss",
    "settings": {
        "endpoint": "oss-cn-hangzhou-internal.aliyuncs.com",
        "access_key_id": "your_accesskeyid",
        "secret_access_key": "your_accesskeysecret",
        "bucket": "es-backup-es",
        "compress": true
    }
}
```

Step 5: Restore data from the created snapshot

In the Kibana console of the Alibaba Cloud Elasticsearch cluster, run the following command to restore all indexes (except system indexes whose names start with `.`) from the created snapshot. Follow the instructions in the "Step 4: Create the same repository for the Alibaba Cloud Elasticsearch cluster" section to perform the restoration.

```
POST _snapshot/es_backup/snapshot_1/_restore
{"indices":"*,-.monitoring*,-.security_audit*","ignore_unavailable":"true"}
```

If the command is successfully executed, `"accepted" : true` is returned.

The preceding command restores all indexes in the snapshot. You can also specify the indexes that you want to restore. In the Alibaba Cloud Elasticsearch cluster, an existing index may have the same name as an index you want to restore. In this case, if you do not want to replace the data in the existing index, you can rename the index you want to restore during the restoration.

```
POST _snapshot/es_backup/snapshot_1/_restore
{
  "indices":"index1",
  "rename_pattern": "index(.+)",
  "rename_replacement": "restored_index_$1"
}
```

> **Note** For more information about the commands that are used to create snapshots or restore data, see Create manual snapshots and restore data from manual snapshots.

Step 6: View restoration results

In the Kibana console of the Alibaba Cloud Elasticsearch cluster, run the following command to view the restoration results. Follow the instructions in the "Step 4: Create the same repository for the Alibaba Cloud Elasticsearch cluster" section to perform the operation.

- View the restored indexes

```
GET /_cat/indices?v
```

```
health status index
green  open   .monitoring-kibana-6-2020.05.28
green  open   .monitoring-es-6-2020.05.28
green  open   index1
green  open   .monitoring-logstash-6-2020.06.03
green  open   .monitoring-kibana-6-2020.06.04
green  open   .kibana_task_manager
```

- View the data in the restored indexes

```
GET /index1/_search
```

If the command is successfully executed, the following result is returned:

```
{
  "took" : 2,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 1,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "index1",
        "_type" : "_doc",
        "_id" : "1",
        "_score" : 1.0,
        "_source" : {
          "productName" : "testpro",
          "annual_rate" : "3.22%",
          "describe" : "testpro"
        }
      }
    ]
  }
}
```

2.2.2. Use Alibaba Cloud Logstash to migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster

Migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster

You can use an Alibaba Cloud Logstash pipeline to migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster. This topic describes the migration procedure in detail.

Prerequisites

- A self-managed Elasticsearch cluster is created.

  We recommend that you create a self-managed Elasticsearch cluster on Alibaba Cloud Elastic Compute Service (ECS) instances. For more information, see Install and Run Elasticsearch.

  > **Notice**
  > - The ECS instances that host the self-managed Elasticsearch cluster must be deployed in a virtual private cloud (VPC). You cannot use ECS instances that are connected to a VPC over ClassicLink.
  > - Alibaba Cloud Logstash clusters are deployed in VPCs. Before you configure a Logstash pipeline, you must check whether the ECS instances that host the self-managed Elasticsearch cluster reside in the same VPC as the Alibaba Cloud Logstash cluster that you want to use. If they reside in different VPCs, you must configure NAT gateways to connect the ECS instances and Logstash cluster to the Internet. For more information, see Configure a NAT gateway for data transmission over the Internet.
  > - You must configure security group rules to allow access from the IP addresses of the nodes in the Logstash cluster for the security groups of the ECS instances that host the self-managed Elasticsearch cluster. In addition, you must enable port 9200. You can obtain the IP addresses of the nodes in the Logstash cluster on the Basic Information page of the Logstash cluster.
  > - In this example, an Alibaba Cloud Logstash V6.7.0 cluster is used to migrate data from a self-managed Elasticsearch 5.6.16 cluster to an Alibaba Cloud Elasticsearch V6.7.0 cluster. The scripts provided in this topic apply only to this type of data migration. If you want to perform other types of data synchronization, you must check whether your Elasticsearch clusters and Logstash cluster meet compatibility requirements based on the instructions in Compatibility matrixes. If they do not meet compatibility requirements, you can upgrade their versions or purchase new clusters.

- An Alibaba Cloud Logstash cluster is created.

  For more information, see Create a cluster.

- An Alibaba Cloud Elasticsearch cluster is created in the VPC where the Alibaba Cloud Logstash cluster resides. Make sure that the Alibaba Cloud Elasticsearch cluster is of the same version as the Logstash cluster. In this example, V6.7.0 is used.

  For more information, see Create an Alibaba Cloud Elasticsearch cluster.

- The Auto Indexing feature is enabled for the Alibaba Cloud Elasticsearch cluster.

  For more information, see Configure the YML file.

  > **Note** Logstash does not synchronize the structure features of data when Logstash migrates data. Therefore, if you enable the Auto Indexing feature, the structure of data may change after the data is migrated to the destination. If you want the structure of the data to remain unchanged, we recommend that you create an empty index in the destination and migrate data to the index. When you create the index, copy the mappings and settings configurations of the source and set the numbers of shards and replicas to appropriate values.

Configure and run a Logstash pipeline

1.

2.
3.
4.
5. In the **Create Task** wizard, enter a pipeline ID and configure the pipeline.

   In this example, the following configurations are used for the pipeline:

```
input {
  elasticsearch {
    hosts => ["http://<IP address of the master node in the self-managed Elasticsearch cluster>:9200"]
    user => "elastic"
    index => "*,-.monitoring*,-.security*,-.kibana*"
    password => "your_password"
    docinfo => true
  }
}
filter {
}
output {
  elasticsearch {
    hosts => ["http://es-cn-mp91cbxsm000c****.elasticsearch.aliyuncs.com:9200"]
    user => "elastic"
    password => "your_password"
    index => "%{[@metadata][_index]}"
    document_type => "%{[@metadata][_type]}"
    document_id => "%{[@metadata][_id]}"
  }
  file_extend {
        path => "/ssd/1/ls-cn-v0h1kzca****/logstash/logs/debug/test"
    }
}
```

   Parameters

Parameter	Description
hosts	The endpoint of the self-managed Elasticsearch cluster or Alibaba Cloud Elasticsearch cluster. In the input part, specify a value for this parameter in the format of `http://<IP address of the master node in the self-managed Elasticsearch cluster>:<Port number>`. In the output part, specify a value for this parameter in the format of `http://<ID of the Alibaba Cloud Elasticsearch cluster>.elasticsearch.aliyuncs.com:9200`. **Notice** When you configure this parameter, you must replace `<IP address of the master node in the self-managed Elasticsearch cluster>`, `<Port number>`, and `<ID of the Alibaba Cloud Elasticsearch cluster>` with your actual values.

Parameter	Description
user	The username that is used to access the self-managed Elasticsearch cluster or Alibaba Cloud Elasticsearch cluster. **Notice** - The user and password parameters are required in most cases. If the X-Pack plug-in is not installed on the self-managed Elasticsearch cluster, you can leave the two parameters empty. - The default username that is used to access the Alibaba Cloud Elasticsearch clusters is elastic. The default username is used in this example. You can use a custom username. Before you use a custom username, you must create a role for it and grant the required permissions to the role. For more information, see Use the RBAC mechanism provided by Elasticsearch X-Pack to implement access control.
password	The password that is used to access the self-managed Elasticsearch cluster or Alibaba Cloud Elasticsearch cluster.
index	The names of the indexes whose data you want to migrate or to which you want to migrate data. If you set this parameter to *,-.monitoring*,-.security*,-.kibana* in the input part, the system migrates data in indexes other than system indexes whose names start with a period (.). If you set this parameter to %{[@metadata][_index]} in the output part, the system matches the index parameter in the metadata. This indicates that the names of the indexes generated on the Alibaba Cloud Elasticsearch cluster are the same as the names of the indexes on the self-managed Elasticsearch cluster.
docinfo	If you set this parameter to true, the system extracts the metadata of documents in the self-managed Elasticsearch cluster, such as the index, type, and id fields.
document_type	If you set this parameter to %{[@metadata][_type]}, the system matches the index type in the metadata. This indicates that the type of the indexes generated on the Alibaba Cloud Elasticsearch cluster is the same as the type of the indexes on the self-managed Elasticsearch cluster.
document_id	If you set this parameter to %{[@metadata][_id]}, the system matches the document IDs in the metadata. This indicates that the IDs of the documents generated on the Alibaba Cloud Elasticsearch cluster are the same as the IDs of the documents on the self-managed Elasticsearch cluster.

Parameter	Description
file_extend	Notice

For more information about how to configure parameters in the Config Settings field, see Logstash configuration files.

6.

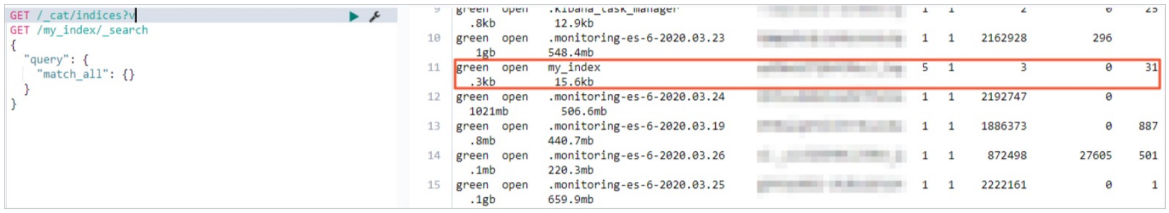
7.

View migration results

1. Log on to the Kibana console of the Alibaba Cloud Elasticsearch cluster.

   For more information, see Log on to the Kibana console.

2. In the left-side navigation pane, click **Dev Tools**.
3. On the **Console** tab of the page that appears, run the `GET /_cat/indices?v` command to view the indexes that store the migrated data.



FAQ

- Q: How do I connect the ECS instances that host the self-managed Elasticsearch cluster to the Alibaba Cloud Logstash cluster when the ECS instances and the Logstash cluster belong to different accounts?

  A: The ECS instances and the Logstash cluster belong to different accounts. Therefore, the ECS instances and the Logstash cluster reside in different VPCs. In this case, you can use Cloud Enterprise Network (CEN) to connect the ECS instances to the Logstash cluster. For more information, see Step 3: Attach network instances.

- Q: An error occurs when Logstash writes data to the destination. How do I do?

  A: Troubleshoot the error based on the instructions provided in FAQ about data transfer by using Logstash.

2.2.3. Use Logstash to migrate full or incremental data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster

If you want to migrate full or incremental data from your self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster, you can deploy a self-managed Logstash cluster on the Elastic Compute Service (ECS) instance that hosts the self-managed Elasticsearch cluster and use the pipeline configuration feature of Logstash to migrate the data. The topic describes the procedure in detail.

Context

The following procedure shows migrating data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster by using a self-managed Logstash cluster.

1. Deploy a self-managed Elasticsearch cluster on an ECS instance and prepare the data that you want to migrate in the self-managed Elasticsearch cluster.
2. Create an Alibaba Cloud Elasticsearch cluster.
3. Run a Python script on the ECS instance to migrate the metadata of indexes in the self-managed Elasticsearch cluster.
4. Deploy a self-managed Logstash cluster on the ECS instance, and configure a pipeline in the Logstash cluster to migrate full or incremental data from the self-managed Elasticsearch cluster to the Alibaba Cloud Elasticsearch cluster.

Precautions

- In this example, the self-managed Logstash cluster is deployed on an Alibaba Cloud ECS instance. The ECS instance must reside in the same virtual private cloud (VPC) as the Alibaba Cloud Elasticsearch cluster. In addition, you must make sure that the Logstash cluster can connect to both the self-managed Elasticsearch cluster and the Alibaba Cloud Elasticsearch cluster.
- Both full migration and incremental migration are supported. If this is the first time that you migrate data from your self-managed Elasticsearch cluster, full migration is performed. For new data that is written to the cluster, you can perform incremental migration. Incremental migration requires that the indexes in the source Elasticsearch cluster have a timestamp field.

Procedure

1. Step 1: Make preparations
2. Step 2: Migrate the metadata of indexes in the self-managed Elasticsearch cluster
3. Step 3: Migrate full data
4. Step 4: Migrate incremental data
5. Step 5: View the data migration results

Step 1: Make preparations

1. Create an Alibaba Cloud Elasticsearch cluster.

   For more information, see Create an Alibaba Cloud Elasticsearch cluster. The following table describes the configurations of the Alibaba Cloud Elasticsearch cluster used in this example.

Configuration item	Description
Region	China (Hangzhou).
Edition and version	V7.10.0 of Standard Edition.

Configuration item	Description
Specifications	Three zones and three data nodes. Specifications of a single data node: 4 vCPUs, 16 GiB of memory, and an enhanced SSD (ESSD) with 100 GiB of storage space.

2. Create an ECS instance, which is used to deploy a self-managed Elasticsearch cluster, the self-managed Kibana service, and a self-managed Logstash cluster.

   For more information about how to create an ECS instance, see Create an instance by using the wizard. The following table describes the configurations of the ECS instance used in this example.

Configuration item	Description
Region	China (Hangzhou).
Specifications	4 vCPUs and 16 GiB of memory.
Image	Public image: CentOS 7.9 64-bit.
Storage	System disk: an ESSD with 100 GiB of storage space.
Network	The ECS instance resides in the same VPC as the Alibaba Cloud Elasticsearch cluster, and **Assign Public IPv4 Address** is selected for the ECS instance. The network usage of the ECS instance is charged based on the pay-by-traffic billing method. The peak bandwidth is 100 Mbit/s.
Security group	An inbound rule that allows traffic on port 5601 is added to a security group of the ECS instance. Port 5601 is the port of the Kibana service. The IP address of your client is added as an authorization object. **Notice** ◦ If your client is in a home network or in a LAN of an office, you must add the IP address of the Internet egress rather than the IP address of the client to a security group of the ECS instance. We recommend that you query the IP address of the Internet egress in the IP address library of Taobao. ◦ You can also add 0.0.0.0/0 as an authorization object. If you make this configuration, all public IPv4 addresses can be used to access the ECS instance. This poses security risks. We recommend that you do not make this configuration in the production environment.

3. Deploy a self-managed Elasticsearch cluster on the ECS instance.

   In this example, a self-managed Elasticsearch 7.6.2 cluster that has one data node is used. To deploy the cluster on the ECS instance, perform the following steps:

i. Connect to the ECS instance.

For more information, see Connect to a Linux instance by using a password or key.

ii. Create a user account named elastic.

```
useradd elastic
passwd <your_password>
```

iii. Switch from the root user to the elastic user.

```
su -l elastic
```

iv. Download the Elasticsearch package and decompress the package.

```
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.6.2-linux
-x86_64.tar.gz
tar -zvxf elasticsearch-7.6.2-linux-x86_64.tar.gz
```

v. Use the elastic user to start the Elasticsearch cluster.

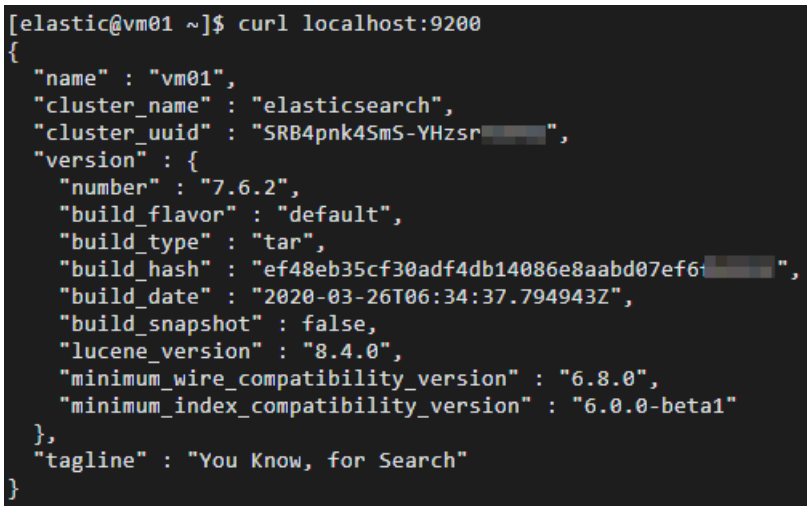
Go to the directory in which the Elasticsearch cluster is deployed and start the Elasticsearch cluster.

```
cd elasticsearch-7.6.2
./bin/elasticsearch -d
```

vi. Check whether the Elasticsearch cluster runs as expected.

```
cd ~
curl localhost:9200
```

If the Elasticsearch cluster runs as expected, the result shown in the following figure is returned. The result contains the version number of the Elasticsearch cluster and the message "You Know, for Search".



4. Deploy the self-managed Kibana service and prepare test data.

In this example, the self-managed Kibana 7.6.2 service that has one data node is used. To deploy the self-managed Kibana service, perform the following steps:

i. Connect to the ECS instance.

For more information, see Connect to a Linux instance by using a password or key.

ii. Download the Kibana package and decompress the package.

```
wget https://artifacts.elastic.co/downloads/kibana/kibana-7.6.2-linux-x86_64.tar.gz
tar -zvxf kibana-7.6.2-linux-x86_64.tar.gz
```

iii. Modify the kibana.yml configuration file in the *config/* installation directory of the Kibana service. Add the `server.host: "0.0.0.0"` configuration to the configuration file to allow access to the Kibana service from all public IP addresses.

Go to the installation directory of the Kibana service and modify the *kibana.yml* configuration file.

```
cd kibana-7.6.2-linux-x86_64
vi config/kibana.yml
```

```
# Kibana is served by a back end server. This setting specifies the port to use.
#server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.
# The default is 'localhost', which usually means remote machines will not be able to connect.
# To allow connections from remote users, set this parameter to a non-loopback address.
#server.host: "localhost"
server.host: "0.0.0.0"
# Enables you to specify a path to mount Kibana at if you are running behind a proxy.
# Use the `server.rewriteBasePath` setting to tell Kibana if it should remove the basePath
# from requests it receives, and to prevent a deprecation warning at startup.
# This setting cannot end in a slash.
#server.basePath: ""
```

iv. Use the elastic user to start the Kibana service.

```
nohup ./bin/kibana &
```

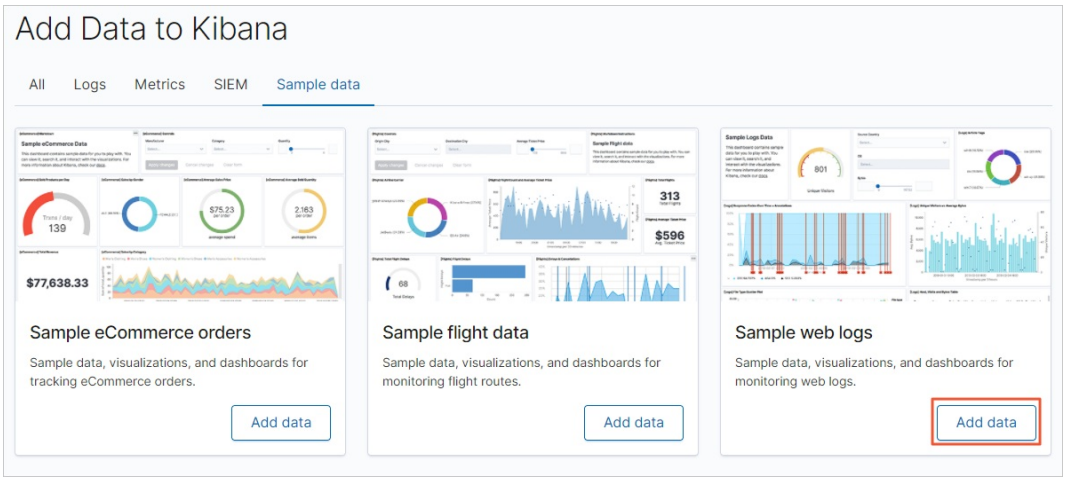
v. Log on to the Kibana console and add the test data.

a. Use the URL that contains the public IP address of the ECS instance to log on to the Kibana console.

The URL is in the following format: **http://<Public IP address of the ECS instance>:5601/app/kibana#/home**.

b. On the homepage of the Kibana console, click **Try our sample data**.

c. On the **Sample data** tab, click **Add data** in the Sample web logs card and add the test data.



5. Deploy a self-managed Logstash cluster on the ECS instance.

In this example, a self-managed Logstash 7.10.0 cluster that has one node is used. To deploy the

self-managed Logstash cluster, perform the following steps:

i. Connect to the ECS instance.

For more information, see Connect to a Linux instance by using a password or key.

ii. Go back to the root directory, download the Logstash package, and then decompress the package.

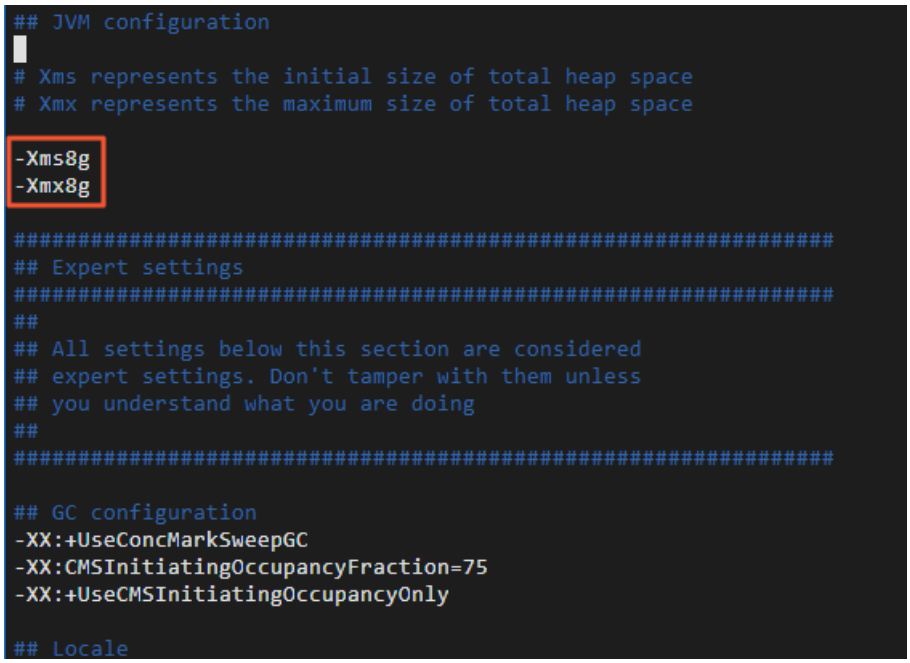
```
cd ~
wget https://artifacts.elastic.co/downloads/logstash/logstash-7.10.0-linux-x86_64.tar.gz
tar -zvxf logstash-7.10.0-linux-x86_64.tar.gz
```

iii. Change the JVM heap memory setting of the Logstash cluster.

The default JVM heap memory size of the Logstash cluster is 1 GiB. You must specify an appropriate heap memory size for the Logstash cluster based on the specifications of the ECS instance to accelerate data migration.

Go to the installation directory of the Logstash cluster and modify the jvm.options configuration file in the config/ directory. Add the -Xms8g and -Xmx8g settings to the configuration file.

```
cd logstash-7.10.0
vi config/jvm.options
```



iv. Change the number of data records that can be written by the Logstash cluster at a time.

Enable the Logstash cluster to write 5 to 15 MiB of data records at a time. This can accelerate data migration.

Modify the pipelines.yml pipeline configuration file in the config/ directory. Change the value of the pipeline.batch.size parameter in the configuration file from 125 to 5000. The pipeline.batch.size parameter specifies the number of data records that can be written by the Logstash cluster at a time.

```
vi config/pipelines.yml
```

```
#   # The path from where to read the configuration text
#   path.config: "/etc/conf.d/logstash/myconfig.cfg"
#
#   # How many worker threads execute the Filters+Outputs stage of the pipeline
#   pipeline.workers: 1 (actually defaults to number of CPUs)
#
   # How many events to retrieve from inputs before sending to filters+workers
   pipeline.batch.size: 5000
#
#   # How long to wait in milliseconds while polling for the next event
#   # before dispatching an undersized batch to filters+outputs
#   pipeline.batch.delay: 50
#
#   # Internal queuing model, "memory" for legacy in-memory based queuing and
#   # "persisted" for disk-based acked queueing. Defaults is memory
#   queue.type: memory
#
```

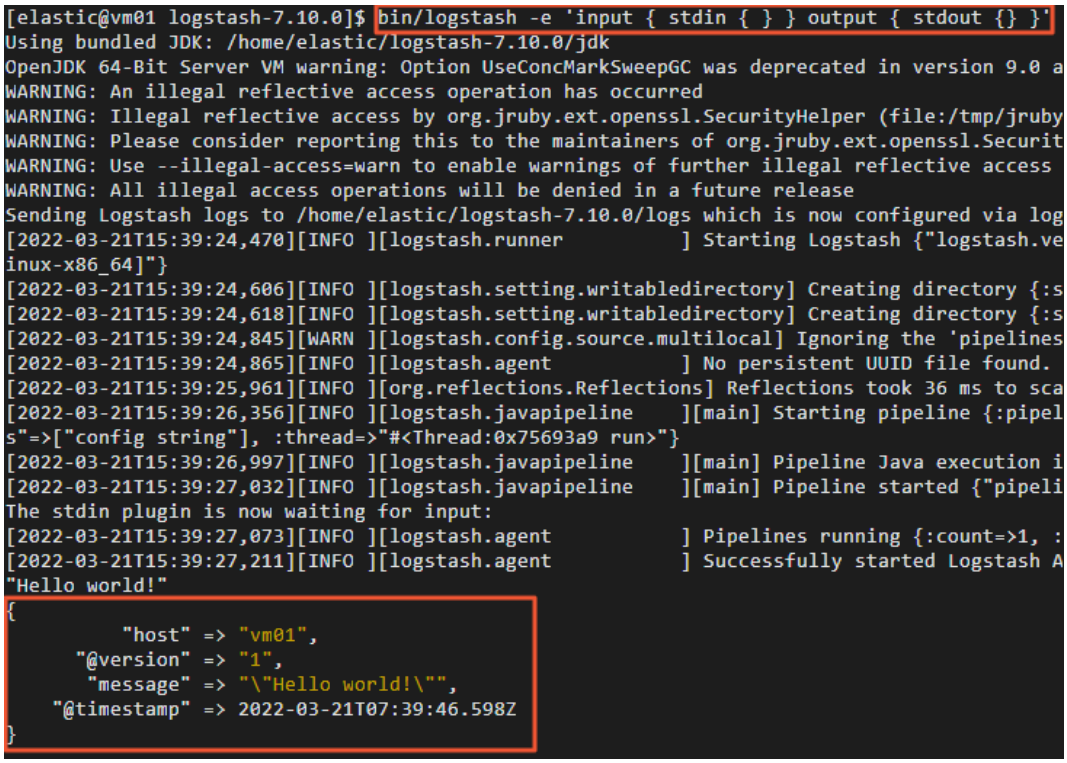
v. Check whether the Logstash cluster runs as expected.

a. Run the following command in the command-line tool of the host of the ECS instance to collect the input and output data:

```
bin/logstash -e 'input { stdin { } } output { stdout {} }'
```

b. Enter *"Hello world!"* in the command-line tool as the input.

If the Logstash cluster runs as expected, "Hello world!" is returned as the output.



Step 2: Migrate the metadata of indexes in the self-managed Elasticsearch cluster

If you enable the Auto Indexing feature for the Alibaba Cloud Elasticsearch cluster, when you migrate data, the system automatically creates indexes in the cluster. However, these indexes may be different from the indexes that you want to migrate from the self-managed Elasticsearch cluster. As a result, the formats of data in the Alibaba Cloud Elasticsearch cluster may be different from those of data in the self-managed Elasticsearch cluster. We recommend that you manually create indexes in the Alibaba Cloud Elasticsearch cluster before data migration. This ensures that data in the Alibaba Cloud Elasticsearch cluster is the same as the data in the self-managed Elasticsearch cluster.

You can use Python scripts to create indexes in the Alibaba Cloud Elasticsearch cluster. To create an index, perform the following steps:

1. Connect to the ECS instance.

   For more information, see Connect to a Linux instance by using a password or key.

2. Create and open a Python script file. In this example, a Python script file named indiceCreate.py is created.

   ```
   vi indiceCreate.py
   ```

3. Modify the Python script file. Copy the following code to the Python script file. You must change the hosts, usernames, and passwords of the source and destination Elasticsearch clusters in the following code based on your business requirements.

   ```
   #!/usr/bin/python
   # -*- coding: UTF-8 -*-
   # File name: indiceCreate.py
   ```

```
import sys
import base64
import time
import httplib
import json
## Specify the host of the source Elasticsearch cluster.
oldClusterHost = "localhost:9200"
## Specify the username of the source Elasticsearch cluster. This parameter can be left
empty.
oldClusterUserName = "elastic"
## Specify the password of the source Elasticsearch cluster. This parameter can be left
empty.
oldClusterPassword = "xxxxxx"
## Specify the host of the destination Elasticsearch cluster. You can obtain the host o
n the Basic Information page of the destination Elasticsearch cluster.
newClusterHost = "es-cn-zvp2m4bko0009****.elasticsearch.aliyuncs.com:9200"
## Specify the username of the destination Elasticsearch cluster.
newClusterUser = "elastic"
## Specify the password of the destination Elasticsearch cluster.
newClusterPassword = "xxxxxx"
DEFAULT_REPLICAS = 0
def httpRequest(method, host, endpoint, params="", username="", password=""):
    conn = httplib.HTTPConnection(host)
    headers = {}
    if (username != "") :
        'Hello {name}, your age is {age} !'.format(name = 'Tom', age = '20')
        base64string = base64.encodestring('{username}:{password}'.format(username = us
ername, password = password)).replace('\n', '')
        headers["Authorization"] = "Basic %s" % base64string;
    if "GET" == method:
        headers["Content-Type"] = "application/x-www-form-urlencoded"
        conn.request(method=method, url=endpoint, headers=headers)
    else :
        headers["Content-Type"] = "application/json"
        conn.request(method=method, url=endpoint, body=params, headers=headers)
    response = conn.getresponse()
    res = response.read()
    return res
def httpGet(host, endpoint, username="", password=""):
    return httpRequest("GET", host, endpoint, "", username, password)
def httpPost(host, endpoint, params, username="", password=""):
    return httpRequest("POST", host, endpoint, params, username, password)
def httpPut(host, endpoint, params, username="", password=""):
    return httpRequest("PUT", host, endpoint, params, username, password)
def getIndices(host, username="", password=""):
    endpoint = "/_cat/indices"
    indicesResult = httpGet(oldClusterHost, endpoint, oldClusterUserName, oldClusterPas
sword)
    indicesList = indicesResult.split("\n")
    indexList = []
    for indices in indicesList:
        if (indices.find("open") > 0):
            indexList.append(indices.split()[2])
    return indexList
```

```
def getSettings(index, host, username="", password=""):
    endpoint = "/" + index + "/_settings"
    indexSettings = httpGet(host, endpoint, username, password)
    print (index + "  Original settings: \n" + indexSettings)
    settingsDict = json.loads(indexSettings)
    ## By default, the number of primary shards is the same as that for the indexes in
the source Elasticsearch cluster.
    number_of_shards = settingsDict[index]["settings"]["index"]["number_of_shards"]
    ## The default number of replica shards is 0.
    number_of_replicas = DEFAULT_REPLICAS
    newSetting = "\"settings\": {\"number_of_shards\": %s, \"number_of_replicas\": %s}"
% (number_of_shards, number_of_replicas)
    return newSetting
def getMapping(index, host, username="", password=""):
    endpoint = "/" + index + "/_mapping"
    indexMapping = httpGet(host, endpoint, username, password)
    print (index + " Original mappings: \n" + indexMapping)
    mappingDict = json.loads(indexMapping)
    mappings = json.dumps(mappingDict[index]["mappings"])
    newMapping = "\"mappings\" : " + mappings
    return newMapping
def createIndexStatement(oldIndexName):
    settingStr = getSettings(oldIndexName, oldClusterHost, oldClusterUserName, oldClust
erPassword)
    mappingStr = getMapping(oldIndexName, oldClusterHost, oldClusterUserName, oldCluste
rPassword)
    createstatement = "{\n" + str(settingStr) + ",\n" + str(mappingStr) + "\n}"
    return createstatement
def createIndex(oldIndexName, newIndexName=""):
    if (newIndexName == "") :
        newIndexName = oldIndexName
    createstatement = createIndexStatement(oldIndexName)
    print ("New index " + newIndexName + " Index settings and mappings: \n" + createsta
tement)
    endpoint = "/" + newIndexName
    createResult = httpPut(newClusterHost, endpoint, createstatement, newClusterUser, n
ewClusterPassword)
    print ("New index " + newIndexName + " Creation result: " + createResult)
## main
indexList = getIndices(oldClusterHost, oldClusterUserName, oldClusterPassword)
systemIndex = []
for index in indexList:
    if (index.startswith(".")):
        systemIndex.append(index)
    else :
        createIndex(index, index)
if (len(systemIndex) > 0) :
    for index in systemIndex:
        print (index + " It may be a system index and will not be recreated. You can ma
nually recreate the index based on your business requirements.")
```

4. Run the Python script to create an index in the Alibaba Cloud Elasticsearch cluster.

```
/usr/bin/python indiceCreate.py
```

5. Log on to the Kibana console of the Alibaba Cloud Elasticsearch cluster and view the created index. Fore more information about how to log on to the Kibana console, see Log on to the Kibana console.

```
GET /_cat/indices?v
```

Step 3: Migrate full data

1. Connect to the ECS instance.

   For more information, see Connect to a Linux instance by using a password or key.

2. Create a pipeline configuration file for the Logstash cluster and open the configuration file in the config directory.

```
cd logstash-7.10.0/config
vi es2es_all.conf
```

3. Modify the configuration file by referring to the following code:

```
input{
    elasticsearch{
        # Specify the host of the source Elasticsearch cluster.
        hosts =>  ["http://localhost:9200"]
        # Specify the username and password of the source Elasticsearch cluster.
        user => "xxxxxx"
        password => "xxxxxx"
        # Specify the names of the indexes that you want to migrate. Separate the names
with commas (,).
        index => "kibana_sample_data_*"
        # Retain the following default settings. The settings are related to the number
of threads used for data migration, the size of data that needs to be migrated, and the
JVM heap memory size of the Logstash cluster.
        docinfo=>true
        slices => 5
        size => 5000
    }
}
filter {
  # Remove some fields that are added by Logstash.
  mutate {
    remove_field => ["@timestamp", "@version"]
  }
}
output{
    elasticsearch{
        # Specify the host of the destination Elasticsearch cluster. You can obtain the
host on the Basic Information page of the destination Elasticsearch cluster.
        hosts => ["http://es-cn-zvp2m4bko0009****.elasticsearch.aliyuncs.com:9200"]
        # Specify the username and password of the destination Elasticsearch cluster.
        user => "elastic"
        password => "xxxxxx"
        # Specify the names of the indexes in the destination Elasticsearch cluster. Th
e following setting indicates that the names of the indexes in the destination Elastics
earch cluster are the same as those of the indexes in the source Elasticsearch cluster.

        index => "%{[@metadata][_index]}"
        # Specify the types of the indexes in the destination Elasticsearch cluster. Th
e following setting indicates that the types of the indexes in the destination Elastics
earch cluster are the same as those of the indexes in the source Elasticsearch cluster.

        document_type => "%{[@metadata][_type]}"
        # Specify the IDs of the documents in the destination Elasticsearch cluster. If
you do not want to retain the IDs of the source documents in the destination Elasticsea
rch cluster, you can delete the setting in the following row. This provides better perf
ormance.
        document_id => "%{[@metadata][_id]}"
        ilm_enabled => false
        manage_template => false
    }
}
```

> **Note** To ensure the accuracy of data migration, we recommend that you create multiple Logstash pipelines and use each pipeline to migrate part of the data.

4. Start the Logstash cluster to migrate full data.

```
cd ../
nohup bin/logstash -f config/es2es_all.conf >/dev/null 2>&1 &
```

Step 4: Migrate incremental data

1. Connect to the ECS instance, create a pipeline configuration file for the Logstash cluster, and then open the configuration file in the config directory.

```
cd config
vi es2es_kibana_sample_data_logs.conf
```

2. Modify the configuration file by referring to the following code:

   Add the schedule parameter to the input part in the pipeline configuration file to enable the Logstash cluster to migrate incremental data. Sample code:

```
input{
    elasticsearch{
        # Specify the host of the source Elasticsearch cluster.
        hosts =>  ["http://localhost:9200"]
        # Specify the username and password of the source Elasticsearch cluster.
        user => "xxxxxx"
        password => "xxxxxx"
        # Specify the names of the indexes that you want to migrate. Separate the names
with commas (,).
        index => "kibana_sample_data_logs"
        # Specify the time range in which you want to query incremental data. The follo
wing setting indicates that incremental data over the last 5 minutes is queried.
        query => '{"query":{"range":{"@timestamp":{"gte":"now-5m","lte":"now/m"}}}}'
        # Specify the interval at which incremental data is collected. The following se
tting indicates that incremental data is collected every minute.
        schedule => "* * * * *"
        scroll => "5m"
        docinfo=>true
        size => 5000
    }
}
filter {
  # Remove some fields that are added by Logstash.
  mutate {
    remove_field => ["@timestamp", "@version"]
  }
}
output{
    elasticsearch{
        # Specify the host of the destination Elasticsearch cluster. You can obtain the
host on the Basic Information page of the destination Elasticsearch cluster.
        hosts => ["http://es-cn-zvp2m4bko0009****.elasticsearch.aliyuncs.com:9200"]
        # Specify the username and password of the destination Elasticsearch cluster.
        user => "elastic"
```

```
        user => "elastic"
        password => "xxxxxx"
        # Specify the names of the indexes in the destination Elasticsearch cluster. Th
e following setting indicates that the names of the indexes in the destination Elastics
earch cluster are the same as those of the indexes in the source Elasticsearch cluster.

        index => "%{[@metadata][_index]}"
        # Specify the types of the indexes in the destination Elasticsearch cluster. Th
e following setting indicates that the types of the indexes in the destination Elastics
earch cluster are the same as those of the indexes in the source Elasticsearch cluster.

        document_type => "%{[@metadata][_type]}"
        # Specify the IDs of the documents in the destination Elasticsearch cluster. If
you do not want to retain the IDs of the source documents in the destination Elasticsea
rch cluster, you can delete the setting in the following row. This provides better perf
ormance.
        document_id => "%{[@metadata][_id]}"
        ilm_enabled => false
        manage_template => false
    }
}
```

3. Start the Logstash cluster to migrate incremental data.

```
nohup bin/logstash -f config/es2es_kibana_sample_data_logs.conf >/dev/null 2>&1 &
```

4. In the Kibana console of the Alibaba Cloud Elasticsearch cluster, query data records that are most recently updated to check whether incremental data is migrated.

   In this example, the data records that are updated over the last 5 minutes in the kibana_sample_data_logs index are queried.

```
GET kibana_sample_data_logs/_search
{
  "query": {
    "range": {
      "@timestamp": {
        "gte": "now-5m",
        "lte": "now/m"
      }
    }
  },
  "sort": [
    {
      "@timestamp": {
        "order": "desc"
      }
    }
  ]
}
```

Step 5: View the data migration results

1. Check whether the full data is migrated.

i. Query information about indexes, the number of data records, and the volume of data stored in the self-managed Elasticsearch cluster.

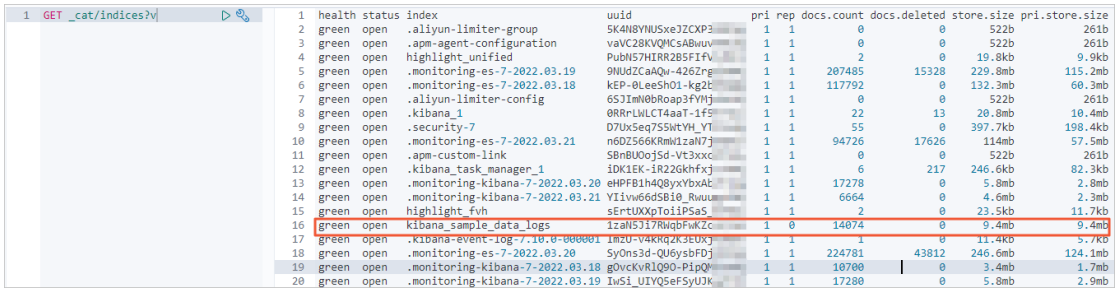
```
GET _cat/indices?v
```

The following result is returned.

health	status	index	uuid	pri	rep	docs.count	docs.deleted	store.size	pri.store.size
green	open	.kibana_task_manager_1	CxAx5J2sT0qHPsWV	1	0	2	0	6.6kb	6.6kb
green	open	.apm-agent-configuration	dYz5bh4dTomjtDP3	1	0	0	0	283b	283b
green	open	kibana_sample_data_logs	PUBQrSkJRMGyI-cV	1	0	14074	0	11.6mb	11.6mb
green	open	.kibana_1	MXhG2XbYTYSORB8G	1	0	49	4	139.5kb	139.5kb

ii. Query information about indexes, the number of data records, and the volume of data stored in the Alibaba Cloud Elasticsearch cluster after the data migration.

If the full data is migrated, the number of returned data records is the same as the number of data records in the self-managed Elasticsearch cluster.



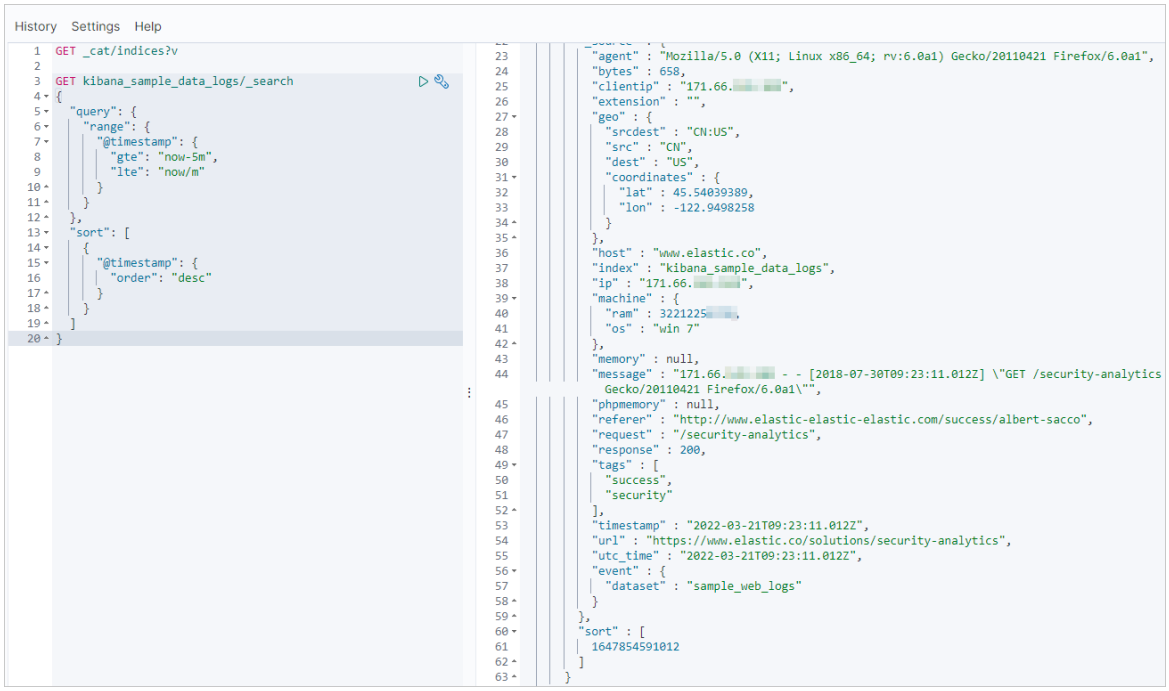
health	status	index	uuid	pri	rep	docs.count	docs.deleted	store.size	pri.store.size
green	open	.aliyun-limiter-group	5K4N8YNUSxeJZCXP3	1	1	0	0	522b	261b
green	open	.apm-agent-configuration	vaVC28KVQMCsABwuv	1	1	0	0	522b	261b
green	open	highlight_unified	PubN57HIRR2B5FIfV	1	1	2	0	19.8kb	9.9kb
green	open	.monitoring-es-7-2022.03.19	9NUdZCaAQw-426Zrg	1	1	207485	15328	229.8mb	115.2mb
green	open	.monitoring-es-7-2022.03.18	kEP-0LeeShO1-kg2b	1	1	117792	0	132.3mb	60.3mb
green	open	.aliyun-limiter-config	6SJImN0bRoap3fYMj	1	1	0	0	522b	261b
green	open	.kibana_1	0RRrLWLCT4aaT-1f5	1	1	22	13	20.8mb	10.4mb
green	open	.security-7	D7Ux5eq7S5WtYH_YT	1	1	55	0	397.7kb	198.4kb
green	open	.monitoring-es-7-2022.03.21	n6DZ566KRmW1zaN7j	1	1	94726	17626	114mb	57.5mb
green	open	.apm-custom-link	SBnBUOojSd-Vt3xxc	1	1	0	0	522b	261b
green	open	.kibana_task_manager_1	iDK1EK-iR22GkhfXj	1	1	6	217	246.6kb	82.3kb
green	open	.monitoring-kibana-7-2022.03.20	eHPFB1h4Q8yxYbxAL	1	1	17278	0	5.8mb	2.8mb
green	open	.monitoring-kibana-7-2022.03.21	YIivw66dSBi0_Rwuu	1	1	6664	0	4.6mb	2.3mb
green	open	highlight_fvh	sErtUXXpToiiPSaS	1	1	2	0	23.5kb	11.7kb
green	open	kibana_sample_data_logs	1zaN5Ji7RWqbFwKZc	1	0	14074	0	9.4mb	9.4mb
green	open	.kibana-event-log-7.10.0-000001	1mzU-V4kRqZK3EUXj	1	1	1	0	11.4kb	5.7kb
green	open	.monitoring-es-7-2022.03.20	SyOns3d-QU6ysbFDj	1	1	224781	43812	246.6mb	124.1mb
green	open	.monitoring-kibana-7-2022.03.18	gOvcKvRlQ9O-PipQM	1	1	10700	0	3.4mb	1.7mb
green	open	.monitoring-kibana-7-2022.03.19	IwSi_UIYQ5eFSyUJk	1	1	17280	0	5.8mb	2.9mb

2. Check whether the incremental data is migrated.

Query the data records that are most recently updated in the self-managed Elasticsearch cluster.

```
GET kibana_sample_data_logs/_search
{
  "query": {
    "range": {
      "@timestamp": {
        "gte": "now-5m",
        "lte": "now/m"
      }
    }
  },
  "sort": [
    {
      "@timestamp": {
        "order": "desc"
      }
    }
  ]
}
```

The following result is returned.



Run the same command to query the data records that are most recently updated in the Alibaba Cloud Elasticsearch cluster. If the incremental data is migrated, the data records that are most recently updated in the Alibaba Cloud Elasticsearch cluster are the same as those in the self-managed Elasticsearch cluster.

2.2.4. Use the reindex API to migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster

This topic describes how to use the reindex API to migrate data from a self-managed Elasticsearch cluster that runs on Elastic Compute Service (ECS) instances to an Alibaba Cloud Elasticsearch cluster. Related operations include index creation and data migration.

Background information

You can use the reindex API to migrate data only to single-zone Alibaba Cloud Elasticsearch clusters. If you want to migrate data to a multi-zone Alibaba Cloud Elasticsearch cluster, we recommend that you use one of the following methods:

- If the self-managed Elasticsearch cluster stores large volumes of data, use snapshots stored in Object Storage Service (OSS). For more information, see Use OSS to migrate data from a user-created Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster.
- If you want to filter source data, use Logstash. For more information, see Use Alibaba Cloud Logstash to migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster.

Prerequisites

- A single-zone Alibaba Cloud Elasticsearch cluster is created.

For more information, see Create an Alibaba Cloud Elasticsearch cluster.

- A self-managed Elasticsearch cluster and the data to be migrated are prepared.

  We recommend that you use a self-managed Elasticsearch cluster deployed on Alibaba Cloud ECS instances. For more information about how to deploy a self-managed Elasticsearch cluster, see Installing and Running Elasticsearch. The self-managed Elasticsearch cluster must meet the following requirements:

  - The ECS instances that host the self-managed Elasticsearch cluster are deployed in the same virtual private cloud (VPC) as the Alibaba Cloud Elasticsearch cluster. You cannot use an ECS instance that is connected to a VPC over a ClassicLink.
  - The IP addresses of nodes in the Alibaba Cloud Elasticsearch cluster are added to the security groups of the ECS instances that host the self-managed Elasticsearch cluster. You can query the IP addresses of the nodes in the Kibana console of the Alibaba Cloud Elasticsearch cluster. In addition, port 9200 is enabled.
  - The self-managed Elasticsearch cluster is connected to the Alibaba Cloud Elasticsearch cluster. You can test the connectivity by running the `curl -XGET http://<host>:9200` command on the server where you run scripts.

    > **Note** You can run all scripts provided in this topic on a server that can be connected to both the self-managed Elasticsearch cluster and Alibaba Cloud Elasticsearch cluster over port 9200.

Limits

The network architecture of Alibaba Cloud Elasticsearch was adjusted in October 2020. Alibaba Cloud Elasticsearch clusters created before October 2020 are deployed in the original network architecture. Alibaba Cloud Elasticsearch cluster created in October 2020 or later are deployed in the new network architecture. Due to the adjustment of the network architecture, you cannot use the reindex API to migrate data between clusters in some scenarios. The following table describes the scenarios and provides data migration solutions in these scenarios.

Scenario	Network architecture	Support for the reindex API	Solution
Migrate data between Alibaba Cloud Elasticsearch clusters	Both clusters are deployed in the original network architecture.	Yes	For more information, see Use the reindex API to migrate data.
	Both clusters are deployed in the new network architecture.	No	Use OSS or Logstash to migrate data between the clusters. For more information, see Use OSS to migrate data from a user-created Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster and Use Alibaba Cloud Logstash to migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster.
	One is deployed in the original network architecture, and the other is deployed in the new network architecture.	No	

Scenario	Network architecture	Support for the reindex API	Solution
Migrate data from a self-managed Elasticsearch cluster that runs on ECS instances to an Alibaba Cloud Elasticsearch cluster	The Alibaba Cloud Elasticsearch cluster is deployed in the original network architecture.	Yes	For more information, see Use the reindex API to migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster.
	The Alibaba Cloud Elasticsearch cluster is deployed in the new network architecture.	Yes	Use the PrivateLink service to establish a private connection between the VPC where the Alibaba Cloud Elasticsearch cluster resides and the VPC where the self-managed Elasticsearch cluster resides. Then, use the domain name of the endpoint you obtained and the reindex API to migrate data between the clusters. For more information, see Migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster deployed in the new network architecture. **Note** Only some regions support PrivateLink. For more information, see Regions and zones that support PrivateLink. If the zone where your Alibaba Cloud Elasticsearch cluster resides does not support PrivateLink, you cannot use the reindex API to migrate data between the clusters.

Precautions

-
- Alibaba Cloud Elasticsearch clusters deployed in the new network architecture reside in the VPC within the service account of Alibaba Cloud Elasticsearch. These clusters cannot access resources in other network environments. Alibaba Cloud Elasticsearch clusters deployed in the original network architecture reside in VPCs that are created by users. These clusters can access resources in other network environments.
- To ensure data consistency and normal data read, we recommend that you do not write data to the self-managed Elasticsearch cluster during the migration. After the migration, you can read data from and write data to the Alibaba Cloud Elasticsearch cluster. If you want to write data to the self-managed Elasticsearch cluster during the migration, we recommend that you configure loop

execution for the reindex operation to shorten the time during which write operations are suspended. For more information, see the method used to migrate a large volume of data (without deletions and with update time) in Step 4: Migrate data.

- If you connect to the self-managed Elasticsearch cluster or the Alibaba Cloud Elasticsearch cluster by using its domain name, do not include path in the URL, such as `http://host:port/path`.

Procedure

1. Step 1: (Optional) Obtain the domain name of an endpoint
2. Step 2: Create destination indexes
3. Step 3: Configure a remote reindex whitelist for the Alibaba Cloud Elasticsearch cluster
4. Step 4: Migrate data

Step 1: (Optional) Obtain the domain name of an endpoint

Alibaba Cloud Elasticsearch clusters created in October 2020 or later are deployed in the new network architecture. These clusters reside in the VPC within the service account of Alibaba Cloud Elasticsearch. If your Alibaba Cloud Elasticsearch cluster is deployed in the new network architecture, you need to use the PrivateLink service to establish a private connection between the VPC and your VPC. Then, obtain the domain name of the related endpoint for future use. To obtain the domain name, perform the following steps:

1. Create a Classic Load Balancer (CLB) instance that supports the PrivateLink service and resides in the same VPC as the Alibaba Cloud Elasticsearch cluster. For more information, see Step 1: Create a CLB instance that supports PrivateLink.
2. Configure the CLB instance. For more information, see Step 2: Configure the CLB instance. You must add all ECS instances that host the self-managed Elasticsearch cluster to the CLB instance as backend servers and specify port 9200 as the listening port.
3. Create an endpoint service. For more information, see Step 3: Create an endpoint service.
4. Obtain the domain name of the endpoint that is used to access the endpoint service. For more information, see View the domain name of an endpoint.

   Record the obtained domain name, which is required in Step 3: Configure a remote reindex whitelist for the Alibaba Cloud Elasticsearch cluster.

Step 2: Create destination indexes

Create destination indexes on the Alibaba Cloud Elasticsearch cluster based on the index settings of the self-managed Elasticsearch cluster. You can also enable the Auto Indexing feature for the Alibaba Cloud Elasticsearch cluster. However, we recommend that you do not use this feature.

The following sample code is a Python script that is used to create multiple indexes on the Alibaba Cloud Elasticsearch cluster at a time. By default, no replica shards are configured for these indexes.

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
# File name: indiceCreate.py
import sys
import base64
import time
import httplib
import json
## Specify the host of the self-managed Elasticsearch cluster.
```

```
oldClusterHost = "old-cluster.com"
## Specify the username of the self-managed Elasticsearch cluster. The field can be empty.
oldClusterUserName = "old-username"
## Specify the password of the self-managed Elasticsearch cluster. The field can be empty.
oldClusterPassword = "old-password"
## Specify the host of the Alibaba Cloud Elasticsearch cluster. You can obtain the information from the Basic Information page of the Alibaba Cloud Elasticsearch cluster in the Alibaba Cloud Elasticsearch console.
newClusterHost = "new-cluster.com"
## Specify the username of the Alibaba Cloud Elasticsearch cluster.
newClusterUser = "elastic"
## Specify the password of the Alibaba Cloud Elasticsearch cluster.
newClusterPassword = "new-password"
DEFAULT_REPLICAS = 0
def httpRequest(method, host, endpoint, params="", username="", password=""):
    conn = httplib.HTTPConnection(host)
    headers = {}
    if (username != "") :
        'Hello {name}, your age is {age} !'.format(name = 'Tom', age = '20')
        base64string = base64.encodestring('{username}:{password}'.format(username = username, password = password)).replace('\n', '')
        headers["Authorization"] = "Basic %s" % base64string;
    if "GET" == method:
        headers["Content-Type"] = "application/x-www-form-urlencoded"
        conn.request(method=method, url=endpoint, headers=headers)
    else :
        headers["Content-Type"] = "application/json"
        conn.request(method=method, url=endpoint, body=params, headers=headers)
    response = conn.getresponse()
    res = response.read()
    return res
def httpGet(host, endpoint, username="", password=""):
    return httpRequest("GET", host, endpoint, "", username, password)
def httpPost(host, endpoint, params, username="", password=""):
    return httpRequest("POST", host, endpoint, params, username, password)
def httpPut(host, endpoint, params, username="", password=""):
    return httpRequest("PUT", host, endpoint, params, username, password)
def getIndices(host, username="", password=""):
    endpoint = "/_cat/indices"
    indicesResult = httpGet(oldClusterHost, endpoint, oldClusterUserName, oldClusterPassword)
    indicesList = indicesResult.split("\n")
    indexList = []
    for indices in indicesList:
        if (indices.find("open") > 0):
            indexList.append(indices.split()[2])
    return indexList
def getSettings(index, host, username="", password=""):
    endpoint = "/" + index + "/_settings"
    indexSettings = httpGet(host, endpoint, username, password)
    print index + "  Original settings: \n" + indexSettings
    settingsDict = json.loads(indexSettings)
    ## By default, the number of primary shards is the same as that for the indexes on the self-managed Elasticsearch cluster.
```

```
    number_of_shards = settingsDict[index]["settings"]["index"]["number_of_shards"]
    ## The default number of replica shards is 0.
    number_of_replicas = DEFAULT_REPLICAS
    newSetting = "\"settings\": {\"number_of_shards\": %s, \"number_of_replicas\": %s}" % (
number_of_shards, number_of_replicas)
    return newSetting
def getMapping(index, host, username="", password=""):
    endpoint = "/" + index + "/_mapping"
    indexMapping = httpGet(host, endpoint, username, password)
    print index + " Original mappings: \n" + indexMapping
    mappingDict = json.loads(indexMapping)
    mappings = json.dumps(mappingDict[index]["mappings"])
    newMapping = "\"mappings\" : " + mappings
    return newMapping
def createIndexStatement(oldIndexName):
    settingStr = getSettings(oldIndexName, oldClusterHost, oldClusterUserName, oldClusterPa
ssword)
    mappingStr = getMapping(oldIndexName, oldClusterHost, oldClusterUserName, oldClusterPas
sword)
    createstatement = "{\n" + str(settingStr) + ",\n" + str(mappingStr) + "\n}"
    return createstatement
def createIndex(oldIndexName, newIndexName=""):
    if (newIndexName == "") :
        newIndexName = oldIndexName
    createstatement = createIndexStatement(oldIndexName)
    print "New index " + newIndexName + " Index settings and mappings: \n" + createstatemen
t
    endpoint = "/" + newIndexName
    createResult = httpPut(newClusterHost, endpoint, createstatement, newClusterUser, newCl
usterPassword)
    print "New index " + newIndexName + " Creation result: " + createResult
## main
indexList = getIndices(oldClusterHost, oldClusterUserName, oldClusterPassword)
systemIndex = []
for index in indexList:
    if (index.startswith(".")):
        systemIndex.append(index)
    else :
        createIndex(index, index)
if (len(systemIndex) > 0) :
    for index in systemIndex:
        print index + " It may be a system index and will not be recreated. You can manuall
y recreate the index based on your business requirements."
```

Step 3: Configure a remote reindex whitelist for the Alibaba Cloud Elasticsearch cluster

1.
2.
3.
4. In the left-side navigation pane of the page that appears, choose **Configuration and Management > Cluster Configuration**.

5. On the page that appears, click **Modify Configuration** on the right side of **YML Configuration**.
6. In the **Other Configurations** field of the **YML File Configuration** panel, configure a remote reindex whitelist.

   The following code provides a configuration example:

   ```
   reindex.remote.whitelist: ["10.0.xx.xx:9200","10.0.xx.xx:9200","10.0.xx.xx:9200","10.15
   .xx.xx:9200","10.15.xx.xx:9200","10.15.xx.xx:9200"]
   ```

   The reindex.remote.whitelist parameter is used to configure a remote reindex whitelist. When you configure the whitelist, you must add the IP addresses of the hosts in the self-managed Elasticsearch cluster to the whitelist. The configuration rules vary based on the network architecture in which the Alibaba Cloud Elasticsearch cluster is deployed.

   - If the Alibaba Cloud Elasticsearch cluster is deployed in the original network architecture, you must configure this parameter in the format of Host:Port number. Separate multiple configurations with commas (,), such as otherhost:9200,another:9200,127.0.10.**:9200,localhost:**. Protocols cannot be identified.
   - If the Alibaba Cloud Elasticsearch cluster is deployed in the new network architecture, you must configure this parameter in the format of Domain name of the related endpoint:Port number, such as ep-bp1hfkx7coy8lvu4****-cn-hangzhou-i.epsrv-bp1zczi0fgoc5qtv****.cn-hangzhou.privatelink.aliyuncs.com:9200. You can obtain the domain name of the related endpoint based on the instructions in Step 1: (Optional) Obtain the domain name of an endpoint. For more information, see View the domain name of an endpoint.

   > **Note** For more information about other parameters, see Configure the YML file.

7.

Step 4: Migrate data

This section describes how to migrate data to an Alibaba Cloud Elasticsearch cluster deployed in the original network architecture. You can use one of the following methods to migrate data. Select a method based on the volume of data that you want to migrate and your business requirements.

Migrate a small volume of data

Run the following script:

```
#!/bin/bash
# file:reindex.sh
indexName="The name of the index"
newClusterUser="The username of the Alibaba Cloud Elasticsearch cluster"
newClusterPass="The password of the Alibaba Cloud Elasticsearch cluster"
newClusterHost="The host of the Alibaba Cloud Elasticsearch cluster"
oldClusterUser="The username of the self-managed Elasticsearch cluster"
oldClusterPass="The password of the self-managed Elasticsearch cluster"
# You must configure the host of the self-managed Elasticsearch cluster in the format of [s
cheme]://[host]:[port]. Example: http://10.37.*.*:9200.
oldClusterHost="The host of the self-managed Elasticsearch cluster"
curl -u ${newClusterUser}:${newClusterPass} -XPOST "http://${newClusterHost}/_reindex?prett
y" -H "Content-Type: application/json" -d'{
    "source": {
        "remote": {
            "host": "'${oldClusterHost}'",
            "username": "'${oldClusterUser}'",
            "password": "'${oldClusterPass}'"
        },
        "index": "'${indexName}'",
        "query": {
            "match_all": {}
        }
    },
    "dest": {
       "index": "'${indexName}'"
    }
}'
```

Migrate a large volume of data (without deletions and with update time)

To migrate a large volume of data without deletions, you can perform a rolling update to shorten the time during which write operations are suspended. The rolling update requires that your data schema has a time-series attribute that indicates the update time. You can stop writing data to the self-managed Elasticsearch cluster after data is migrated. Then, use the reindex API to perform a rolling update to synchronize the data that is updated during the migration. After the rolling update is complete, you can read data from and write data to the Alibaba Cloud Elasticsearch cluster.

```
#!/bin/bash
# file: circleReindex.sh
# CONTROLLING STARTUP:
# This is a script that uses the reindex API to remotely reindex data. Requirements:
# 1. Indexes are created on the Alibaba Cloud Elasticsearch cluster, or the Auto Indexing a
nd dynamic mapping features are enabled for the cluster.
# 2. A remote reindex whitelist is configured for the Alibaba Cloud Elasticsearch cluster i
n the YML File Configuration panel of the Alibaba Cloud Elasticsearch console. For example,
the following information is specified in the Other Configurations field: reindex.remote.wh
itelist: 172.16.**.**:9200.
# 3. The host is configured in the format of [scheme]://[host]:[port].
USAGE="Usage: sh circleReindex.sh <count>
       count: the number of reindex operations that you can perform. A negative number indi
cates loop execution.
```

```
Example:
        sh circleReindex.sh 1
        sh circleReindex.sh 5
        sh circleReindex.sh -1"
indexName="The name of the index"
newClusterUser="The username of the Alibaba Cloud Elasticsearch cluster"
newClusterPass="The password of the Alibaba Cloud Elasticsearch cluster"
oldClusterUser="The username of the self-managed Elasticsearch cluster"
oldClusterPass="The password of the self-managed Elasticsearch cluster"
## http://myescluster.com
newClusterHost="The host of the Alibaba Cloud Elasticsearch cluster"
# You must configure the host of the self-managed Elasticsearch cluster in the format of [s
cheme]://[host]:[port]. Example: http://10.37.*.*:9200.
oldClusterHost="The host of the self-managed Elasticsearch cluster"
timeField="The update time of data"
reindexTimes=0
lastTimestamp=0
curTimestamp=`date +%s`
hasError=false
function reIndexOP() {
    reindexTimes=$[${reindexTimes} + 1]
    curTimestamp=`date +%s`
    ret=`curl -u ${newClusterUser}:${newClusterPass} -XPOST "${newClusterHost}/_reindex?pre
tty" -H "Content-Type: application/json" -d '{
        "source": {
            "remote": {
                "host": "'${oldClusterHost}'",
                "username": "'${oldClusterUser}'",
                "password": "'${oldClusterPass}'"
            },
            "index": "'${indexName}'",
            "query": {
                "range" : {
                    "'${timeField}'" : {
                        "gte" : '${lastTimestamp}',
                        "lt" : '${curTimestamp}'
                    }
                }
            }
        },
        "dest": {
            "index": "'${indexName}'"
        }
    }'`
    lastTimestamp=${curTimestamp}
    echo "${reindexTimes} reindex operations are performed. The last reindex operation is c
omplete at ${lastTimestamp}. Result: ${ret}."
    if [[ ${ret} == *error* ]]; then
        hasError=true
        echo "An unknown error occurred when you perform this operation. All subsequent ope
rations are suspended."
    fi
}
function start() {
```

```
    ## A negative number indicates loop execution.
    if [[ $1 -lt 0 ]]; then
        while :
        do
            reIndexOP
        done
    elif [[ $1 -gt 0 ]]; then
        k=0
        while [[ k -lt $1 ]] && [[ ${hasError} == false ]]; do
            reIndexOP
            let ++k
        done
    fi
}
## main
if [ $# -lt 1 ]; then
    echo "$USAGE"
    exit 1
fi
echo "Start the reindex operation for the ${indexName} index."
start $1
echo "${reindexTimes} reindex operations are performed."
```

Migrate a large volume of data (without deletions and update time)

You can migrate a large volume of data if no update time is defined in the index mappings of the self-managed Elasticsearch cluster. However, you must add an update time field to the index mappings. After the field is added, you can migrate existing data. Then, perform a rolling update that is described in the second data migration method to migrate incremental data.

```
#!/bin/bash
# file:miss.sh
indexName="The name of the index"
newClusterUser="The username of the Alibaba Cloud Elasticsearch cluster"
newClusterPass="The password of the Alibaba Cloud Elasticsearch cluster"
newClusterHost="The host of the Alibaba Cloud Elasticsearch cluster"
oldClusterUser="The username of the self-managed Elasticsearch cluster"
oldClusterPass="The password of the self-managed Elasticsearch cluster"
# You must configure the host of the self-managed Elasticsearch cluster in the format of [s
cheme]://[host]:[port]. Example: http://10.37.*.*:9200.
oldClusterHost="The host of the self-managed Elasticsearch cluster"
timeField="updatetime"
curl -u ${newClusterUser}:${newClusterPass} -XPOST "http://${newClusterHost}/_reindex?prett
y" -H "Content-Type: application/json" -d '{
    "source": {
        "remote": {
            "host": "'${oldClusterHost}'",
            "username": "'${oldClusterUser}'",
            "password": "'${oldClusterPass}'"
        },
        "index": "'${indexName}'",
        "query": {
            "bool": {
                "must_not": {
                    "exists": {
                        "field": "'${timeField}'"
                    }
                }
            }
        }
    },
    "dest": {
       "index": "'${indexName}'"
    }
}'
```

FAQ

- Problem: When I run the curl command, the system displays `{"error":"Content-Type header [application/x-www-form-urlencoded] is not supported","status":406}`. What do I do?

  Solution: Add `-H "Content-Type: application/json"` to the curl command and try again.

```
  // Obtain all the indexes on the self-managed Elasticsearch cluster. If you do not have the required permissions, remove the "-u user:pass" parameter. Replace oldClusterHost with the information about the host of the self-managed Elasticsearch cluster.
  curl -u user:pass -XGET http://oldClusterHost/_cat/indices | awk '{print $3}'
  // Obtain the settings and mappings of the index that you want to migrate for the specified user based on the returned indexes. Replace indexName with the index name that you want to query.
  curl -u user:pass -XGET http://oldClusterHost/indexName/_settings,_mapping?pretty=true
  // Create an index on the Alibaba Cloud Elasticsearch cluster based on the settings and mappings that you obtained. You can set the number of replica shards to 0 to accelerate data migration, and change the number to 1 after data is migrated.
  // Replace newClusterHost with the host information of the Alibaba Cloud Elasticsearch cluster, testindex with the name of the index that you have created, and testtype with the type of the index.
  curl -u user:pass -XPUT http://<newClusterHost>/<testindex> -d '{
    "testindex" : {
        "settings" : {
            "number_of_shards" : "5", // Specify the number of primary shards for the index on the self-managed Elasticsearch cluster, such as 5.
            "number_of_replicas" : "0" // Set the number of replica shards to 0.
          }
        },
        "mappings" : { // Specify the mappings of the index on the self-managed Elasticsearch cluster. Example:
            "testtype" : {
                "properties" : {
                    "uid" : {
                        "type" : "long"
                    },
                    "name" : {
                        "type" : "text"
                    },
                    "create_time" : {
                      "type" : "long"
                    }
                }
            }
       }
   }
}'
```

- Problem: What do I do if the source index stores large volumes of data and the data migration is slow?

  Solution:

  - If you use the reindex API to migrate data, data is migrated in scroll mode. To improve the efficiency of data migration, you can increase the scroll size or configure a sliced scroll. The sliced scroll can parallelize the reindex process. For more information, see the reindex API.
  - If the self-managed Elasticsearch cluster stores large volumes of data, we recommend that you use snapshots stored in OSS to migrate data. For more information, see Use OSS to migrate data from a user-created Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster.

- If the source index stores large volumes of data, you can set the number of replica shards to 0 and the refresh interval to -1 for the destination index before you migrate data to accelerate data migration. After data is migrated, restore the settings to the original values.

```
// You can set the number of replica shards to 0 and disable the refresh feature to acc
elerate the data migration.
curl -u user:password -XPUT 'http://<host:port>/indexName/_settings' -d' {
        "number_of_replicas" : 0,
        "refresh_interval" : "-1"
}'
// After data is migrated, set the number of replica shards to 1 and the refresh interv
al to 1s, which is the default value.
curl -u user:password -XPUT 'http://<host:port>/indexName/_settings' -d' {
        "number_of_replicas" : 1,
        "refresh_interval" : "1s"
}'
```

2.2.5. Migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster deployed in the new network architecture

This topic describes how to migrate data from a self-managed Elasticsearch cluster that runs on Elastic Compute Service (ECS) instances to an Alibaba Cloud Elasticsearch cluster that is deployed in the new network architecture. You can use PrivateLink to establish a private connection to the Alibaba Cloud Elasticsearch cluster and use the reindex API to migrate data. The reindex API includes two operations: index creation and data migration.

Prerequisites

- The self-managed Elasticsearch cluster meets the following requirements:
  -
  -
  -
  - The source index is prepared. In this example, the source index shown in the following figure is used.

```
[root@elasticsearch1 ~]# curl -XGET  http://172.16.  :9200/_cat/indices?v
health status index  uuid                   pri rep docs.count docs.deleted store.size pri.store.size
green  open   source lGFcaUIgT1-Nsj9b_EezAQ   1   1          6            0     28.2kb         19.1kb
green  open   dest   Kn3Tu9TmT62J4ouMHi_37w   1   1          6            0     23.3kb          9.1kb
[root@elasticsearch1 ~]#
```

- The Alibaba Cloud Elasticsearch cluster meets the following requirements:
  - The Auto Indexing feature is enabled for the cluster, or the destination index is created in the cluster.
  - Default whitelists are used.

Limits

The network architecture of Alibaba Cloud Elasticsearch was adjusted in October 2020. Alibaba Cloud Elasticsearch clusters created before October 2020 are deployed in the original network architecture. Alibaba Cloud Elasticsearch cluster created in October 2020 or later are deployed in the new network architecture. Due to the adjustment of the network architecture, you cannot use the reindex API to migrate data between clusters in some scenarios. The following table describes the scenarios and provides data migration solutions in these scenarios.

Scenario	Network architecture	Support for the reindex API	Solution
Migrate data between Alibaba Cloud Elasticsearch clusters	Both clusters are deployed in the original network architecture.	Yes	For more information, see Use the reindex API to migrate data.
	Both clusters are deployed in the new network architecture.	No	Use OSS or Logstash to migrate data between the clusters. For more information, see Use OSS to migrate data from a user-created Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster and Use Alibaba Cloud Logstash to migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster.
	One is deployed in the original network architecture, and the other is deployed in the new network architecture.	No	
Migrate data from a self-	The Alibaba Cloud Elasticsearch cluster is deployed in the original network architecture.	Yes	For more information, see Use the reindex API to migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster.

Scenario	Network architecture	Support for the reindex API	Solution
managed Elasticsearch cluster that runs on ECS instances to an Alibaba Cloud Elasticsearch cluster	The Alibaba Cloud Elasticsearch cluster is deployed in the new network architecture.	Yes	Use the PrivateLink service to establish a private connection between the VPC where the Alibaba Cloud Elasticsearch cluster resides and the VPC where the self-managed Elasticsearch cluster resides. Then, use the domain name of the endpoint you obtained and the reindex API to migrate data between the clusters. For more information, see Migrate data from a self-managed Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster deployed in the new network architecture. **Note** Only some regions support PrivateLink. For more information, see Regions and zones that support PrivateLink. If the zone where your Alibaba Cloud Elasticsearch cluster resides does not support PrivateLink, you cannot use the reindex API to migrate data between the clusters.

Procedure

1. Step 1: Configure a CLB instance that supports PrivateLink

   Only Classic Load Balancer (CLB) instances that support PrivateLink can serve as service resources for endpoint services. Before you use PrivateLink to establish private connections to access services across VPCs, you must create a CLB instance that supports PrivateLink and configure listening settings for the CLB instance.

2. Step 2: Create an endpoint service

   After you create an endpoint service in a VPC, you can use an endpoint that is deployed in another VPC to access the endpoint service over a private connection.

3. Step 3: Configure a private connection to the Alibaba Cloud Elasticsearch cluster

   In the Elasticsearch console, associate the Alibaba Cloud Elasticsearch cluster with the endpoint service that is created in Step 2.

4. Step 4: Obtain the domain name of the endpoint

After the Alibaba Cloud Elasticsearch cluster is associated with the endpoint service, you can obtain the domain name of the associated endpoint.

5. Step 5: Configure a remote reindex whitelist for the Alibaba Cloud Elasticsearch cluster

   In the Elasticsearch console, add the domain name that is obtained in Step 4 to the remote reindex whitelist of the Alibaba Cloud Elasticsearch cluster for authorization.

6. Step 6: Migrate data

   After you complete the preceding steps, you can migrate data from the self-managed Elasticsearch cluster to the Alibaba Cloud Elasticsearch cluster.

Step 1: Configure a CLB instance that supports PrivateLink

1. Create a CLB instance.

   Make sure that the CLB instance and the ECS instances that act as backend servers are deployed in the same region. For more information, see Create a CLB instance that supports PrivateLink.

   > **Note** Only some regions support PrivateLink. For more information, see Regions and zones that support PrivateLink. If the zone where your Alibaba Cloud Elasticsearch cluster resides does not support PrivateLink, you cannot use the reindex API to migrate data between the clusters.

2. Configure **protocol and listening settings**. Set Select Listener Protocol to **TCP** and Listening Port to **9200**.

   For more information, see Configure protocol and listening settings.

3. Configure **backend servers**. Add the ECS instances that host the self-managed Elasticsearch cluster as backend servers and specify port **9200** for the ECS instances.

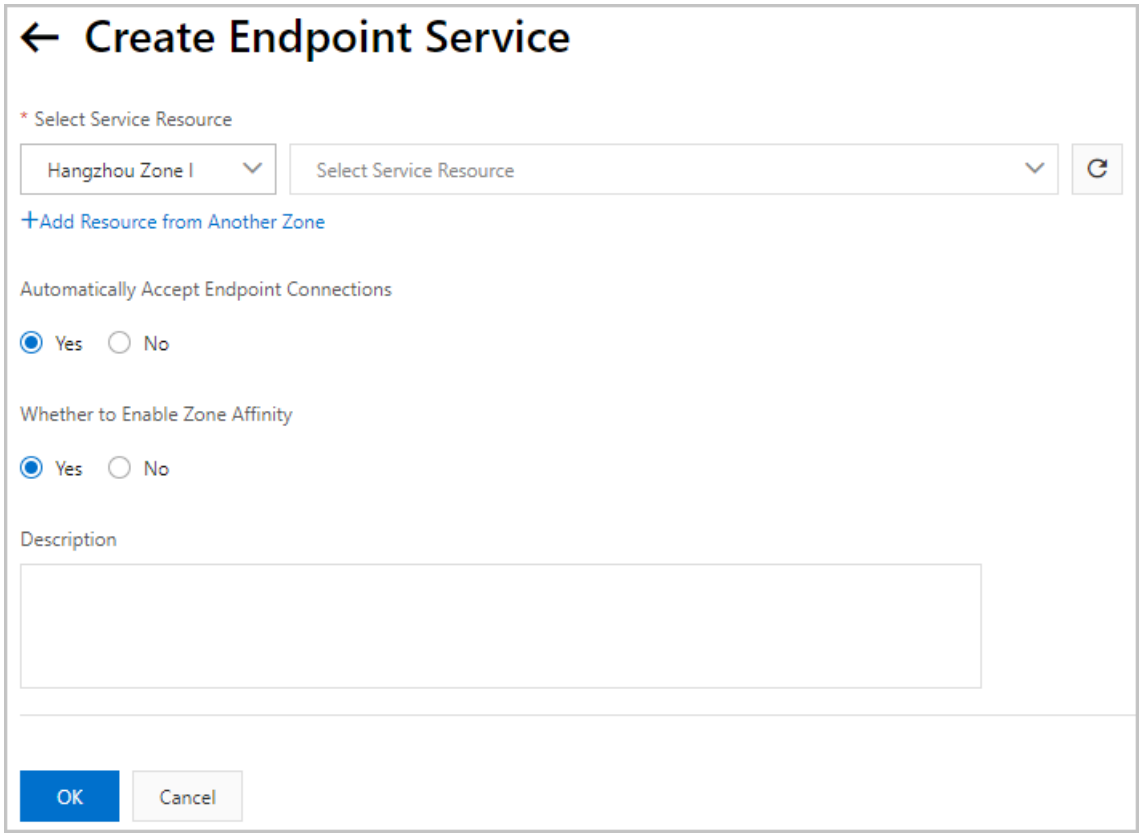
   For more information, see Configure backend servers.

4.
5.
6. In the **Configure Server Load Balancer** message, click **OK**. The **Instances** page appears.

   If the health check status of an ECS instance is Normal, the ECS instance is ready to process requests.

Step 2: Create an endpoint service

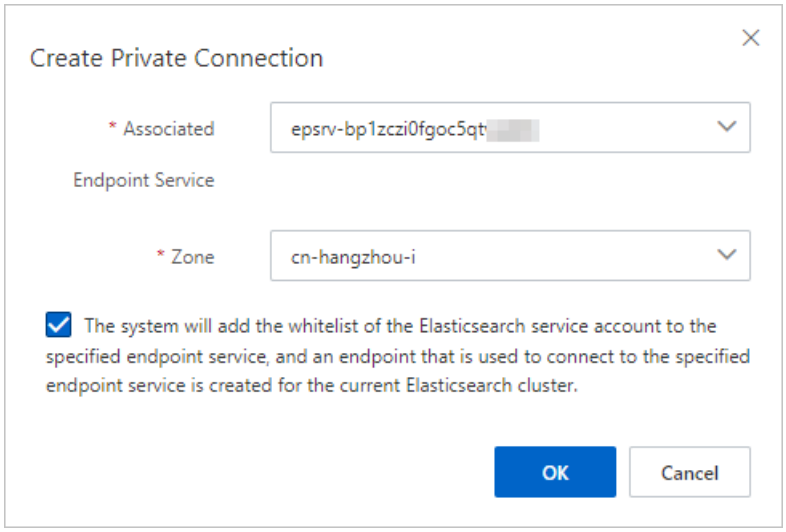
1.
2.
3.
4.
5. On the **Create Endpoint Service** page, configure the parameters based on your business requirements.



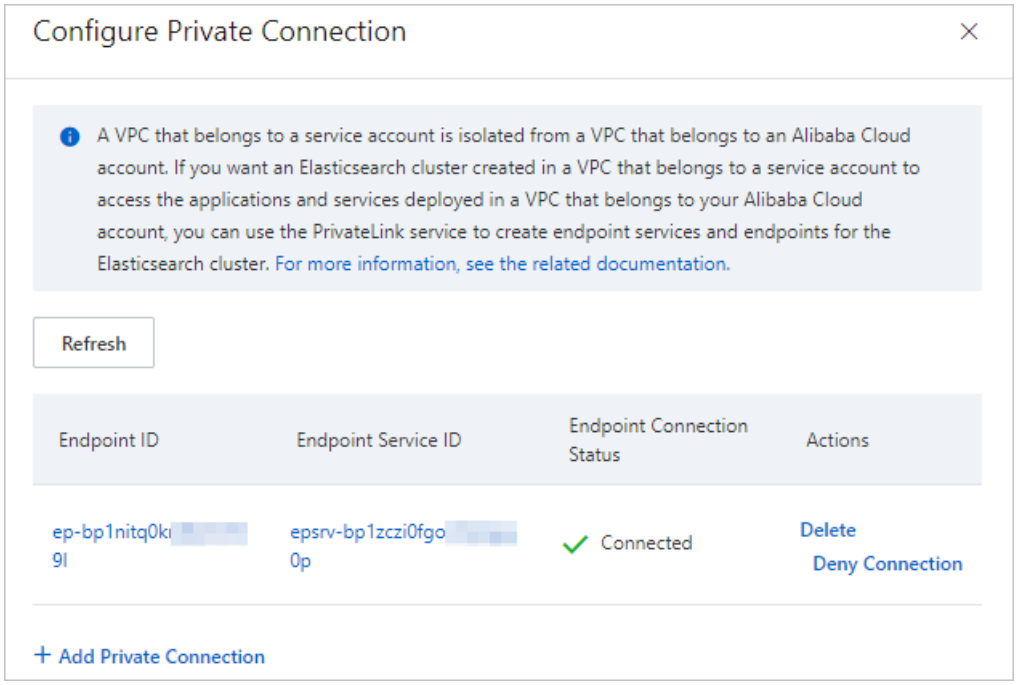
6.

Step 3: Configure a private connection to the Alibaba Cloud Elasticsearch cluster

1.
2.
3.
4.
5.
6. In the **Configure Private Connection** panel, click **Add Private Connection**. In the Create Private Connection dialog box, select the endpoint service that is created in Step 2 and select a zone. Then, select the check box.



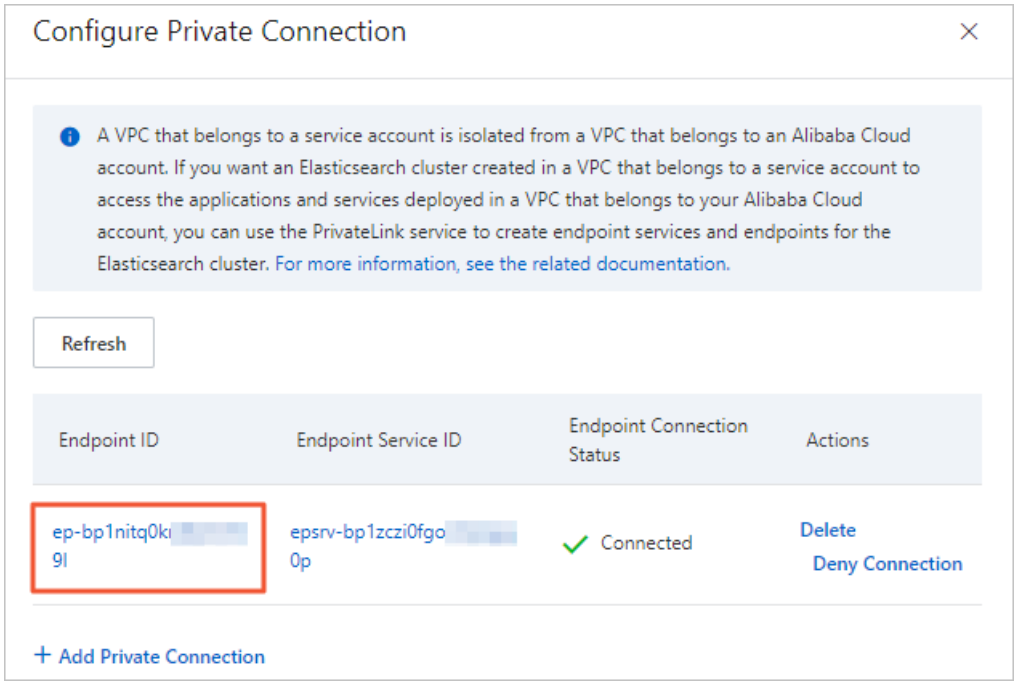
7. Click **OK**. Then, the endpoint service attempts to connect to the associated endpoint. If the value of Endpoint Connection Status is **Connected**, the endpoint service is connected to the associated endpoint.



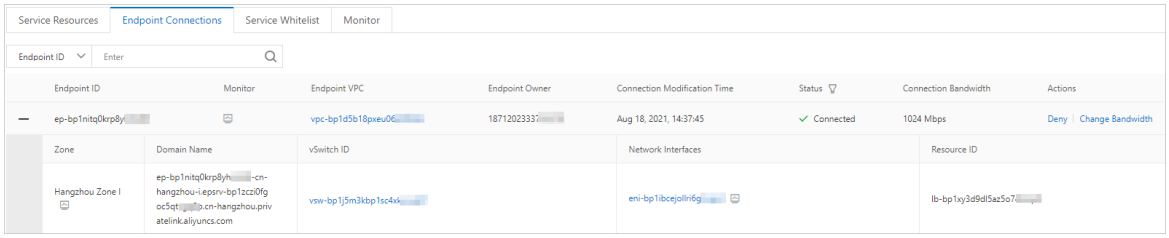
Step 4: Obtain the domain name of the endpoint

After the preceding steps are performed, you must obtain the domain name of the associated endpoint to configure a remote reindex whitelist.

1. In the **Configure Private Connection** panel, click the ID of the endpoint in the **Endpoint ID** column.



2. On the **Endpoint Connections** tab of the page that appears, click the + icon next to the ID of the endpoint. Then, you can view the domain name of the endpoint.



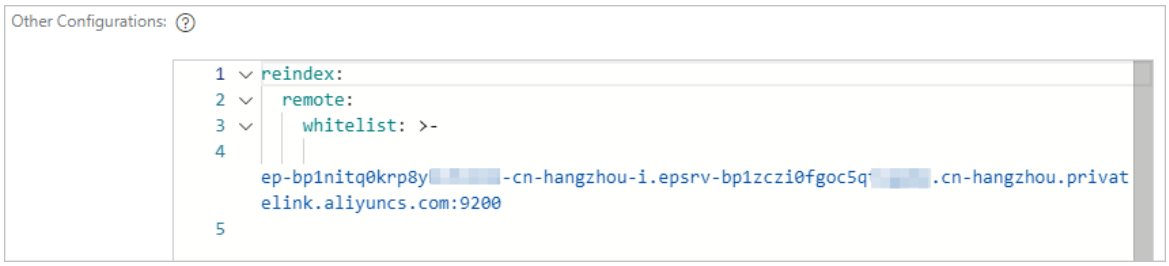
Step 5: Configure a remote reindex whitelist for the Alibaba Cloud Elasticsearch cluster

Notice After you configure a remote reindex whitelist for the Alibaba Cloud Elasticsearch cluster, the system restarts the cluster. We recommend that you perform this operation during off-peak hours.

1.
2.
3.
4. In the left-side navigation pane of the page that appears, choose **Configuration and Management > Cluster Configuration**.
5. On the page that appears, click **Modify Configuration** on the right side of **YML Configuration**.
6. In the **YML File Configuration** panel, specify the domain name that is obtained in Step 4 in **Other Configurations**.

   Sample code:

```
reindex.remote.whitelist: 'ep-bp1nitq0krp8yhcf****-cn-hangzhou-i.epsrv-bp1zczi0fgoc5qtv
****.cn-hangzhou.privatelink.aliyuncs.com:9200'
```



7.

Step 6: Migrate data

1. On the **Dev Tools** page in the Kibana console of the Alibaba Cloud Elasticsearch cluster, run the following command to migrate data.

   > **Note** For more information about how to log on to the Kibana console, see Log on to the Kibana console.

   ```
   POST /_reindex?pretty
   {
     "source": {
       "remote": {
         "host": "http://ep-bp1nitq0krp8yhcf****-cn-hangzhou-i.epsrv-bp1zczi0fgoc5qtv****.
   cn-hangzhou.privatelink.aliyuncs.com:9200",
         "username": "elastic",
         "password": "Elastic@123***"
       },
       "index": "source",
       "size": 5000
     },
     "dest": {
       "index": "dest"
     }
   }
   ```

   For more information, see the reindex API.

2. (Optional)If you want to obtain detailed information about all running reindex requests during data migration, run the following command:

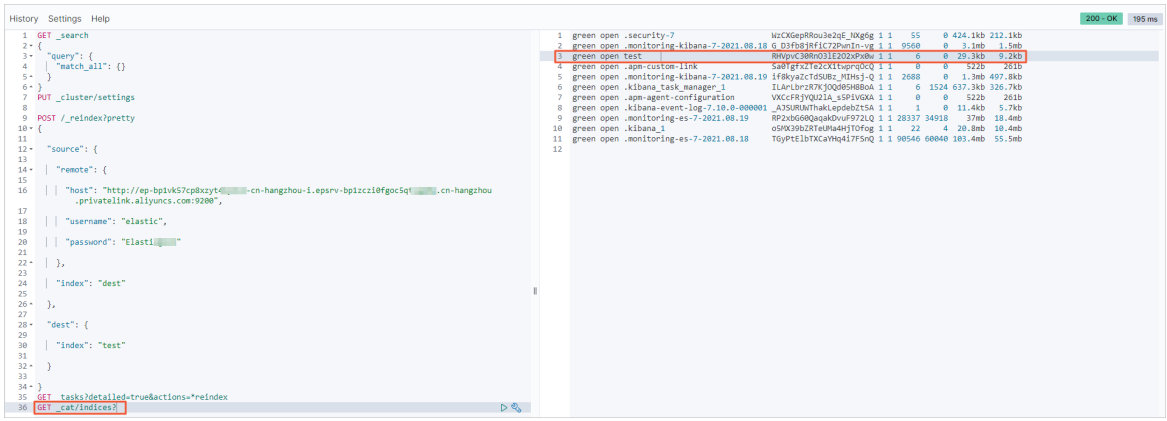
   ```
   GET _tasks?detailed=true&actions=*reindex
   ```

3. View data migration results.

   After the data migration is complete, you can run the following command to view the data migration results:

   ```
   GET _cat/indices?
   ```

   In the following figure, the test index is the destination index. If the health status and data volume of the index are normal, the data migration is successful.



FAQ

Problem: What do I do if the source index stores large volumes of data and the data migration is slow?

Solution:

- If you use the reindex API to migrate data, data is migrated in scroll mode. To improve the efficiency of data migration, you can increase the scroll size or configure a sliced scroll. The sliced scroll can parallelize the reindex process. For more information, see the reindex API.
- If the self-managed Elasticsearch cluster stores large volumes of data, we recommend that you use snapshots stored in Object Storage Service (OSS) to migrate data. For more information, see Use OSS to migrate data from a user-created Elasticsearch cluster to an Alibaba Cloud Elasticsearch cluster.
- If the source index stores large volumes of data, you can set the number of replica shards to 0 and the refresh interval to -1 for the destination index before you migrate data to accelerate data migration. After data is migrated, restore the settings to the original values.

```
// You can set the number of replica shards to 0 and disable the refresh feature to accel
erate the data migration.
curl -u user:password -XPUT 'http://<host:port>/indexName/_settings' -d' {
        "number_of_replicas" : 0,
        "refresh_interval" : "-1"
}'
// After data is migrated, set the number of replica shards to 1 and the refresh interval
to 1s, which is the default value.
curl -u user:password -XPUT 'http://<host:port>/indexName/_settings' -d' {
        "number_of_replicas" : 1,
        "refresh_interval" : "1s"
}'
```

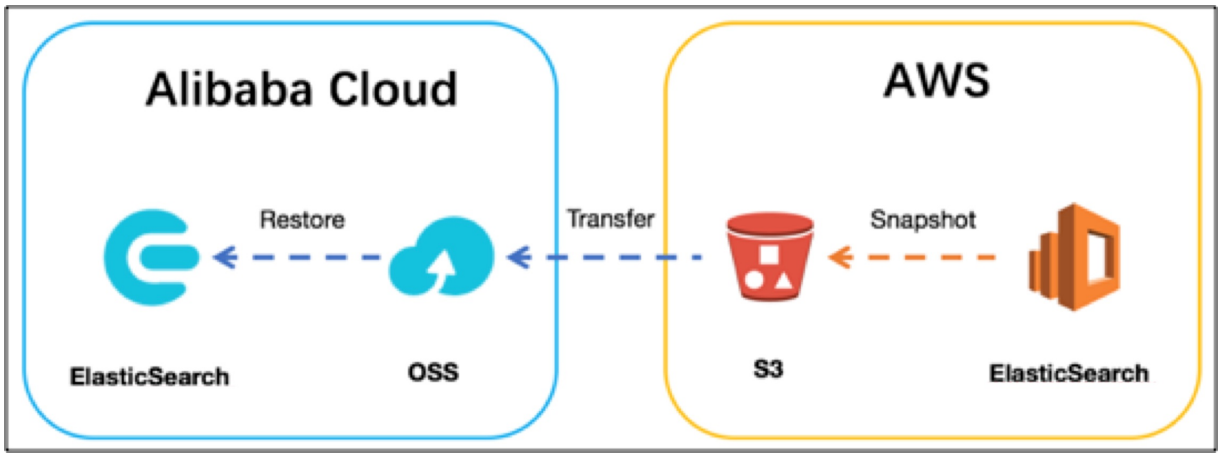
2.3. Migrate data from a third-party Elasticsearch instance to Alibaba Cloud Elasticsearch

2.3.1. Migrate data from an Amazon ES domain to an Alibaba Cloud Elasticsearch cluster

This topic describes how to migrate data from an Amazon Elasticsearch Service (Amazon ES) domain to an Alibaba Cloud Elasticsearch cluster.

Background information

The following figure shows the reference architecture for the migration.



Terms

- Elasticsearch: a distributed, RESTful search and analytics engine designed for various scenarios. As the core of the Elastic Stack, Elasticsearch stores your data in a centralized manner and searches for and analyzes data.
- Kibana: provides a visual interface for you to search for and analyze data.
- Amazon ES: a fully managed service that offers easy-to-use Elasticsearch API operations and real-time analytics capabilities. This service also provides the availability, scalability, and security that are required for production workloads. You can use Amazon ES to easily deploy, protect, manage, and scale Elasticsearch clusters for scenarios such as log analytics, full-text search, and application monitoring.
- Alibaba Cloud Elasticsearch: It is designed based on open source Elasticsearch for scenarios such as data analytics and searches. It provides enterprise-grade access control, automated reporting, and security monitoring and alerting.
- Snapshot and restore: You can store snapshots of individual indexes or an entire cluster in a remote repository like a shared file system, such as Amazon Simple Storage Service (Amazon S3) or HDFS. The snapshots can be used to restore data. However, the data can be restored only to Elasticsearch clusters of specific versions:
  - Data in a snapshot created in an Elasticsearch 5.x cluster can be restored to an Elasticsearch 6.x cluster.
  - Data in a snapshot created in an Elasticsearch 2.x cluster can be restored to an Elasticsearch 5.x cluster.
  - Data in a snapshot created in an Elasticsearch 1.x cluster can be restored to an Elasticsearch 2.x cluster.

Migration plan

To migrate data from an Amazon ES domain to an Alibaba Cloud Elasticsearch cluster, perform the following steps:

1. Create a baseline index.
   i. Create a snapshot repository and associate it with an S3 bucket

i. Create a snapshot repository and associate it with an S3 bucket.

ii. Create the first snapshot for the index whose data you want to migrate. The first snapshot is a full snapshot.

This snapshot is automatically stored in the S3 bucket.

iii. Create an Object Storage Service (OSS) bucket on Alibaba Cloud, and register it with the snapshot repository of your Alibaba Cloud Elasticsearch cluster.

iv. Use ossimport to transfer the full snapshot from the S3 bucket to the OSS bucket.

v. Restore data from the full snapshot to your Alibaba Cloud Elasticsearch cluster.

2. Process incremental snapshots on a regular basis.

Repeat the preceding steps to restore data from incremental snapshots.

3. Identify the final snapshot and perform a service switchover.

i. Stop services that may modify index data.

ii. Create the final snapshot for your Amazon ES domain.

iii. Transfer the final snapshot to your OSS bucket. Then, restore data from the snapshot to your Alibaba Cloud Elasticsearch cluster.

iv. Perform a service switchover to the cluster.

Prerequisites

You have completed the following operations:

- Create an Amazon ES 5.5.2 domain in the Asia Pacific (Singapore) region.

  For more information, see Create an Amazon ES domain.

- Create an Alibaba Cloud Elasticsearch V5.5.3 cluster in the China (Hangzhou) region.

  For more information, see Create an Alibaba Cloud Elasticsearch cluster.

- Create an OSS bucket.

  In this topic, an OSS bucket is created in the China (Hangzhou) region. The storage class of the bucket is Standard, and the access control list (ACL) of the bucket is Private. Default settings are used for other parameters. For more information, see Create buckets.

- Prepare the index whose data you want to migrate. The `movies` index is used in this topic.

Prerequisites for creating manual snapshots in an Amazon ES domain

Amazon ES automatically creates snapshots for the primary index shards in a domain every day and stores them in a pre-configured S3 bucket. These snapshots are retained for a maximum of 14 days free of charge. You can use these snapshots to restore data to the domain. However, you cannot use them to migrate data to other domains. To migrate data, you must use manual snapshots stored in your S3 bucket. Standard S3 charges apply to manual snapshots.

To create manual snapshots and restore data from the snapshots, you must use AWS Identity and Access Management (IAM) and S3. Before you create snapshots, perform the operations that are listed in the following table.

Operation	Description
Create an S3 bucket	The bucket stores the manual snapshots of your Amazon ES domain.
Create an IAM role	The role is used to grant permissions on Amazon ES. When you add a trust relationship for the role, you must specify Amazon ES in the `Principal` element. This role is also required when you register a snapshot repository with Amazon ES. Only IAM users that assume this role can register the snapshot repository.
Create an IAM policy	This policy specifies the actions that S3 can perform on your S3 bucket. The policy must be attached to the IAM role that is used to grant permissions on Amazon ES. You must specify your S3 bucket in the Resource element of the policy.

- Create an S3 bucket

  You need an S3 bucket to store manual snapshots. Take note of its Amazon Resource Name (ARN). The ARN is used by the following items:

  - Resource element in the IAM policy that is attached to your IAM role
  - Python client that is used to register a snapshot repository

  The following example shows the ARN of an S3 bucket:

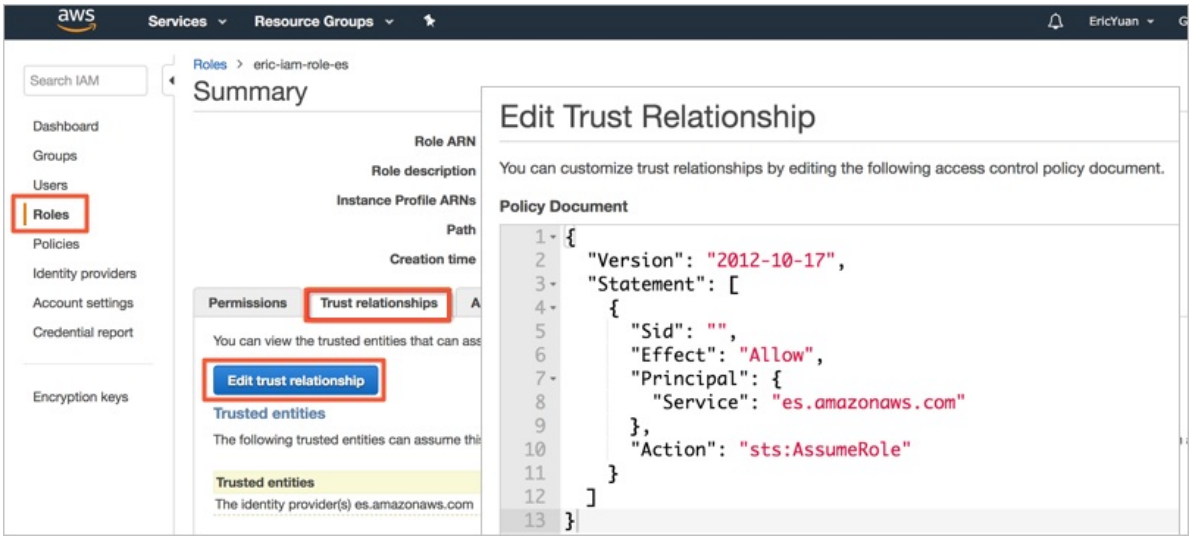
  ```
  arn:aws:s3:::eric-es-index-backups
  ```

- Create an IAM role

  You must have an IAM role, for which Amazon ES (es.amazonaws.com) is specified in the `Service` element in its trust relationship. Example:

  ```
  {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Allow",
        "Principal": {
          "Service": "es.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
      }
    ]
  }
  ```

  You can view the trust relationship details in the AWS IAM console.



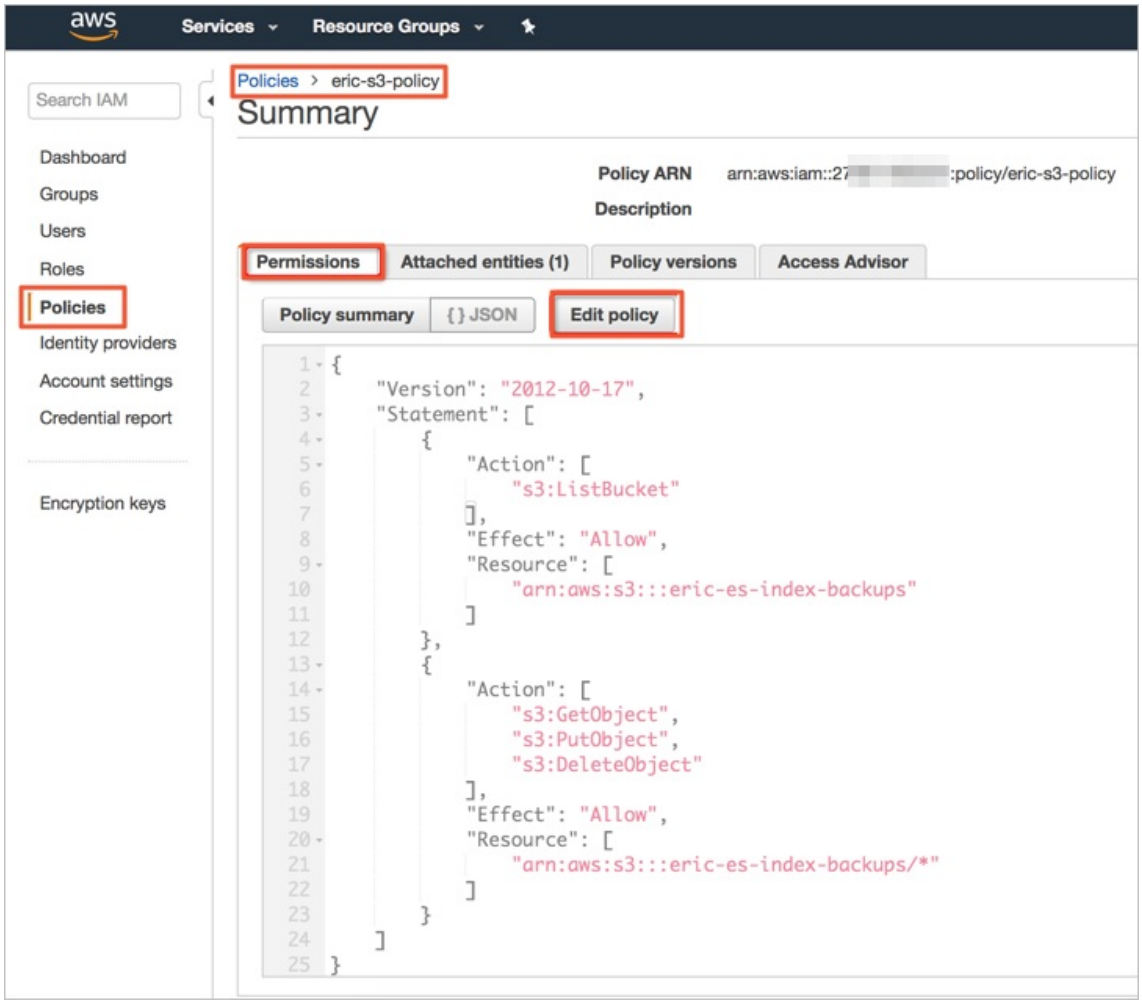
> **Note** When you create a role in the IAM console, Amazon ES is not included in the **Select role type** drop-down list. You can select **Amazon EC2** from the drop-down list and create the role as prompted. Then, change `ec2.amazonaws.com` in the trust relationship of the role to `es.amazonaws.com`.

- Create an IAM policy

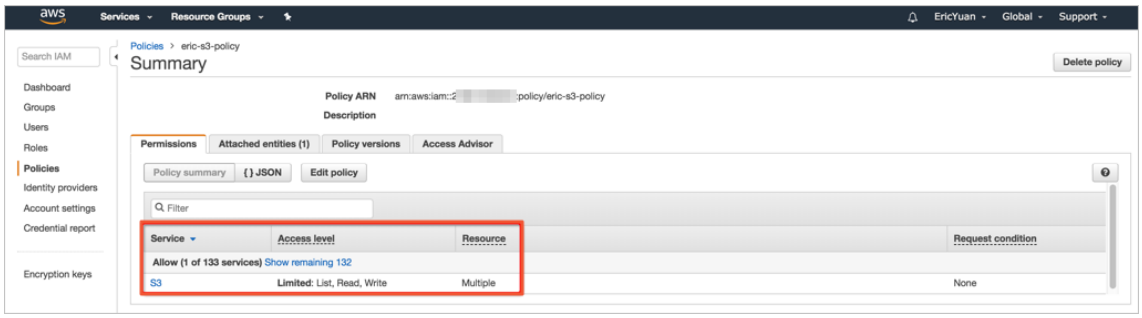
  You must attach an IAM policy to the IAM role. The policy specifies the S3 bucket that is used to store the manual snapshots of your Amazon ES domain. The following example specifies the ARN of the `eric-es-index-backups` bucket:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::eric-es-index-backups"
            ]
        },
        {
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::eric-es-index-backups/*"
            ]
        }
    ]
}
```

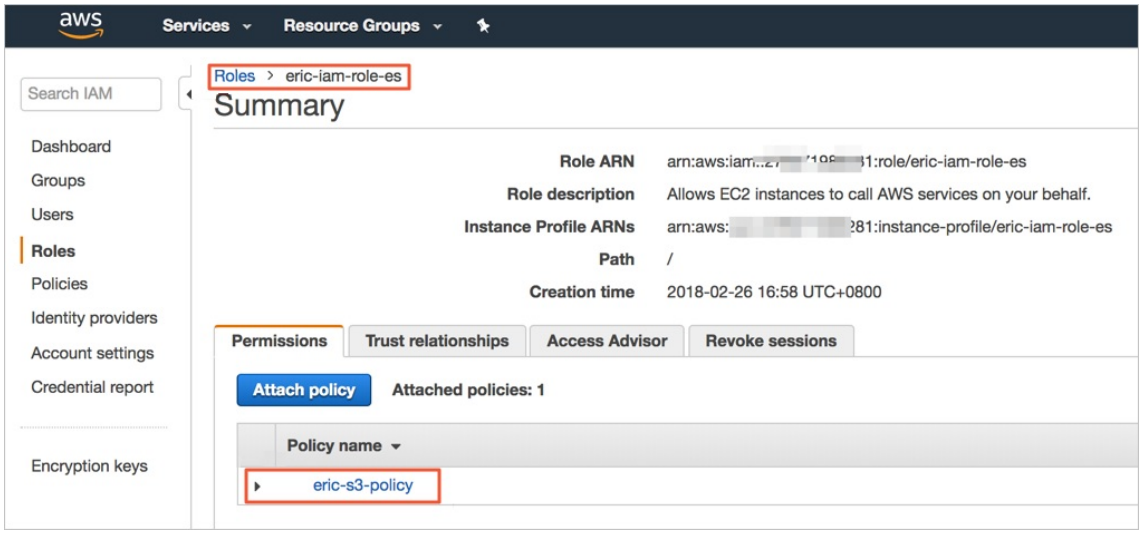
i. Copy the policy content to the Edit policy section.



ii. Check whether the specified policy is correct.



iii. Attach the policy to the role.



Step 1: Register a manual snapshot repository

You can create manual snapshots only after you register a snapshot repository with Amazon ES. Before you create manual snapshots, sign your AWS request to the user or role specified in the trust relationship of the IAM role. For more information, see Prerequisites for creating manual snapshots in an Amazon ES domain.

> **Notice** You cannot use a curl command to register a snapshot repository because the command does not support AWS request signing. Instead, use the sample Python client to register a snapshot repository.

1. Download the Sample Python Client file.
2. Modify the file.

   Change the values highlighted in yellow in the file based on actual conditions. Then, copy the content into a Python file named snapshot.py.

   The following table describes the parameters in the Sample Python Client file.

Parameter	Description
region	The AWS region where the snapshot repository is created.
host	The endpoint of your Amazon ES domain.
aws_access_key_id	The ID of your IAM credential.
aws_secret_access_key	The key of your IAM credential.
path	The path of the snapshot repository.

Parameter	Description
data	The value must include the name and ARN of the S3 bucket for the IAM role that you created in Prerequisites for creating manual snapshots in an Amazon ES domain. **Notice** ◦ If you want to enable server-side encryption with S3-managed keys for the snapshot repository, add `"server_side_encryption": true` to the settings JSON array. ◦ If the S3 bucket resides in the ap-southeast-1 region, replace `"region": "ap-southeast-1"` with `"endpoint": "s3.amazonaws.com"`.

3. Install Amazon Web Services Library boto-2.48.0.

   The preceding sample Python client requires that you install the boto package of version 2.x on the computer where you register your snapshot repository.

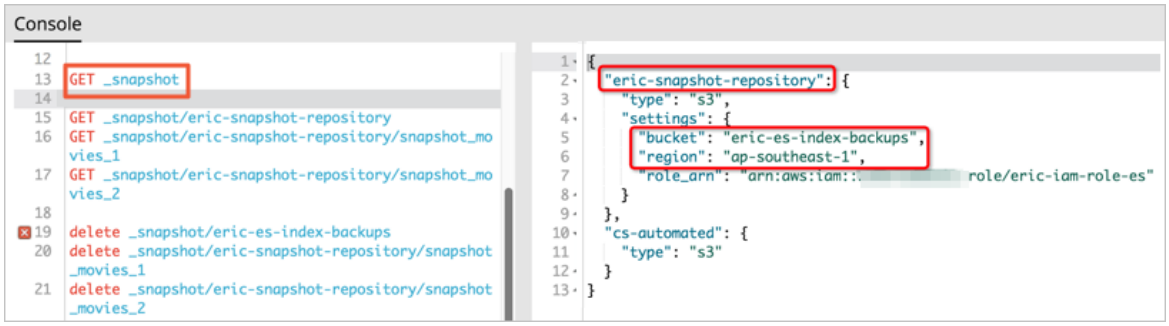
```
# wget https://pypi.python.org/packages/66/e7/fe1db6a5ed53831b53b8a6695a8f134a58833cadb5f2740802bc3730ac15/boto-2.48.0.tar.gz#md5=ce4589dd9c1d7f5d347363223ae1b970
# tar zxvf boto-2.48.0.tar.gz
# cd boto-2.48.0
# python setup.py install
```

4. Run the Python client to register the snapshot repository.

```
# python snapshot.py
```

5. Log on to the Kibana console of your AWS ES domain. In the left-side navigation pane, click **Dev Tools**. On the **Console** tab of the page that appears, run the following command to view the registration result:

```
GET _snapshot
```



Step 2: Create the first snapshot and restore data from the snapshot

1. Create a snapshot in your Amazon ES domain.

   > **Note** You can run the following commands in the Kibana console or by using curl commands in the Linux or Mac OS X command line interface (CLI).

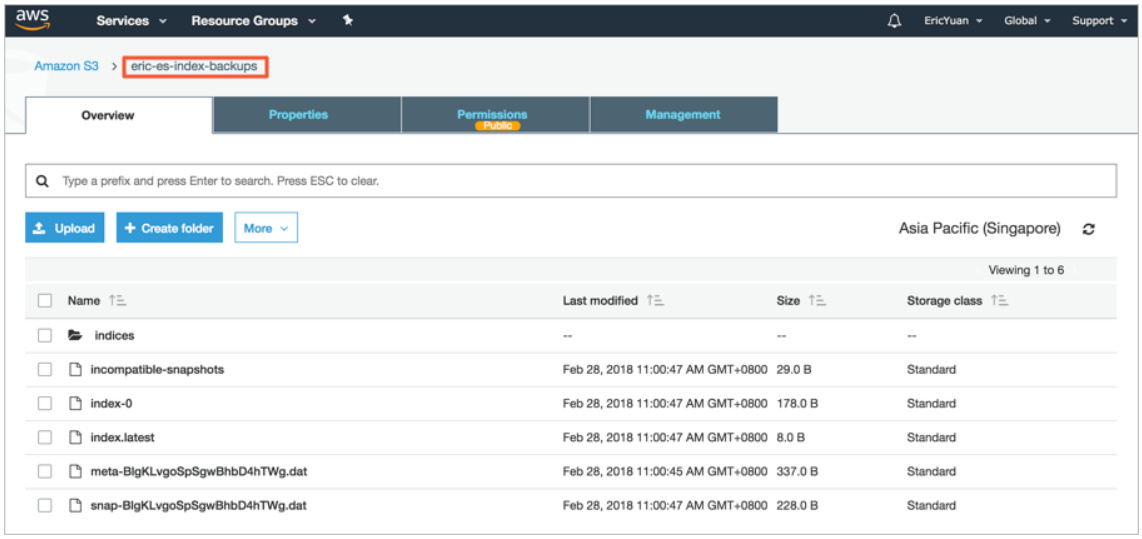
   - Create a snapshot named `snapshot_movies_1` for the `movies` index in the `eric-snapshot-repository` snapshot repository.

     ```
     PUT _snapshot/eric-snapshot-repository/snapshot_movies_1
     {
     "indexes": "movies"
     }
     ```

   - View snapshot status.

     ```
     GET _snapshot/ eric-snapshot-repository/snapshot_movies_1
     ```

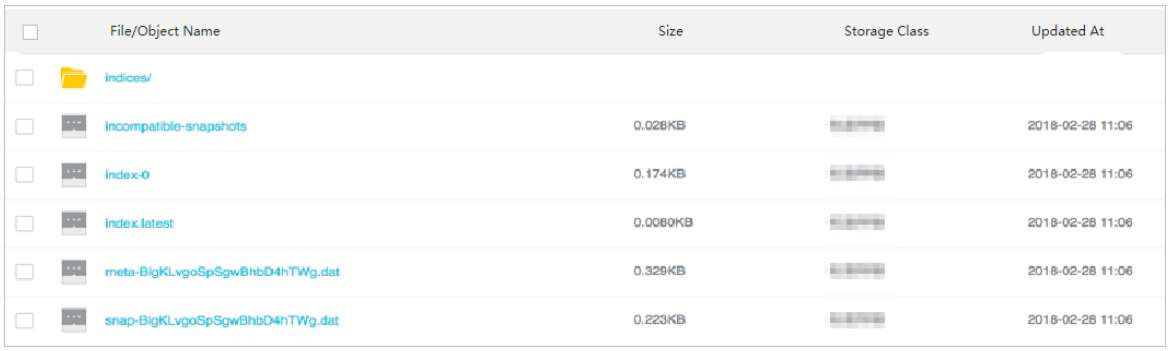
   - In the S3 console, view snapshot objects.



2. Transfer the created snapshot from your S3 bucket to your OSS bucket.

   For more information, see Seamlessly migrate data from Amazon S3 to Alibaba Cloud OSS.

   After the snapshot is transferred, view the snapshot in the OSS console.



3. Restore data from the snapshot to your Alibaba Cloud Elasticsearch cluster.

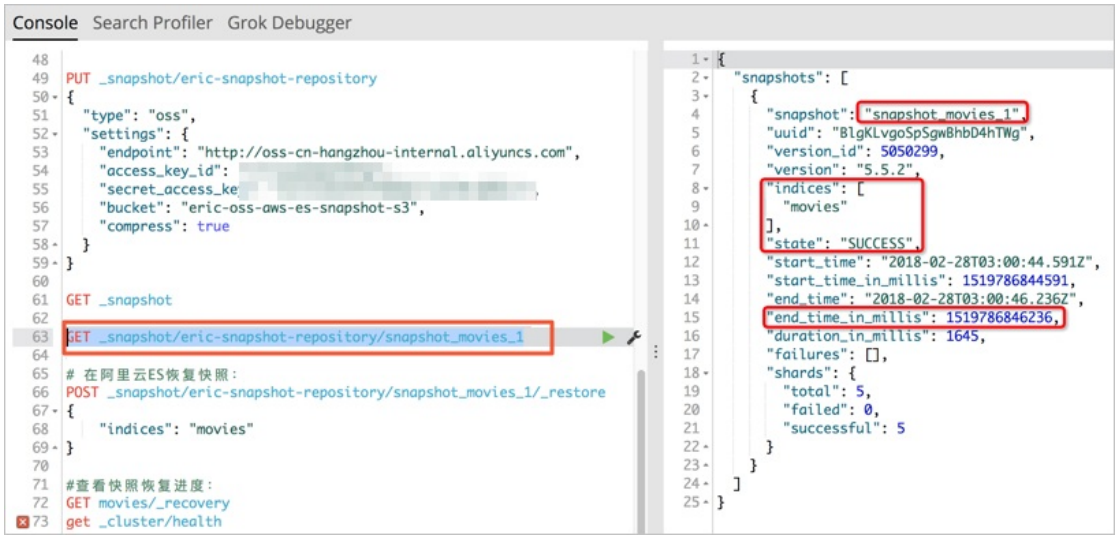
   i. Create a snapshot repository.

   Log on to the Kibana console of your Elasticsearch cluster. For more information, see Log on to the Kibana console. Then, in the left-side navigation pane, click **Dev Tools**. On the **Console** tab of the page that appears, run the following command to create a snapshot repository. The name of the snapshot repository must be the same as that of the snapshot repository registered with Amazon ES.

```
PUT _snapshot/eric-snapshot-repository
{
"type": "oss",
"settings": {
            "endpoint": "http://oss-cn-hangzhou-internal.aliyuncs.com",
            "access_key_id": "your AccessKeyID",
            "secret_access_key": "your AccessKeySecret ",
            "bucket": "eric-oss-aws-es-snapshot-s3",
            "compress": true
      }
}
```

ii. View the status of the snapshot named `snapshot_movies_1`.

```
GET _snapshot/eric-snapshot-repository/snapshot_movies_1
```



Note Take note of the start time and end time of the snapshot creation operation. This record is used when you use ossimport to migrate data in incremental snapshots. Example:

- "start_time_in_millis": 1519786844591
- "end_time_in_millis": 1519786846236

4. Restore data from the snapshot.

   Run the following command to check the availability of the `movies` index:

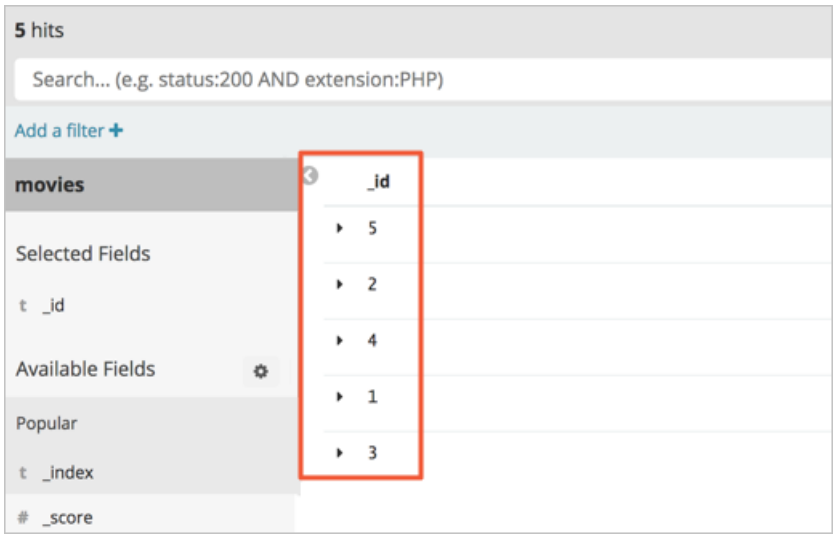
```
POST _snapshot/eric-snapshot-repository/snapshot_movies_1/_restore
{
    "indexes": "movies"
}
GET movies/_recovery
```

After the command is successfully executed, you can view three sets of data in the `movies` index. In addition, the data is the same as that in the Amazon ES domain.

Step 3: Create the final snapshot and restore data from the snapshot

1. Insert data into the `movies` index in your Amazon ES domain.

   The `movies` index contains three sets of data. Insert another two sets of data.

   You can run the `GET movies/_count` command to view the data volume of the index.

2. Create a snapshot.

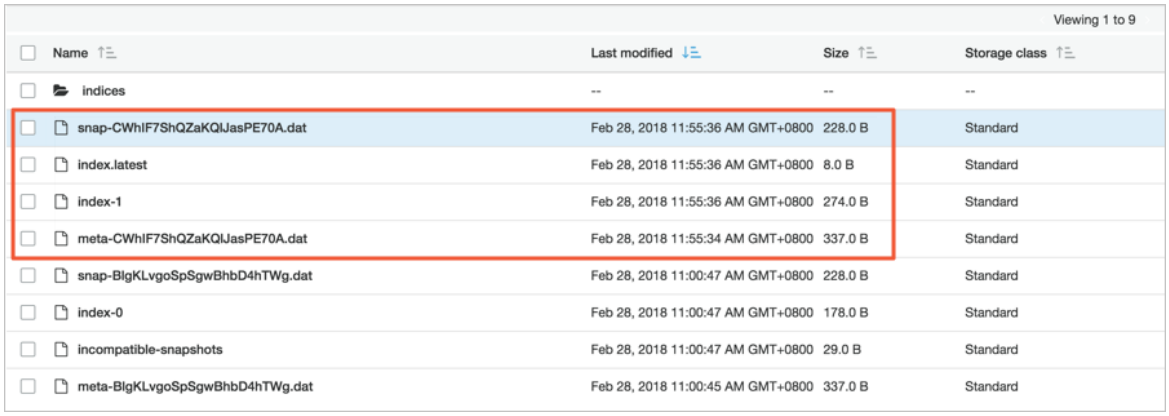
   Run the following command to create a snapshot. For more information, see Create a snapshot in your Amazon ES domain.

   ```
   PUT _snapshot/eric-snapshot-repository/snapshot_movies_2
   {
   "indices": "movies"
   }
   ```

   After the snapshot is created, run the following command to view the status of the snapshot:

   ```
   GET _snapshot/eric-snapshot-repository/snapshot_movies_2
   ```

   View objects in your S3 bucket.

3. Transfer the snapshot from your S3 bucket to your OSS bucket.

You can use ossimport to transfer the snapshot. The S3 bucket stores two snapshot objects. You can change the value of the `isSkipExistFile` variable in the *local_job.cfg* file to migrate the incremental snapshot object.

The `isSkipExistFile` variable indicates whether existing objects are skipped during data migration. The value of this variable is of the Boolean type. The default value is false. If you set the value to true, objects are skipped based on the `size` and `LastModifiedTime` settings. If you set the value to false, existing objects are overwritten. If `jobType` is set to `audit`, this variable is invalid.

Then, you can view the incremental snapshot object in the OSS bucket.

4. Restore data from the incremental snapshot.

   For more information, see the "Step 2: Create the first snapshot and restore data from the snapshot" section. Before you restore data, you must disable the `movies` index. After the restoration, you can enable the index.

   - Disable the `movies` index

     ```
     POST /movies/_close
     ```

   - View the status of the `movies` index

     ```
     GET movies/_stats
     ```

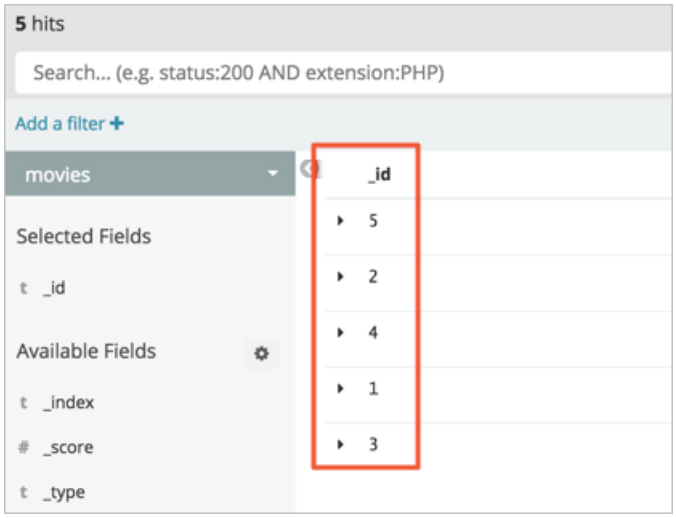
   - Restore data from the snapshot

     ```
     POST _snapshot/eric-snapshot-repository/snapshot_movies_2/_restore
     {
         "indexes": "movies"
     }
     ```

   - Enable the `movies` index

     ```
     POST /movies/_open
     ```

   After data is restored from the snapshot, the number of documents in the `movies` index of your Elasticsearch cluster is `5`. This number is the same as that in the index of your Amazon ES domain.



Summary

You can use the snapshot and restore feature to migrate data from an Amazon ES domain to an Alibaba Cloud Elasticsearch cluster. This feature requires that you disable the index whose data you want to migrate to avoid requests and write operations during the migration.

References:

- Open source Elasticsearch documentation
- Snapshot module
- Working with Amazon Elasticsearch Service Index Snapshots
- Seamlessly migrate data from Amazon S3 to Alibaba Cloud OSS
- ossimport description and configuration

3.Migrate and synchronize MySQL data

3.1. RDS synchronization

3.1.1. Select a synchronization method

If you encounter slow queries when you use an ApsaraDB RDS database, you can synchronize data from the database to an Alibaba Cloud Elasticsearch cluster for data queries and analytics. Alibaba Cloud Elasticsearch is a Lucene-based, distributed search and analytics engine. It allows you to store, query, and analyze large amounts of datasets in near real time. You can use Data Transmission Service (DTS), Logstash, DataWorks, or Canal to synchronize data from an ApsaraDB RDS database to an Alibaba Cloud Elasticsearch cluster. This topic describes the use scenarios of each method. You can select an appropriate method based on your business requirements.

Method	Description	Use scenario	Usage note	References
Use DTS to synchronize data in real time	DTS uses binary logs to synchronize data. You can use DTS to synchronize data within milliseconds, without affecting the source database.	You require a high real-time performance for data synchronization.	• DTS uses the read and write resources of the source database and destination cluster during data initialization. This may increase the loads of the database and cluster. • You can customize mappings for indexes. However, you must make sure that the fields defined in the mappings are the same as those in the source database. • You must purchase a data synchronization instance in the DTS console. For more information about how to purchase such an instance, see Purchase procedure. For more information about the pricing of DTS, see Pricing.	Use DTS to synchronize MySQL data to an Alibaba Cloud Elasticsearch cluster in real time

Method	Description	Use scenario	Usage note	References
Use the logstash-input-jdbc plug-in to synchronize data	You can use the logstash-input-jdbc plug-in to query the data in an ApsaraDB RDS database and migrate the data to an Elasticsearch cluster. During data synchronization, the plug-in uses a round-robin method to identify the latest inserted or updated data in the database on a regular basis. Then, it queries all identified data at a time and migrates the data to an Elasticsearch cluster. The logstash-input-jdbc plug-in provides lower real-time performance than DTS. Data is synchronized within seconds.	• You want to synchronize full data and can accept a latency of a few seconds. • You want to query specific data at a time and synchronize the data.	• Before you use this method, upload an SQL JDBC driver that is compatible with the version of the ApsaraDB RDS database. • You must add the IP addresses of the nodes in your Logstash cluster to the whitelist of your ApsaraDB RDS instance. • Your Logstash cluster and ApsaraDB RDS instance must reside in the same zone. This avoids inconsistent timestamps during data synchronization. • You must make sure that the `_id` field in your Elasticsearch cluster is the same as the `id` field in the ApsaraDB RDS database. • When you insert or update data in your ApsaraDB RDS database, make sure that the related record contains a field that indicates the insertion or update time.	None

Method	Description	Use scenario	Usage note	References
Use DataWorks to synchronize offline data	DataWorks is a comprehensive service that provides modules such as Data Integration, DataStudio, and Data Quality. You can use DataWorks to import and store structured data, convert and develop the data, and then synchronize the processed data to Elasticsearch clusters or other data systems.	• You want to synchronize offline big data. DataWorks can collect offline data at a minimum interval of 5 minutes. • You want to customize query statements, perform joint queries on multiple tables, and then synchronize data.	• You must activate the DataWorks service. • If a high transmission speed is required or the environment is complex, you must customize resource groups. • You must add the IP addresses of the resource groups to the whitelist of your ApsaraDB RDS instance.	Use DataWorks to synchronize data from a MySQL database to an Alibaba Cloud Elasticsearch cluster
Use Canal to synchronize MySQL data	You can use binary logs to synchronize and subscribe to data in real time.	You require a high real-time performance for data synchronization.	• You must build a Canal environment on an Elastic Compute Service (ECS) instance. However, this increases the costs of data synchronization. • Canal V1.1.4 cannot be used to synchronize data to an Elasticsearch V7.X cluster. We recommend that you use Logstash or DTS to synchronize MySQL data to an Elasticsearch V7.X cluster. • You can customize mappings for indexes. However, you must make sure that the fields defined in the mappings are the same as those in the source database.	Use Canal to synchronize data to an Alibaba Cloud Elasticsearch cluster

3.1.2. Use Logstash to synchronize data from ApsaraDB RDS for MySQL to Alibaba Cloud Elasticsearch

Use Logstash to synchronize data from an ApsaraDB RDS for MySQL database to an Alibaba Cloud Elasticsearch cluster

If you want to synchronize data from an ApsaraDB RDS for MySQL database to an Alibaba Cloud Elasticsearch cluster, you can use the logstash-input-jdbc plug-in and the pipeline configuration feature provided by Alibaba Cloud Logstash. logstash-input-jdbc is a built-in plug-in of Alibaba Cloud Logstash and cannot be removed. You can use this method to synchronize full or incremental data from an ApsaraDB RDS for MySQL database to an Alibaba Cloud Elasticsearch cluster. This topic describes the procedure in detail.

Background information

A lot of service providers deploy Elasticsearch on top of relational databases. The service providers may need to make sure that data in an Elasticsearch cluster is automatically synchronized from the relational database with which the Elasticsearch cluster is associated. In this case, the service providers can use Logstash to synchronize data by referring to the operations described in this topic. For more information, see Use Logstash and JDBC to synchronize data from a relational database to an Elasticsearch cluster.

Limits

The synchronization of data from an ApsaraDB RDS for MySQL database to an Alibaba Cloud Elasticsearch cluster by using the logstash-input-jdbc plug-in is achieved based on the following logic: The plug-in uses a round-robin method to identify the latest inserted or updated data records in the ApsaraDB RDS for MySQL database on a regular basis and synchronizes the data records to the Elasticsearch cluster. To ensure that a data synchronization task can run as expected, the Elasticsearch cluster and the ApsaraDB RDS for MySQL database must meet the following conditions:

- The values of the _id field in the Elasticsearch cluster must be the same as those of the id field in the ApsaraDB RDS for MySQL database.

  This condition ensures that the synchronization task can establish a mapping between data records in the ApsaraDB RDS for MySQL database and documents in the Elasticsearch cluster. If you update a data record in the ApsaraDB RDS for MySQL database, the synchronization task uses the updated data record to overwrite the document that has the same ID in the Elasticsearch cluster.

  > **Note** In essence, an update operation in Elasticsearch deletes the original document and indexes the new document. Therefore, the overwrite operation is as efficient as an update operation performed by the synchronization task.

- If you insert a data record to or update a data record in the ApsaraDB RDS for MySQL database, the data record must contain a field that indicates the time when the data record is inserted or updated.

Each time the logstash-input-jdbc plug-in performs a round robin, the plug-in records the time when the last data record in the round robin is inserted to or updated in the ApsaraDB RDS for MySQL database. Logstash synchronizes only data records that meet the following requirements from the ApsaraDB RDS for MySQL database: The time when the data records are inserted to or updated in the ApsaraDB RDS for MySQL database is later than the time when the last data record in the previous round robin is inserted or updated in the ApsaraDB RDS for MySQL database.

> **Notice** If you delete data records in the ApsaraDB RDS for MySQL database, the logstash-input-jdbc plug-in cannot delete the documents that have the same IDs in the Elasticsearch cluster. To delete the documents in the Elasticsearch cluster, you must run the related command on the Elasticsearch cluster.

- The ApsaraDB RDS for MySQL database and the Elasticsearch cluster must reside in the same time zone. If they do not reside in the same time zone, the time-related data may have a time zone offset after the synchronization.

Procedure

1. Step 1: Make preparations
2. Step 2: Configure a Logstash pipeline
3. Step 3: Verify the result

Step 1: Make preparations

1. Create an Alibaba Cloud Elasticsearch cluster and enable the Auto Indexing feature for the cluster.

   For more information, see Create an Alibaba Cloud Elasticsearch cluster and Access and configure an Elasticsearch cluster. In this example, an Elasticsearch V7.10.0 cluster is used.

2. Create an Alibaba Cloud Logstash cluster and upload an SQL JDBC driver that is compatible with the version of your ApsaraDB RDS for MySQL instance. In this example, the driver file mysql-connector-java-5.1.48.jar is used.

   The Logstash cluster must reside in the same virtual private cloud (VPC) as the Elasticsearch cluster, and the version of the Logstash cluster must be the same as that of the Elasticsearch cluster. For more information about how to create a Logstash cluster and upload a driver file for the Logstash cluster, see Step 1: Create a Logstash cluster and Configure third-party libraries.

   > **Note** You can also use Logstash to synchronize data from an ApsaraDB RDS for MySQL instance that is deployed on the Internet. Before you perform the operation, you must configure a Source Network Address Translation (SNAT) entry for the Logstash cluster, enable the public IP address of the ApsaraDB RDS for MySQL instance, and add the IP addresses of the nodes in the Logstash cluster to the whitelist of the ApsaraDB RDS for MySQL instance. For more information about how to configure an SNAT entry, see Configure a NAT gateway for data transmission over the Internet.

3. Prepare test data and add the IP addresses of the nodes in the Logstash cluster to the whitelist of the ApsaraDB RDS for MySQL instance. You can obtain the IP addresses on the Basic Information page of the Logstash cluster.

   For more information about how to configure an IP address whitelist for an ApsaraDB RDS for MySQL instance, see Use a database client or the CLI to connect to an ApsaraDB RDS for MySQL instance.

In this example, the following statement is used to create a table in the ApsaraDB RDS for MySQL database:

```
CREATE table food(
id int PRIMARY key AUTO_INCREMENT,
name VARCHAR (32),
insert_time DATETIME,
update_time DATETIME );
```

The following statements are used to insert data into the table:

```
INSERT INTO food values(null,'Chocolates',now(),now());
INSERT INTO food values(null,'Yogurt',now(),now());
INSERT INTO food values(null,'Ham sausages',now(),now());
```

Step 2: Configure a Logstash pipeline

1.
2.
3.
4.
5. On the **Create Task** page, configure **Pipeline ID** and Config Settings.

   In this topic, the following configurations are entered in the Config Settings field:

```
input {
  jdbc {
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_driver_library => "/ssd/1/share/<Logstash cluster ID>/logstash/current/config/custom/mysql-connector-java-5.1.48.jar"
    jdbc_connection_string => "jdbc:mysql://rm-bp1xxxxx.mysql.rds.aliyuncs.com:3306/<Name of the ApsaraDB RDS for MySQL database>?useUnicode=true&characterEncoding=utf-8&useSSL=false&allowLoadLocalInfile=false&autoDeserialize=false"
    jdbc_user => "xxxxx"
    jdbc_password => "xxxx"
    jdbc_paging_enabled => "true"
    jdbc_page_size => "50000"
    statement => "select * from food where update_time >= :sql_last_value"
    schedule => "* * * * *"
    record_last_run => true
    last_run_metadata_path => "/ssd/1/<Logstash cluster ID>/logstash/data/last_run_metadata_update_time.txt"
    clean_run => false
    tracking_column_type => "timestamp"
    use_column_value => true
    tracking_column => "update_time"
  }
}
filter {
}
output {
 elasticsearch {
    hosts => "http://es-cn-0h****dd0hcbnl.elasticsearch.aliyuncs.com:9200"
    index => "rds_es_dxhtest_datetime"
    user => "elastic"
    password => "xxxxxxx"
    document_id => "%{id}"
  }
}
```

Note You must replace the `Logstash cluster ID` in the preceding code with the ID of the Logstash cluster that you use. For more information about how to obtain the ID of a Logstash cluster, see View the basic information of a cluster.

Description of the configurations in the Config Settings field

Part	Description
input	Specifies the input data source. For more information about the supported data source types, see Input plugins. In this example, an input data source that is connected by using JDBC is used. For more information about the related parameters, see Parameters in the input part.
filter	Specifies the plug-in that is used to preprocess input data. For more information about the supported plug-ins, see Filter plugins.

Part	Description
output	Specifies the output data source. For more information about the supported data source types, see Output plugins. In this example, data in an ApsaraDB RDS for MySQL database is synchronized to an Elasticsearch cluster. Therefore, the information of the Elasticsearch cluster is configured in the output part. For more information about the related parameters, see Step 2: Create and run a Logstash pipeline. **Notice** If the file_extend parameter is specified in the output configuration of a pipeline, you must make sure that the logstash-output-file_extend plug-in is installed for the Logstash cluster. For more information, see Install a Logstash plug-in.

Parameters in the input part

Parameter	Description
jdbc_driver_class	The class of the JDBC driver.
jdbc_driver_library	The driver file that is used for the JDBC-based connection to the ApsaraDB RDS for MySQL database. Configure this parameter in the /ssd/1/share/<Logstash cluster ID>/logstash/current/config/custom/<Name of the driver file> format. You must upload the desired driver file in the Elasticsearch console in advance. For more information about the driver files that are supported by Logstash and how to upload a driver file, see Configure third-party libraries.

Parameter	Description
jdbc_connection_string	The connection string that is used to connect to the ApsaraDB RDS for MySQL database. The connection string contains the endpoint and port number of the related ApsaraDB RDS for MySQL instance, and the name of the ApsaraDB RDS for MySQL database. Configure this parameter in the following format: jdbc:mysql://<Endpoint of the ApsaraDB RDS for MySQL instance>:<Port number>/<Name of the ApsaraDB RDS for MySQL database>?useUnicode=true&characterEncoding=utf-8&useSSL=false&allowLoadLocalInfile=false&autoDeserialize=false. **Notice** ◦ <Endpoint of the ApsaraDB RDS for MySQL instance>: The internal endpoint of the ApsaraDB RDS for MySQL instance must be used. If you use the public endpoint of the ApsaraDB RDS for MySQL instance, you must configure a Network Address Translation (NAT) gateway for the Logstash cluster to enable the Logstash cluster to connect to the Internet. For more information, see Configure a NAT gateway for data transmission over the Internet. ◦ Port number: The port number must be the same as the port number of the outbound traffic of the ApsaraDB RDS for MySQL instance. In most cases, the port number is 3306.
jdbc_user	The username of the ApsaraDB RDS for MySQL database.
jdbc_password	The password of the ApsaraDB RDS for MySQL database.
jdbc_paging_enabled	Specifies whether to enable paging. Default value: false.
jdbc_page_size	The number of entries to return on each page.
statement	The SQL statement that is used to query data from the ApsaraDB RDS for MySQL database. If you want to query data from multiple tables in the ApsaraDB RDS for MySQL database, you can use a JOIN statement. **Note** The value of sql_last_value is used to calculate the rows to query. By default, this parameter is set to Thursday, 1 January 1970. For more information, see Jdbc input plugin.
schedule	The interval at which the SQL statement is executed. The value "* * * * *" indicates that the SQL statement is executed every minute. Set this parameter to a cron expression that is supported by Rufus.

Parameter	Description
record_last_run	Specifies whether to record the last execution result. If this parameter is set to true, the value of tracking_column in the last execution result is stored in the file in the path specified by using the last_run_metadata_path parameter.
last_run_metadata_path	The path where the file that contains the last execution time is stored. A path in which you can store the file is provided at the backend. The path is in the `/ssd/1/<Logstash cluster ID>/logstash/data/` format. **Note** We recommend that you configure this parameter in the `/ssd/1/<Logstash cluster ID>/logstash/data/` format when you configure a Logstash pipeline. If you configure this parameter in another format, the condition records for synchronization cannot be stored in the file in the path specified by using the last_run_metadata_path parameter. The storage failure is due to insufficient permissions.
clean_run	Specifies whether to clear the path that is specified by using the last_run_metadata_path parameter. Default value: false. If this parameter is set to true, each query starts from the first entry in the database.
use_column_value	Specifies whether to record the values of a specific column. If this parameter is set to true, the system records the latest value of the column that is specified by using tracking_column and determines the records that need to be updated in the file based on the value of tracking_column when the SQL statement runs for the next time.
tracking_column_type	The type of the column whose values you want to track. Default value: numeric.
tracking_column	The column whose values you want to track. The values must be sorted in ascending order. In most cases, this column is the primary key.

> **Notice**
>
> - The preceding configurations are based on test data. You can configure the pipeline based on your business requirements. For more information about other parameters supported by the input plug-in, see Logstash Jdbc input plugin.
> - If your configurations contain a parameter similar to last_run_metadata_path, the file path must be provided by Alibaba Cloud Logstash. A path in the `/ssd/1/<Logstash cluster ID>/logstash/data/` format is provided at the backend and is available for tests. The system does not delete the data in this path. Make sure that your disk has sufficient storage space when you use this path.
> - For security purposes, if you use a JDBC driver to configure a pipeline, you must add allowLoadLocalInfile=false&autoDeserialize=false at the end of the jdbc_connection_string parameter, such as `jdbc_connection_string => "jdbc:mysql://xxx.drds.aliyuncs.com:3306/<Database name>?allowLoadLocalInfile=false&autoDeserialize=false"`. Otherwise, the system displays an error message that indicates a check failure.

For more information about how to configure parameters in the Config Settings field, see Logstash configuration files.

6.
7.

Step 3: Verify the result

1.
2.
3. On the **Console** tab of the page that appears, run the following command to view the number of indexes that store synchronized data:

```
GET rds_es_dxhtest_datetime/_count
{
  "query": {"match_all": {}}
}
```

If the command is successfully run, the following result is returned:

```
{
  "count" : 3,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  }
}
```

4. Update the data in the MySQL table and insert data into the table.

```
UPDATE food SET name='Chocolates',update_time=now() where id = 1;
INSERT INTO food values(null,'Egg',now(),now());
```

5. In the Kibana console, view the updated and inserted data.
   - Query the data record in which the value of name is Chocolates.

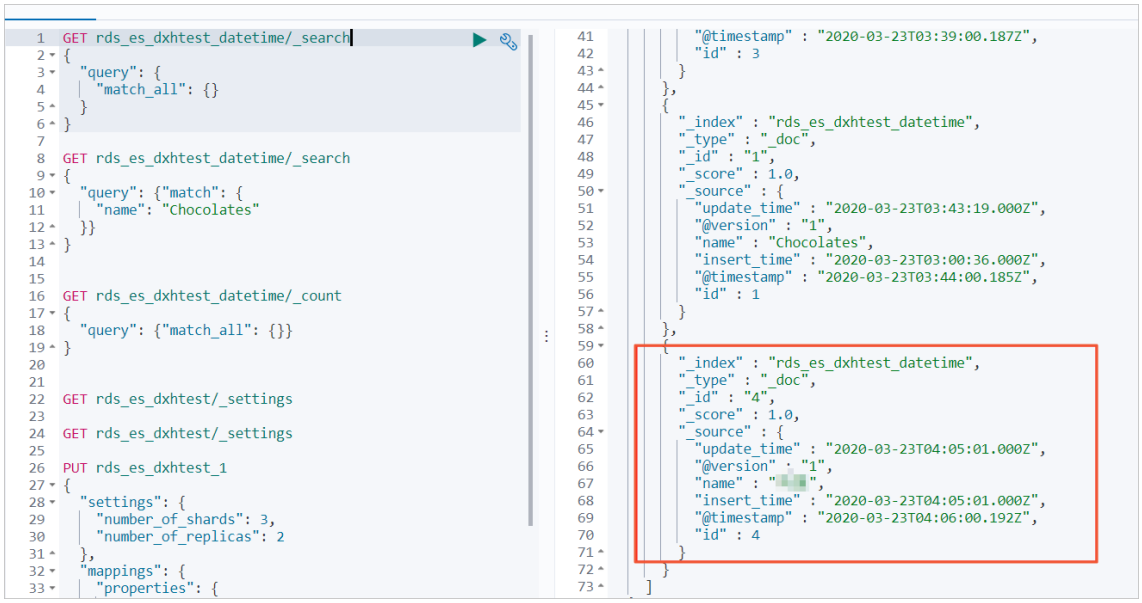
```
GET rds_es_dxhtest_datetime/_search
{
  "query": {
    "match": {
      "name": "Chocolates"
    }}
}
```

If the command is successfully run, the following result is returned:

   - Query all data.

```
GET rds_es_dxhtest_datetime/_search
{
  "query": {
    "match_all": {}
  }
}
```

If the command is successfully run, the following result is returned:



FAQ

Q: What do I do if my synchronization task fails because the pipeline is stuck in the initializing state, data before and after synchronization is inconsistent, or the connection to the database fails?

A: Check whether the cluster logs of your Logstash cluster contain error information and identify the cause based on the error information. For more information, see Query logs. The following table describes common causes of errors and solutions to the errors.

> **Note** If an update operation is being performed on your Logstash cluster when you perform the operations described in the following solutions, pause the update operation by referring to View the progress of a cluster task. After the operations described in the solutions are complete, the system restarts the Logstash cluster and resumes the update operation.

Cause	Solution
The IP addresses of nodes in the Logstash cluster are not added to the whitelist of the ApsaraDB RDS for MySQL instance.	Add the IP addresses of nodes in the Logstash cluster to the whitelist of the ApsaraDB RDS for MySQL instance. For more information, see Use a database client or the CLI to connect to an ApsaraDB RDS for MySQL instance. **Note** For more information about how to obtain the IP addresses of the nodes in a Logstash cluster, see View the basic information of a cluster.

Cause	Solution
You use the Logstash cluster to synchronize data from a self-managed MySQL database that is hosted on an ECS instance, but the private IP addresses and internal ports of nodes in the cluster are not added to a security group of the ECS instance.	Add the private IP addresses and internal ports of nodes in the cluster to a security group of the ECS instance. For more information, see Add a security group rule. **Note** For more information about how to obtain the IP address and port of a node in a Logstash cluster, see View the basic information of a cluster.
The Elasticsearch cluster does not reside in the same VPC as the Logstash cluster.	Use one of the following solutions: • Purchase an Elasticsearch cluster that resides in the same VPC as the Logstash cluster. For more information, see Create an Alibaba Cloud Elasticsearch cluster. After the Elasticsearch cluster is purchased, modify the pipeline configuration of the Logstash cluster. • Configure a NAT gateway to transmit data over the Internet. For more information, see Configure a NAT gateway for data transmission over the Internet.
The endpoint of the ApsaraDB RDS for MySQL instance is incorrect, and the port number of the instance is not 3306.	Obtain the correct endpoint and port number. For more information, see View and change the internal and public endpoints and port numbers of an ApsaraDB RDS for MySQL instance. Replace the endpoint and port number in the value of the jdbc_connection_string parameter with the endpoint and port number that you obtained. **Notice** <Endpoint of the ApsaraDB RDS for MySQL instance>: The internal endpoint of the ApsaraDB RDS for MySQL instance must be used. If you use the public endpoint of the ApsaraDB RDS for MySQL instance, you must configure a Network Address Translation (NAT) gateway for the Logstash cluster to enable the Logstash cluster to connect to the Internet. For more information, see Configure a NAT gateway for data transmission over the Internet.
The Auto Indexing feature is disabled for the Elasticsearch cluster.	Enable the Auto Indexing feature for the Elasticsearch cluster. For more information, see Configure the YML file.

Cause	Solution
The load of the Elasticsearch or Logstash cluster is excessively high.	Upgrade the configuration of the Elasticsearch or Logstash cluster. For more information, see Upgrade the configuration of a cluster. **Note** If the load of the Elasticsearch cluster is excessively high, you can view the monitoring data collected based on metrics in the Elasticsearch console to obtain the load information of the Elasticsearch cluster. For more information, see View cluster monitoring data. If the load of the Logstash cluster is excessively high, you can enable the X-Pack Monitoring feature for the Logstash cluster, use the feature to monitor the Logstash cluster, and then view the monitoring data. For more information, see Enable the X-Pack Monitoring feature.
No driver file that is used for JDBC-based connection to the ApsaraDB RDS for MySQL database is uploaded.	Upload a driver file. For more information, see Configure third-party libraries.
The file_extend parameter is specified in the configuration of the pipeline. However, the logstash-output-file_extend plug-in is not installed.	Use one of the following solutions: • Install the logstash-output-file_extend plug-in. For more information, see Install a Logstash plug-in. • Remove the file_extend parameter from the configuration of the pipeline.

For more information about the cause of and solution to this issue, see FAQ about data transfer by using Logstash.

3.1.3. Use DataWorks to synchronize data from a MySQL database to an Alibaba Cloud Elasticsearch cluster

Synchronize data from MySQL to Alibaba Cloud Elasticsearch

Alibaba Cloud provides a variety of cloud storage and database services. If you want to search for and analyze data stored in these services, you can use the Data Integration service provided by DataWorks to collect the offline data at a minimum interval of 5 minutes and synchronize the collected data to Alibaba Cloud Elasticsearch. In this topic, data is synchronized from an ApsaraDB RDS for MySQL database to an Alibaba Cloud Elasticsearch cluster.

Procedure

1. Preparations

   Prepare a MySQL data source and create a DataWorks workspace and an Alibaba Cloud Elasticsearch cluster. Configure the Elasticsearch cluster.
2. Step 1: Purchase and create an exclusive resource group
3. Step 2: Add data sources

Connect the MySQL data source and Elasticsearch cluster to the Data Integration service provided by DataWorks.

4. Step 3: Create and run a data synchronization node

   Use the codeless user interface (UI) to create a node to synchronize data from the MySQL data source to the Elasticsearch cluster and configure the node. Select the exclusive resource group that you created when you configure the node. The data synchronization node runs on the selected exclusive resource group for Data Integration and writes data to the Elasticsearch cluster.

5. Step 4: View the synchronized data

   View the synchronized data in the Kibana console of the Elasticsearch cluster.

Preparations

1. Create a database.

   You can use an ApsaraDB RDS database or create a database on your on-premises machine. In this topic, an ApsaraDB RDS for MySQL database is used. Join two MySQL tables and synchronize data in the tables to Alibaba Cloud Elasticsearch. The following figures show the two tables. For more information, see Create an ApsaraDB RDS for MySQL instance.

   Table 1

	id	stu_id	c_name	grade
1	1	901	compute	98
2	2	901	english	80
3	3	902	compute	65
4	4	902	chinese	88
5	5	903	chinese	95
6	6	904	compute	70
7	7	904	english	92
8	8	905	english	94
9	9	906	compute	90
10	10	906	english	85

   Table 2

	id	name	sex	birth	department	address
1	901	zhangda	man	1985	compute	beijing
2	902	zhanger	man	1986	chinese	beijing
3	903	zhangsan	woman	1990	chinese	hunan
4	904	lisi	man	1990	english	liaoning
5	905	wangwu	woman	1991	english	fujian
6	906	wangliu	man	1988	compute	hunan

2. Create a DataWorks workspace.

   For more information, see Create a workspace. The workspace must reside in the same region as the ApsaraDB RDS for MySQL database.

3. Create an Alibaba Cloud Elasticsearch cluster and enable the Auto Indexing feature for the cluster.

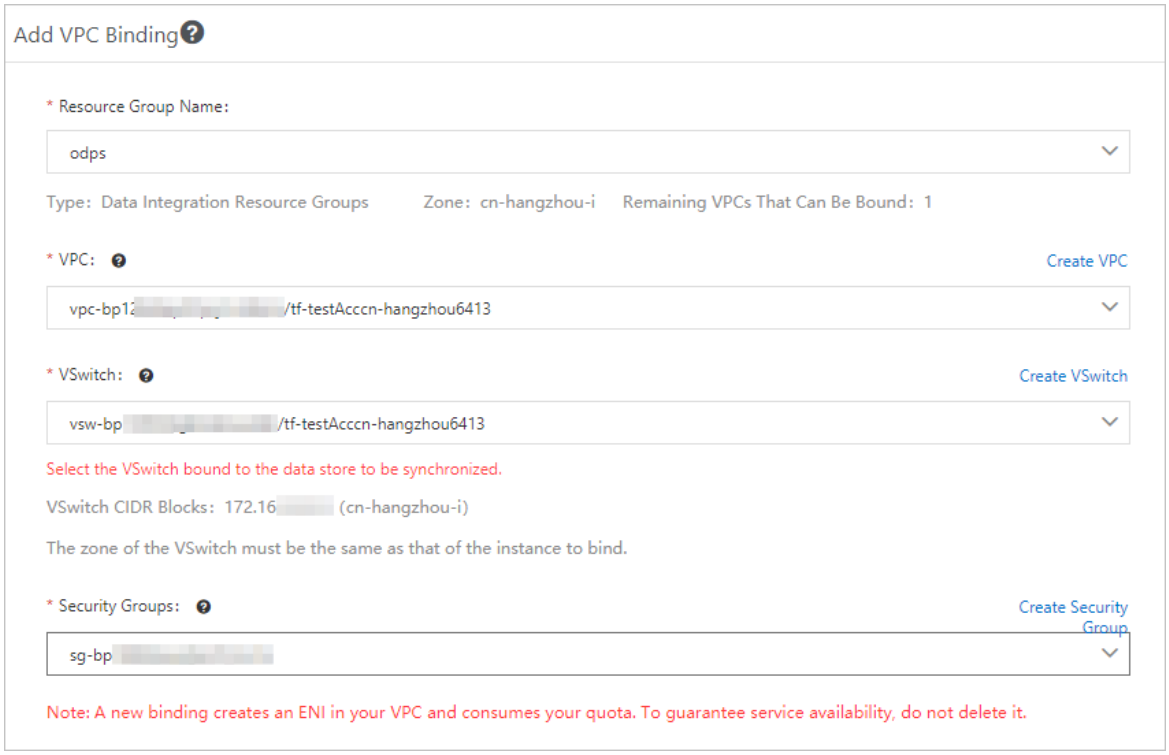
   The Elasticsearch cluster must reside in the same virtual private cloud (VPC) as the ApsaraDB RDS for MySQL database. For more information, see Create an Alibaba Cloud Elasticsearch cluster and

Configure the YML file.

Step 1: Purchase and create an exclusive resource group

1.
2.
3.
4.
5. Find the exclusive resource group that you create and click Network Settings in the Actions column. The **VPC Binding** tab appears. On the VPC Binding tab, click Add Binding to bind the exclusive resource group to a VPC. For more information, see Configure network settings.

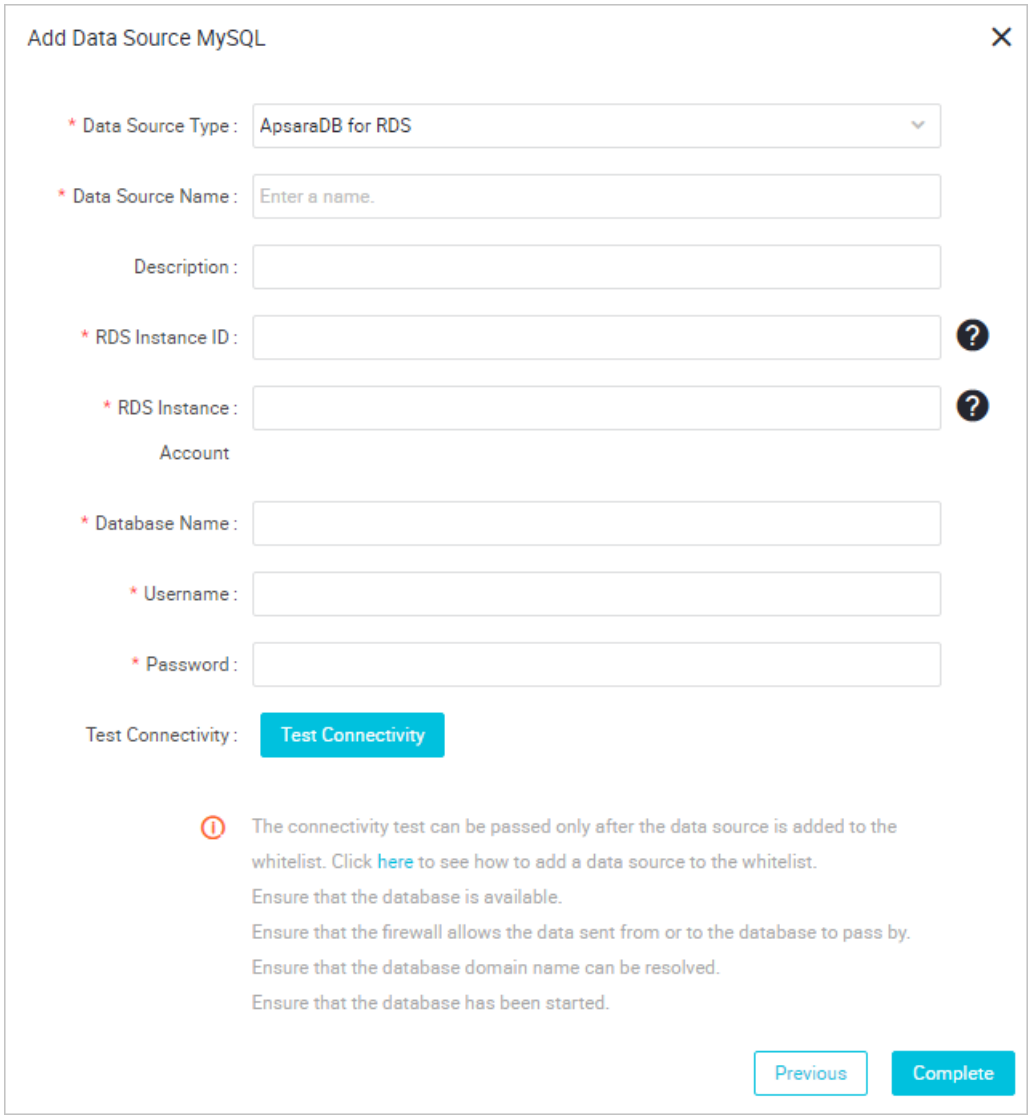
   Exclusive resources are deployed in the VPC where DataWorks resides. You can use DataWorks to synchronize data from the ApsaraDB RDS for MySQL database to the Elasticsearch cluster only after DataWorks connects to the VPCs where the database and cluster reside. In this topic, the ApsaraDB RDS for MySQL database and Elasticsearch cluster reside in the same VPC. Therefore, when you bind the exclusive resource group to a VPC, you need to select the **VPC** and **vSwitch** to which the Elasticsearch cluster belongs.



6.

Step 2: Add data sources

1.
2.
3.
4. In the Relational Database section of the **Add data source** dialog box, click **MySQL**. In the **Add MySQL data source** dialog box, configure the parameters.



Data source type: In this example, this parameter is set to **Alibaba Cloud instance mode**. You can also set this parameter to **Connection string mode**. For more information about the configurations of other parameters, see Add a MySQL data source.

> **Notice** If you set the Data source type parameter to **Connection string mode**, you can set the **JDBC URL** parameter to the public endpoint of the ApsaraDB RDS for MySQL instance. You must add the elastic IP address (EIP) of the exclusive resource group to the whitelist of the ApsaraDB RDS for MySQL instance. For more information, see Configure an IP address whitelist for an ApsaraDB RDS for MySQL instance and Add the EIP or CIDR block of an exclusive resource group for Data Integration to the whitelist of a data source.

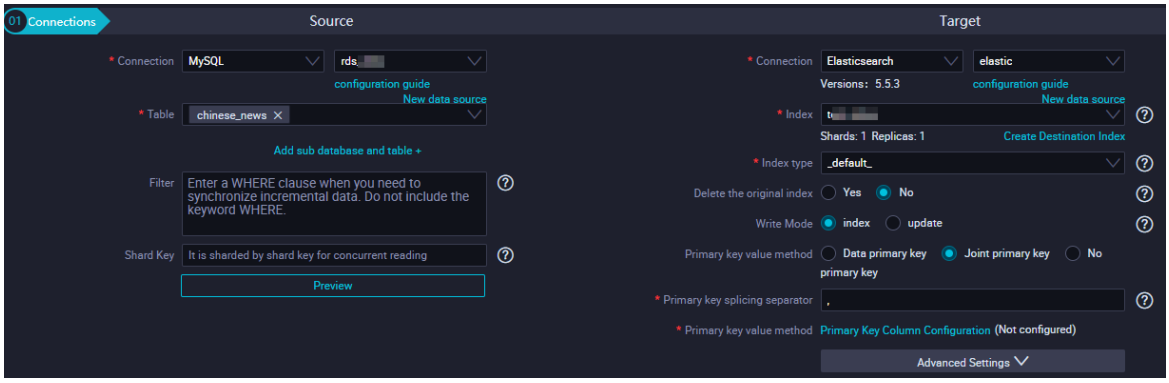
5. Click **Complete**.
6.

Step 3: Create and run a data synchronization node

1. On the DataStudio page of the DataWorks console, create a workflow.

   For more information, see Manage workflows.

2. Create a batch synchronization node.
   i. In the DataStudio pane, open the newly created workflow, right-click **Data Integration**, and then choose **Create > Batch Synchronization**.
   ii. In the **Create Node** dialog box, configure the **Node Name** parameter and click **Commit**.
3. In the **Source** section of the **Connections** step, specify the MySQL data source and the name of the table that you created. In the **Target** section, specify the Elasticsearch data source, index name, and index type.



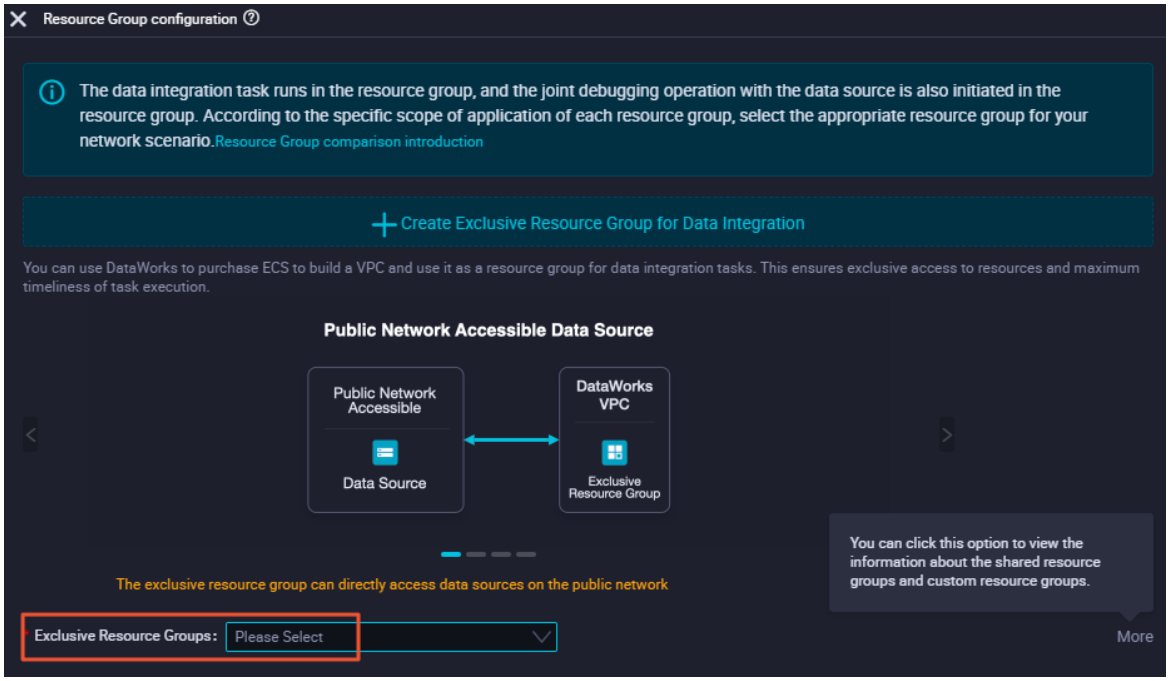
> **Note**
> - You can also use the code editor to configure the node. For more information, see Create a synchronization node by using the code editor, DRDS Reader, and Elasticsearch Writer.
> - We recommend that you set **Enable node discovery** to **No** in the **advanced settings** of the **Elasticsearch** data source. Otherwise, a connection timeout error occurs during data synchronization.

4. In the **Mappings** step, configure mappings between **source fields** and **destination fields**.
5. In the **Channel** step, configure the parameters.
6. Configure properties for the node.

   In the right-side navigation pane of the configuration tab of the node, click **Properties**. On the Properties tab, configure properties for the node. For more information about the parameters, see Basic properties.

> **Notice**
> - Before you commit a node, you must configure a **dependent ancestor node** for the node in the Dependencies section of the Properties tab. For more information, see Configure same-cycle scheduling dependencies.
> - If you want the system to periodically run a node, you must configure time properties for the node in the **Schedule** section of the Properties tab. The time properties include Validity Period, Scheduling Cycle, Run At, and Rerun.
> - The configuration of an auto triggered node takes effect at 00:00 of the next day.

7. Configure the resource group that you want to use to run the synchronization node.



i. In the right-side navigation pane of the configuration tab of the node, click the **Resource Group configuration** tab.

ii. Select the exclusive resource group that you create from the **Exclusive Resource Groups** drop-down list.

8. Commit the node.

   i. Save the current configurations and click the icon in the top toolbar.

   ii. In the **Commit Node** dialog box, enter your comments in the **Change description** field.

   iii. Click **OK**.

9. Click the icon in the top toolbar to run the node.

   You can view the operational logs of the node when the node is running. After the node is successfully run, the result shown in the following figure is returned.

```
2020-05-14 17:09:23 [INFO] Sandbox context cleanup temp file success.
2020-05-14 17:09:23 [INFO] Data synchronization ended with return code: [0].
2020-05-14 17:09:23 INFO ================================================================
2020-05-14 17:09:23 INFO Exit code of the Shell command 0
2020-05-14 17:09:23 INFO --- Invocation of Shell command completed ---
2020-05-14 17:09:23 INFO Shell run successfully!
2020-05-14 17:09:23 INFO Current task status: FINISH
2020-05-14 17:09:23 INFO Cost time is: 26.854s
```

Step 4: View the synchronized data

1. Log on to the Kibana console of the destination Elasticsearch cluster.

   For more information, see Log on to the Kibana console.

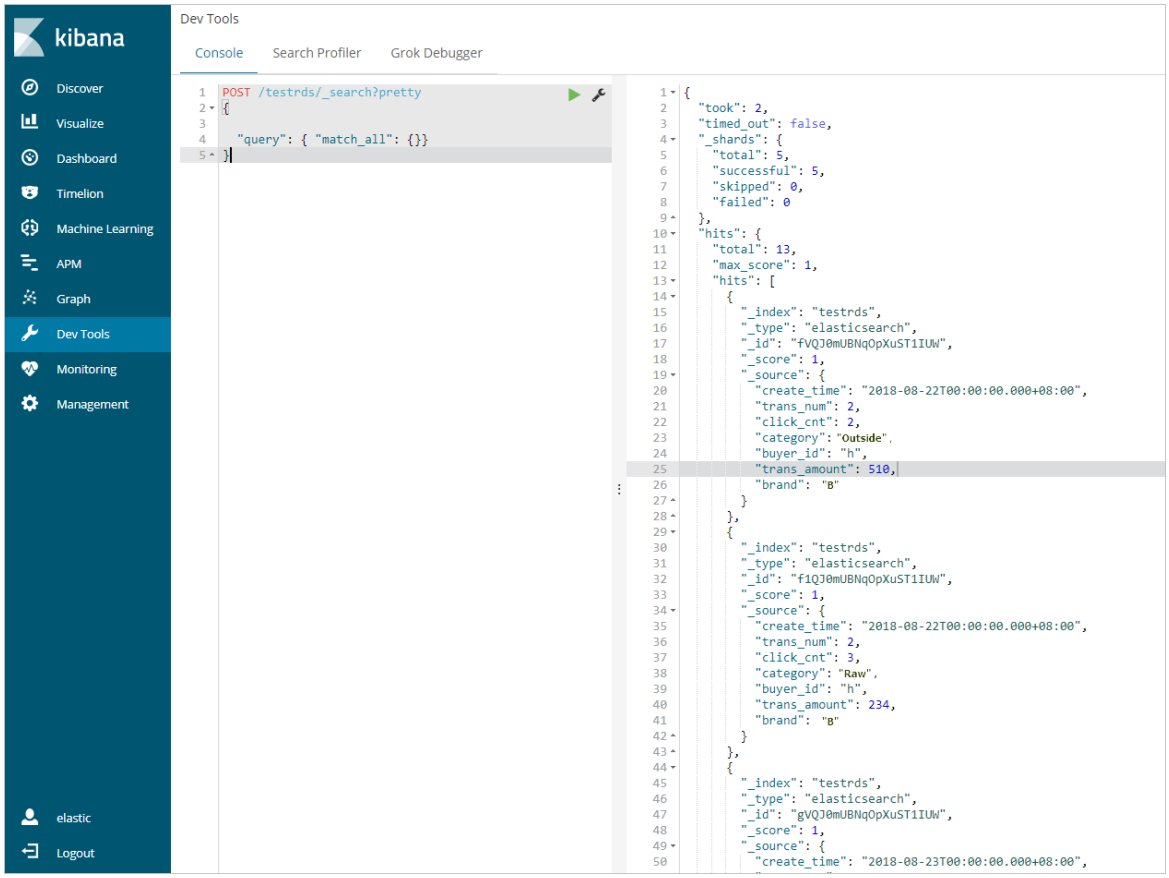
2. In the left-side navigation pane, click **Dev Tools**.

3. On the **Console** tab of the page that appears, run the following command to query the synchronized data:

```
POST /mysqljoin/_search?pretty
{
"query": { "match_all": {}}
}
```

Note `mysqljoin` is the value that you configured for the `index` field when you configure the node by using the code editor.

If the data is successfully synchronized, the result shown in the following figure is returned.



3.1.4. Use DTS to synchronize MySQL data to an Alibaba Cloud Elasticsearch cluster in real time

当前中文与控制台界面不符，与TW确认，取消翻译

3.1.5. Use Canal to synchronize data to an Alibaba Cloud Elasticsearch cluster

This topic describes how to use Canal to synchronize incremental data from an ApsaraDB RDS for MySQL database to an Alibaba Cloud Elasticsearch cluster.

Prerequisites

> **Notice** Make sure that you have specified the same **region**, **zone**, **Virtual Private Cloud (VPC)**, **subnet**, and **security group** for the required services. These services include ApsaraDB RDS for MySQL, Alibaba Cloud Elasticsearch, and Alibaba Cloud Elastic Compute Service (ECS).

- ApsaraDB RDS for MySQL is activated.

  ApsaraDB RDS for MySQL stores data that you want to synchronize. For more information about how to activate ApsaraDB RDS MySQL, see Create an ApsaraDB RDS for MySQL instance. The following figure shows the ApsaraDB RDS for MySQL configuration that is used in this topic.

- canal.adapter-1.1.4.tar.gz and canal.deployer-1.1.4.tar.gz are prepared.

  The Canal packages. Canal is a GitHub open-source extract, transform, and load (ETL) tool. It is used to parse database logs and retrieve incremental data for data synchronization. For more information, see Canal.

- Alibaba Cloud Elasticsearch is activated.

  Alibaba Cloud Elasticsearch receives the synchronized incremental data. For more information about how to activate Alibaba Cloud Elasticsearch, see Create an Alibaba Cloud Elasticsearch cluster. This topic uses an Alibaba Cloud Elasticsearch **V6.7** cluster of **the Standard Edition** as an example.

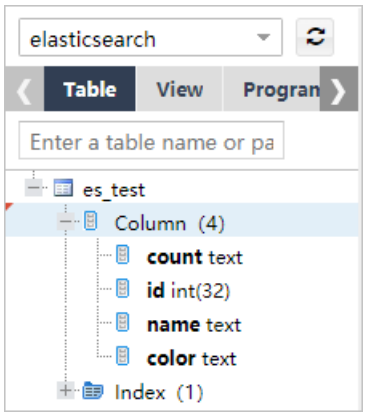
- Alibaba Cloud ECS is activated.

  Alibaba Cloud ECS connects ApsaraDB RDS for MySQL and Elasticsearch. In addition, both Canal deployer and Canal adapter are deployed on an Alibaba Cloud ECS instance. For more information about how to activate Alibaba Cloud ECS, see Step 1: Create an ECS instance. The **image** of the ECS instance is a **CentOS 7.6 64-bit** image.

Create a table and add fields

1. Create a table in an ApsaraDB RDS for MySQL instance and add fields to the table.

   In this topic, table **es_test** is created. The following figure shows the fields that are added to the table.



2. Create an index on the Elasticsearch cluster and configure mappings.

   Log on to the Kibana console. In the left-side navigation pane, click **Dev Tools**. On the **Console** tab of the page that appears, create an index and configure mappings.

> **Notice** Make sure that the field names and field types specified in the following command are the same as those in the created table.

```
PUT es_test?include_type_name=true
{
    "settings" : {
      "index" : {
        "number_of_shards" : "5",
        "number_of_replicas" : "1"
      }
    },
    "mappings" : {
        "_doc" : {
            "properties" : {
              "count": {
                   "type": "text"
               },
              "id": {
                   "type": "integer"
               },
                "name" : {
                    "type" : "text"
                },
                "color" : {
                    "type" : "text"
                }
            }
        }
    }
}
```

If the index is created and mappings are configured, the following result is returned:

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

Install MySQL

1. Connect to the ECS instance.
2. Download the MySQL source package.

   ```
   wget http://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm
   ```

3. Install the MySQL source.

   ```
   yum -y install mysql57-community-release-el7-11.noarch.rpm
   ```

4. Check whether the MySQL source is successfully installed.

```
yum repolist enabled | grep mysql.*
```

If the MySQL source is successfully installed, the result shown in the following figure is returned.

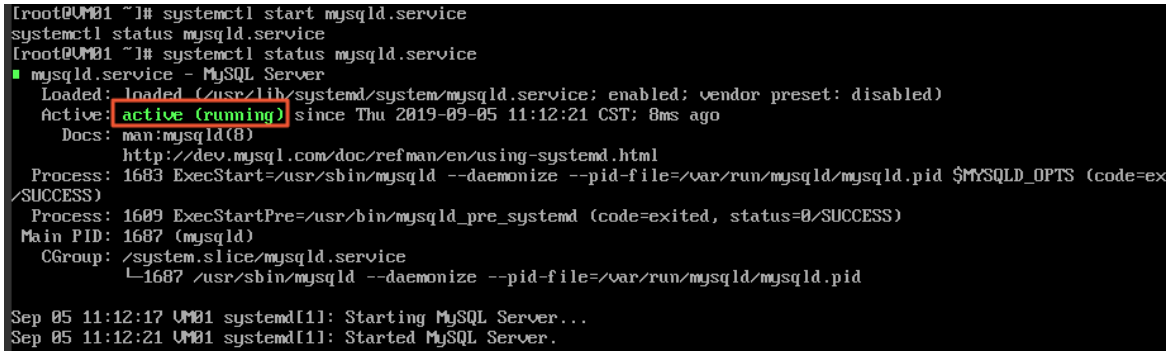
5. Install MySQL.

```
yum install mysql-community-server
```

6. Start the MySQL service and check the service status.

```
systemctl start mysqld.service
systemctl status mysqld.service
```

If MySQL is successfully started, the result shown in the following figure is returned.



7. Connect to an ApsaraDB RDS for MySQL database.

> **Notice**
>
> - Before you run the required command to connect to an ApsaraDB RDS for MySQL database, you must add the **private IP address** of your ECS instance to the whitelist of the corresponding ApsaraDB RDS for MySQL instance. For more information, see Use a database client or the CLI to connect to an ApsaraDB RDS for MySQL instance.
> - To use Canal, you must enable the MySQL binlog mode. By default, this mode is enabled for ApsaraDB RDS for MySQL. You can run the following command to query the status of the binlog mode:
>
>   ```
>   show variables like '%log_bin%';
>   ```
>
>   If the binlog mode is enabled, the result shown in the following figure is returned.
>
>   
>

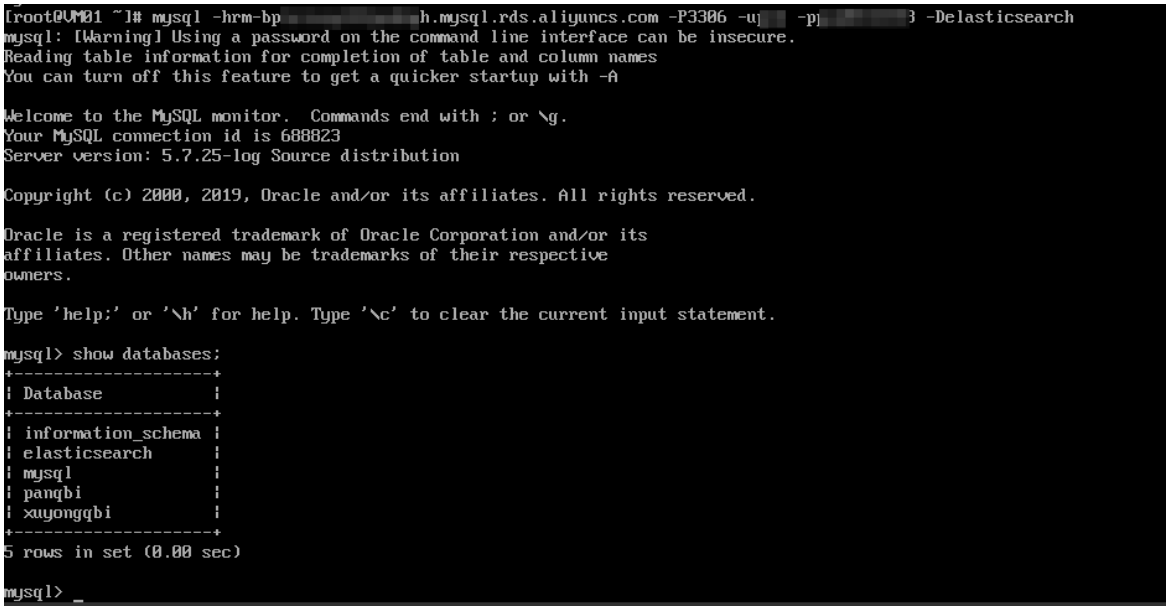
```
mysql -h<Hostname> -P<Port> -u<Username> -p<Password> -D<Database>
```

Variable	Description
<Hostname>	The internal endpoint of the ApsaraDB RDS for MySQL instance. You can query the internal endpoint on the **Basic Information** page of the ApsaraDB RDS for MySQL instance.
<Port>	The internal port of the ApsaraDB RDS for MySQL instance. The default port is **3306**. You can query the internal port on the **Basic Information** page of the ApsaraDB RDS for MySQL instance.
<Username>	The username that is used to log on to the ApsaraDB RDS for MySQL database. You can query the username on the **Accounts** page of the ApsaraDB RDS for MySQL instance. If no account is available, you must create an account. For more information, see Create databases and accounts for an ApsaraDB RDS for MySQL instance.
<Database>	The name of the ApsaraDB RDS for MySQL database. You can query the database name on the **Databases** page of the ApsaraDB RDS for MySQL instance. If no database is available, you must create a database. For more information, see Create databases and accounts for an ApsaraDB RDS for MySQL instance.
<Password>	The password that is used to log on to the ApsaraDB RDS for MySQL database.

Example command:

```
mysql -hrm-bp1u1xxxxxxxxx6ph.mysql.rds.aliyuncs.com -P3306 -ues -pmima -Delasticsearch
```

If ApsaraDB RDS for MySQL is successfully connected, the result shown in the following figure is returned.



Install the JDK

1. Connect to the ECS instance and query available JDK packages.

```
yum search java | grep -i --color JDK
```

2. Install the JDK of the required version.

   java-1.8.0-openjdk-devel.x86_64 is used in this topic.

```
yum install java-1.8.0-openjdk-devel.x86_64
```

3. Configure environment variables.

   i. Open the profile file in the etc folder.

```
vi /etc/profile
```

   ii. Add the following environment variables to the file:

```
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.71-2.b15.el7_2.x86_64
export CLASSPATH=.:$JAVA_HOME/jre/lib/rt.jar:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib
/tools.jar
export PATH=$PATH:$JAVA_HOME/bin
```

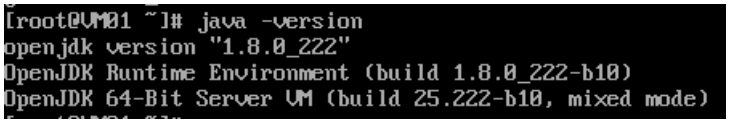
   iii. Enter `:wq` to save the file and quit the vi mode. Then, run the following command for the modification to take effect:

```
source /etc/profile
```

4. Run the following commands to check whether the JDK is successfully installed:

   - `java`
   - `javac`
   - `java -version`

   If the JDK is successfully installed, the result shown in the following figure is returned.



Install and start the Canal server

1. Connect to the ECS instance. Then, download and decompress the Canal deployer package.

   Canal deployer 1.1.4 is used in this topic.

```
wget https://github.com/alibaba/canal/releases/download/canal-1.1.4/canal.deployer-1.
1.4.tar.gz
```

2. Decompress the *canal.deployer-1.1.4.tar.gz* package.

```
tar -zxvf canal.deployer-1.1.4.tar.gz
```

3. Modify the *conf/example/instance.properties* file.

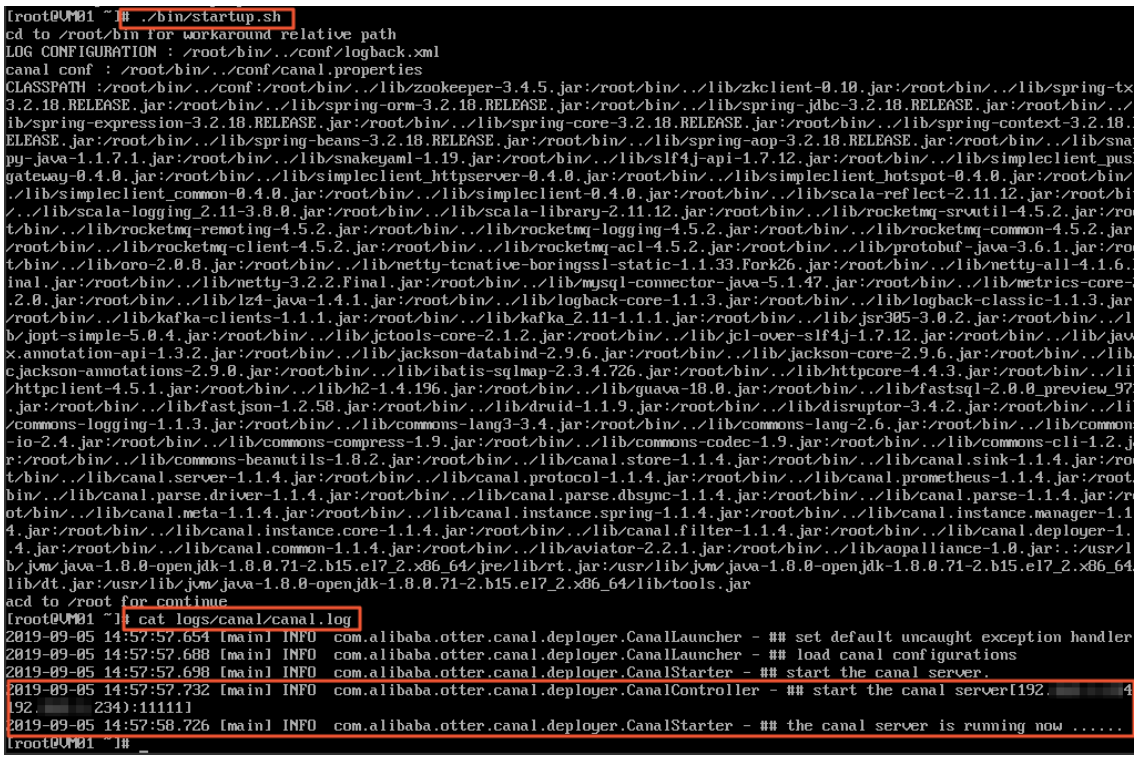
```
vi conf/example/instance.properties
```

```
#################################################
## mysql serverId , v1.0.26+ will autoGen
# canal.instance.mysql.slaveId=0

# enable gtid use true/false
canal.instance.gtidon=false

# position info
canal.instance.master.address=rm-bp[redacted]d6ph.mysql.rds.aliyuncs.com:3306
canal.instance.master.journal.name=
canal.instance.master.position=
canal.instance.master.timestamp=
canal.instance.master.gtid=

# rds oss binlog
canal.instance.rds.accesskey=
canal.instance.rds.secretkey=
canal.instance.rds.instanceId=

# table meta tsdb info
canal.instance.tsdb.enable=true
#canal.instance.tsdb.url=jdbc:mysql://127.0.0.1:3306/canal_tsdb
#canal.instance.tsdb.dbUsername=canal
#canal.instance.tsdb.dbPassword=canal

#canal.instance.standby.address =
#canal.instance.standby.journal.name =
#canal.instance.standby.position =
#canal.instance.standby.timestamp =
#canal.instance.standby.gtid=

# username/password
canal.instance.dbUsername=[redacted]
canal.instance.dbPassword=[redacted]
canal.instance.connectionCharset = UTF-8
# enable druid Decrypt database password
canal.instance.enableDruid=false
```

Parameter	Description
`canal.instance.master.address`	The parameter is in the format of `<Internal endpoint of the ApsaraDB RDS for MySQL instance>:<Internal port>`. You can query the required information on the **Basic Information** page of the ApsaraDB RDS for MySQL instance. Example: `rm-bp1u1xxxxxxxxx6ph.mysql.rds.aliyuncs.com:3306`.
`canal.instance.dbUsername`	The username that is used to log on to the ApsaraDB RDS for MySQL database. You can query the username on the **Accounts** page of the ApsaraDB RDS for MySQL instance.
`canal.instance.dbPassword`	The password that is used to log on to the ApsaraDB RDS for MySQL database.

4. Enter **:wq** to save the file and quit the vi mode.
5. Start the Canal server and query the log.

```
./bin/startup.sh
cat logs/canal/canal.log
```



Install and start the Canal adapter

1. Connect to the ECS instance. Then, download and decompress the Canal adapter package.

   Canal adapter 1.1.4 is used in this topic.

   ```
   wget https://github.com/alibaba/canal/releases/download/canal-1.1.4/canal.adapter-1.1.4.tar.gz
   ```

2. Decompress the *canal.adapter-1.1.4.tar.gz* package.

   ```
   tar -zxvf canal.adapter-1.1.4.tar.gz
   ```

3. Modify the *conf/application.yml* file.

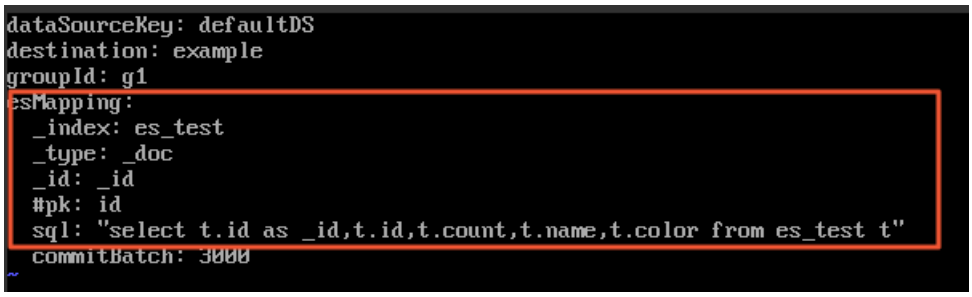
   ```
   vi conf/application.yml
   ```

```
canal.conf:
  mode: tcp # kafka rocketMQ
  canalServerHost: 127.0.0.1:11111
#  zookeeperHosts: slave1:2181
#  mqServers: 127.0.0.1:9092 #or rocketmq
#  flatMessage: true
  batchSize: 500
  syncBatchSize: 1000
  retries: 0
  timeout:
  accessKey:
  secretKey:
  srcDataSources:
    defaultDS:
      url: jdbc:mysql://rm-bp[redacted]ph.mysql.rds.aliyuncs.com:3306/elasticsearch?useUnicode=true
      username: [redacted]
      password: p[redacted]
  canalAdapters:
  - instance: example # canal instance Name or mq topic name
    groups:
    - groupId: g1
      outerAdapters:
      - name: logger
#      - name: rdb
#        key: mysql1
#        properties:
#          jdbc.driverClassName: com.mysql.jdbc.Driver
#          jdbc.url: jdbc:mysql://127.0.0.1:3306/mytest2?useUnicode=true
#          jdbc.username: root
#          jdbc.password: 121212
#      - name: rdb
#        key: oracle1
#        properties:
#          jdbc.driverClassName: oracle.jdbc.OracleDriver
#          jdbc.url: jdbc:oracle:thin:@localhost:49161:XE
#          jdbc.username: mytest
#          jdbc.password: m121212
#      - name: rdb
#        key: postgres1
#        properties:
#          jdbc.driverClassName: org.postgresql.Driver
#          jdbc.url: jdbc:postgresql://localhost:5432/postgres
#          jdbc.username: postgres
#          jdbc.password: 121212
#          threads: 1
#          commitSize: 3000
#      - name: hbase
#        properties:
#          hbase.zookeeper.quorum: 127.0.0.1
#          hbase.zookeeper.property.clientPort: 2181
#          zookeeper.znode.parent: /hbase
      - name: es
        hosts: es-cn-v6[redacted]dp.elasticsearch.aliyuncs.com:9200
        properties:
          mode: rest # or transport
          security.auth: elastic:[redacted]3
          cluster.name: es-cn-v6[redacted]dp
```

Parameter	Description
`canal.conf.canalServerHost`	The endpoint of the Canal deployer. Retain the default value: `127.0.0.1:11111`.
`canal.conf.srcDataSources.defaultDS.url`	`jdbc:mysql://<Internal endpoint of the ApsaraDB RDS for MySQL instance>:<Internal port>/<Database name>?useUnicode=true`. You can query the required information on the **Basic Information** page of the ApsaraDB RDS for MySQL instance. Example: `jdbc:mysql://rm-bp1xxxxxxxxxnd6ph.mysql.rds.aliyuncs.com:3306/elasticsearch?useUnicode=true`.
`canal.conf.srcDataSources.defaultDS.username`	The username that is used to log on to the ApsaraDB RDS for MySQL database. You can query the username on the **Accounts** page of the ApsaraDB RDS for MySQL instance.

Parameter	Description
canal.conf.srcDataSources.defaultDS.password	The password that is used to log on to the ApsaraDB RDS for MySQL database.
canal.conf.canalAdapters.groups.outerAdapters.hosts	Find name:es and set hosts to a value in the format of <Internal endpoint of the Elasticsearch cluster>:<Internal port>. You can query the required information on the Basic Information page of the Elasticsearch cluster. Example: es-cn-v64xxxxxxxx3medp.elasticsearch.aliyuncs.com:9200.
canal.conf.canalAdapters.groups.outerAdapters.mode	Set the value to rest.
canal.conf.canalAdapters.groups.outerAdapters.properties.security.auth	The parameter is in the format of <Username of the Elasticsearch cluster>:<Password>. Example: elastic:es_password.
canal.conf.canalAdapters.groups.outerAdapters.properties.cluster.name	The ID of the Elasticsearch cluster. You can query the cluster ID on the Basic Information page of the Elasticsearch cluster. Example: es-cn-v64xxxxxxxx3medp.

4. Enter **:wq** to save the file and exit the vi mode.
5. Repeat the preceding steps to modify the *conf/es/*.yml* file and specify the fields that you want to map from an ApsaraDB RDS for MySQL database to an Elasticsearch cluster.



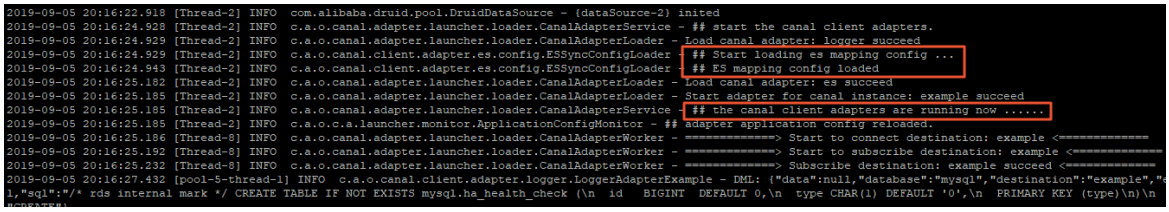
Parameter	Description
esMapping._index	Set the value to the name of the index created on the Elasticsearch cluster in the Create a table and add fields section. **es_test** is used in this topic.
esMapping._type	Set the value to the type of the index created on the Elasticsearch cluster in the Create a table and add fields section. **_doc** is used in this topic.
esMapping._id	The ID of the document that you want to synchronize to the Elasticsearch cluster. This parameter is user-defined. **_id** is used in this topic.

Parameter	Description
esMapping.sql	The SQL statement that is used to query the fields to be synchronized to the Elasticsearch cluster. The `select t.id as _id,t.id,t.count,t.name,t.color from es_test t;` statement is used in this topic.

6. Start the Canal adapter and query the log.

```
./bin/startup.sh
cat logs/adapter/adapter.log
```

If the Canal adapter is successfully started, the result shown in the following figure is returned.



Verify the incremental data synchronization result

1. In an ApsaraDB RDS for MySQL database, add, modify, or delete data in the **es_test** table.

```
insert `elasticsearch`.`es_test`(`count`,`id`,`name`,`color`) values('11',2,'canal_test2','red');
```

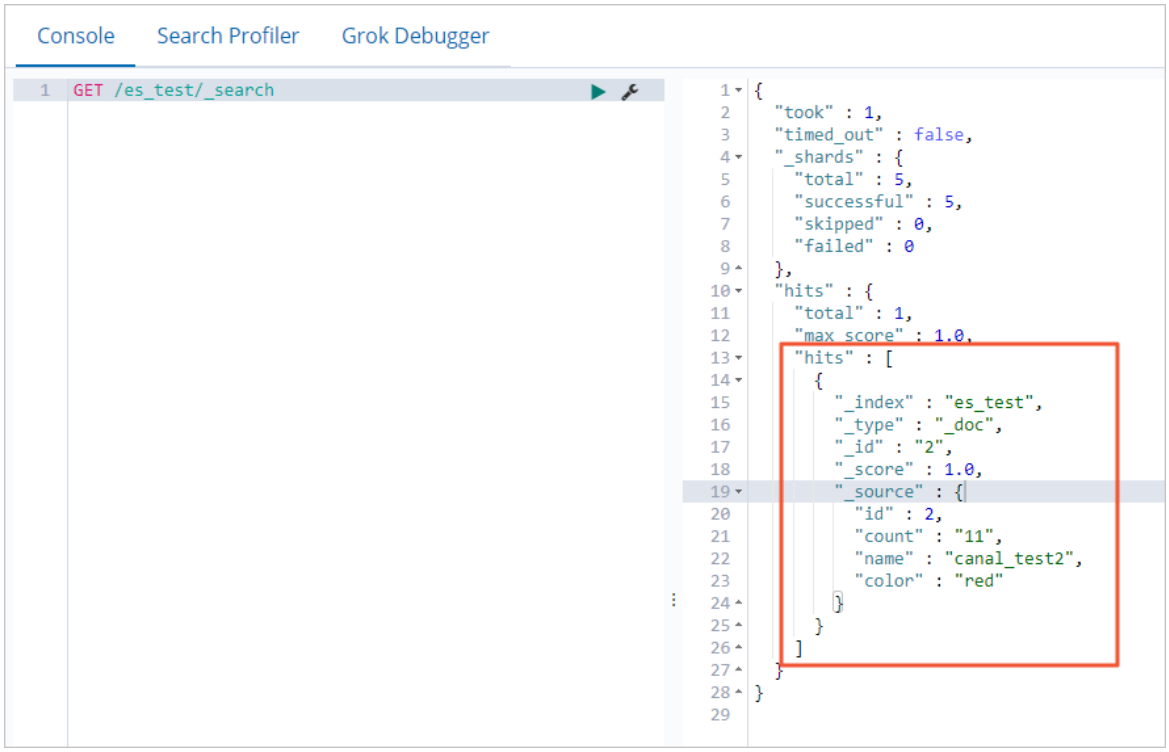
2. Log on to the Kibana console of the Elasticsearch cluster.

   For more information, see Log on to the Kibana console.

3. In the left-side navigation pane, click **Dev Tools**. On the **Console** tab of the pane that appears, run the following command to query synchronized data:

```
GET /es_test/_search
```

If the incremental data is successfully synchronized, the result shown in the following figure is returned.



3.2. PolarDB-X(DRDS) synchronization

3.2.1. Use DataWorks to synchronize data from a DRDS database to an Elasticsearch cluster in offline mode

Your business data is stored in a Distributed Relational Database Service (DRDS) database. If you want to perform full-text searches and semantic analytics on the data, you can synchronize the data to an Alibaba Cloud Elasticsearch cluster in offline mode.

Context

DRDS is developed by Alibaba Cloud. It integrates the distributed SQL engine DRDS and the proprietary distributed storage X-DB. DRDS supports tens of millions of concurrent connections and can store hundreds of petabytes of data based on the integrated cloud-native architecture. DRDS aims to provide solutions for massive data storage, ultra-high concurrent throughput, large table performance bottlenecks, and complex computing efficiency. DRDS has become a mature service after it is applied to Double 11 and the business of Alibaba Cloud customers in various industries. The application of DRDS boosts the digital transformation of enterprises. For more information, see Overview.

Alibaba Cloud Elasticsearch is compatible with open source Elasticsearch features, such as Security, Machine Learning, Graph, and Application Performance Monitoring (APM). It is released in 5.5.3, 6.3.2, 6.7.0, 6.8.0, 7.4.0, and 7.7.1 versions. It supports the commercial plug-in X-Pack and is ideal for scenarios such as data analytics and searches. Alibaba Cloud Elasticsearch implements enterprise-grade access control, security monitoring and alerting, and automated reporting based on open source Elasticsearch. For more information, see What is Alibaba Cloud Elasticsearch.

Procedure

1. Preparations

   Create a DRDS instance and an Alibaba Cloud Elasticsearch cluster in the same virtual private cloud (VPC). Prepare the data that you want to migrate in the DRDS instance. Activate the Data Integration and DataStudio services of DataWorks.

   > **Note** To improve the stability of synchronization nodes, we recommend that you synchronize data within a VPC.

2. Step 1: Purchase and create an exclusive resource group

   In the DataWorks console, purchase and create an exclusive resource group. To ensure the network connection, you must bind the exclusive resource group to the VPC where the DRDS instance resides.

   > **Note** Exclusive resource groups can be used to transmit data in a fast and stable manner.

3. Step 2: Add data sources

   In the DataWorks console, add the DRDS instance and the Elasticsearch cluster as data sources.

4. Step 3: Create and run a data synchronization node

   Use the codeless user interface (UI) to create a node to synchronize data from the MySQL data source to the Elasticsearch cluster and configure the node. Select the exclusive resource group that you created when you configure the node. The data synchronization node runs on the selected exclusive resource group for Data Integration and writes data to the Elasticsearch cluster.

5. Step 4: View synchronization results

   In the Kibana console of the Elasticsearch cluster, view the synchronized data and search for data by using a specific field.

Preparations

1. Create a DRDS V1.0 instance, a DRDS database, and a table. Then, insert data into the table.

   For more information, see Basic SQL operations. The following figure shows the test data that is used in this topic.

Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Critic_Score
Wii Sports	Wii	2006	Sports	Nintendo	41.36	28.96	3.77	8.45	82.53	76
Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24	null
Mario Kart Wii	Wii	2008	Racing	Nintendo	15.68	12.76	3.79	3.29	35.52	82
Wii Sports Resort	Wii	2009	Sports	Nintendo	15.61	10.93	3.28	2.95	32.77	80
Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	Nintendo	11.27	8.89	10.22	1	31.37	null
Tetris	GB	1989	Puzzle	Nintendo	23.2	2.26	4.22	0.58	30.26	null
New Super Mario Bros.	DS	2006	Platform	Nintendo	11.28	9.14	6.5	2.88	29.8	89
Wii Play	Wii	2006	Misc	Nintendo	13.96	9.18	2.93	2.84	28.92	58
New Super Mario Bros. Wii	Wii	2009	Platform	Nintendo	14.44	6.94	4.7	2.24	28.32	87
Duck Hunt	NES	1984	Shooter	Nintendo	26.93	0.63	0.28	0.47	28.31	null
Nintendogs	DS	2005	Simulation	Nintendo	9.05	10.95	1.93	2.74	24.67	null
Mario Kart DS	DS	2005	Racing	Nintendo	9.71	7.47	4.13	1.9	23.21	91
Pokemon Gold/Pokemon Silver	GB	1999	Role-Playing	Nintendo	9	6.18	7.2	0.71	23.1	null
Wii Fit	Wii	2007	Sports	Nintendo	8.92	8.03	3.6	2.15	22.7	80
Kinect Adventures!	X360	2010	Misc	Microsoft Game Studios	15	4.89	0.24	1.69	21.81	61
Wii Fit Plus	Wii	2009	Sports	Nintendo	9.01	8.49	2.53	1.77	21.79	80
Grand Theft Auto V	PS3	2013	Action	Take-Two Interactive	7.02	9.09	0.98	3.96	21.04	97
Grand Theft Auto: San Andreas	PS2	2004	Action	Take-Two Interactive	9.43	0.4	0.41	10.57	20.81	95
Super Mario World	SNES	1990	Platform	Nintendo	12.78	3.75	3.54	0.55	20.61	null
Brain Age: Train Your Brain in Minutes a Day	DS	2005	Misc	Nintendo	4.74	9.2	4.16	2.04	20.15	77
Pokemon Diamond/Pokemon Pearl	DS	2006	Role-Playing	Nintendo	6.38	4.46	6.04	1.36	18.25	null
Super Mario Land	GB	1989	Platform	Nintendo	10.83	2.71	4.18	0.42	18.14	null
Super Mario Bros. 3	NES	1988	Platform	Nintendo	9.54	3.44	3.84	0.46	17.28	null

> **Notice** After a database is created, all IP addresses are allowed to access the database by default. For security purposes, we recommend that you add only the IP address of the host that you use to the whitelist of the DRDS instance. For more information, see Set an IP address whitelist.

2. Create a DataWorks workspace.

   For more information, see Create a workspace. The workspace must reside in the same region as the DRDS instance that you created.

3. Create an Elasticsearch cluster and enable the Auto Indexing feature for the cluster.

   For more information, see Create an Alibaba Cloud Elasticsearch cluster and Access and configure an Elasticsearch cluster. The cluster must belong to the same VPC and vSwitch as the DRDS instance.

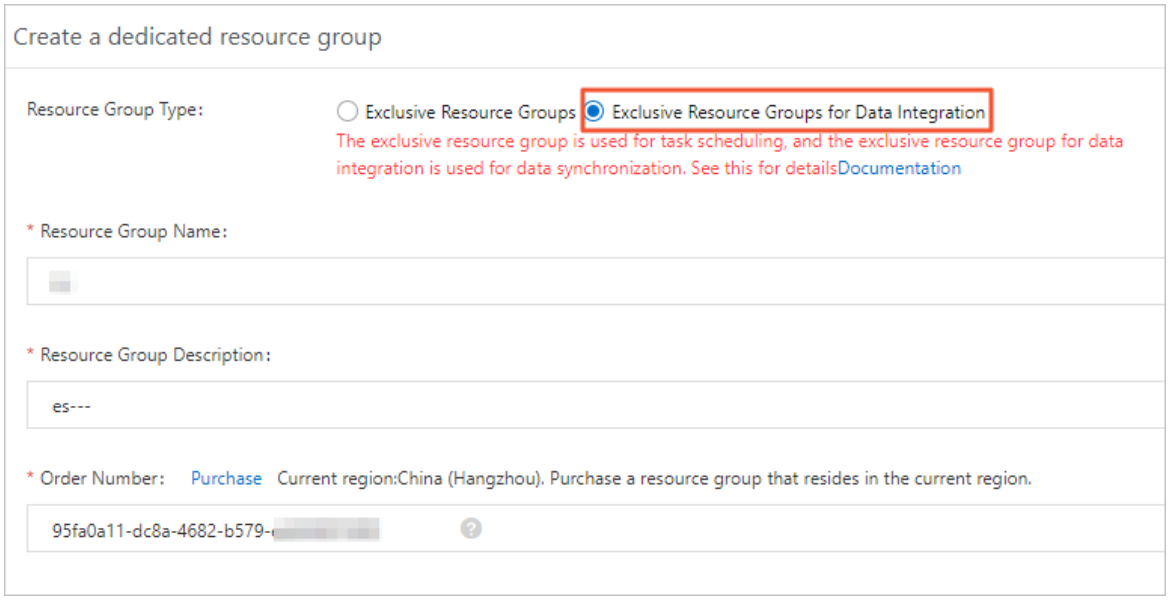
Step 1: Purchase and create an exclusive resource group

1. Log on to the DataWorks console.
2. In the top navigation bar, select the desired region. In the left-side navigation pane, click **Resource Groups**.
3. Purchase exclusive resources for Data Integration. For more information, see Purchase exclusive resources for Data Integration.

> **Notice** The exclusive resources for Data Integration must reside in the same region as the DataWorks workspace that you created.

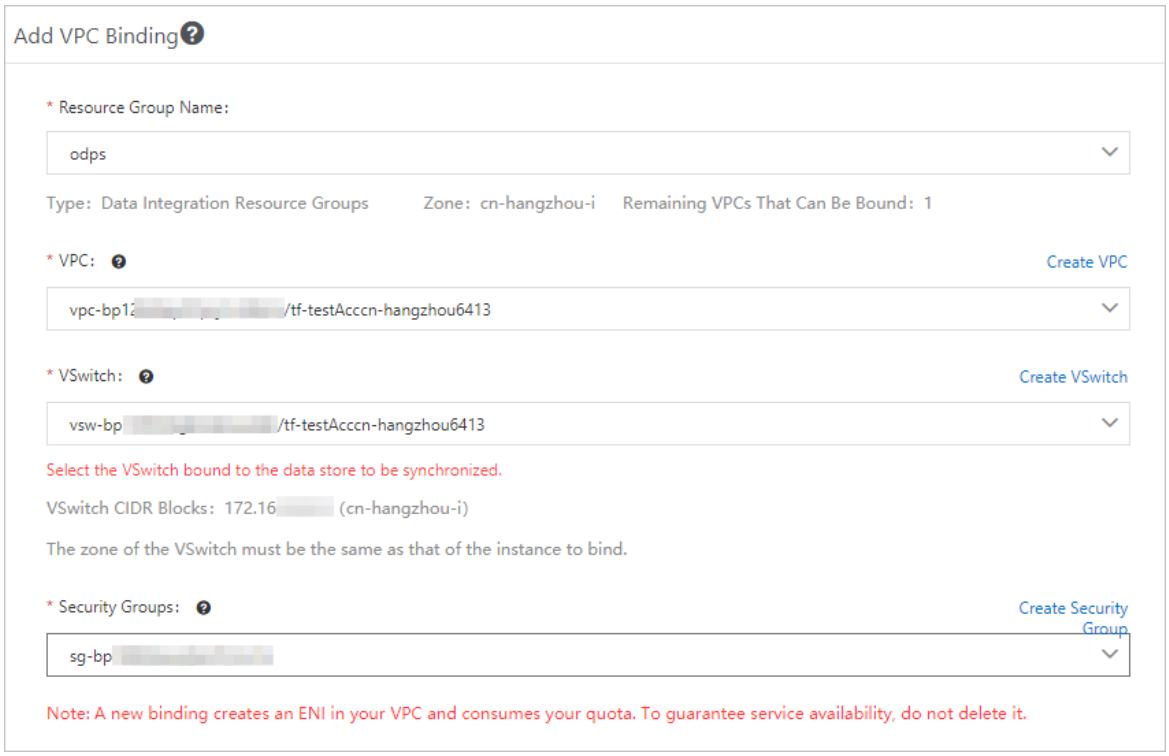
4. Create an exclusive resource group for Data Integration. For more information, see Create an exclusive resource group for Data Integration.

   The following figure shows the configuration used in this example. **Resource Group Type** is set to **Exclusive Resource Groups for Data Integration**.



5. Find the created exclusive resource group and click Network Settings in the Actions column. The VPC Binding tab appears. On the **VPC Binding** tab, click Add Binding to bind the exclusive resource group to a VPC. For more information, see Configure network settings.

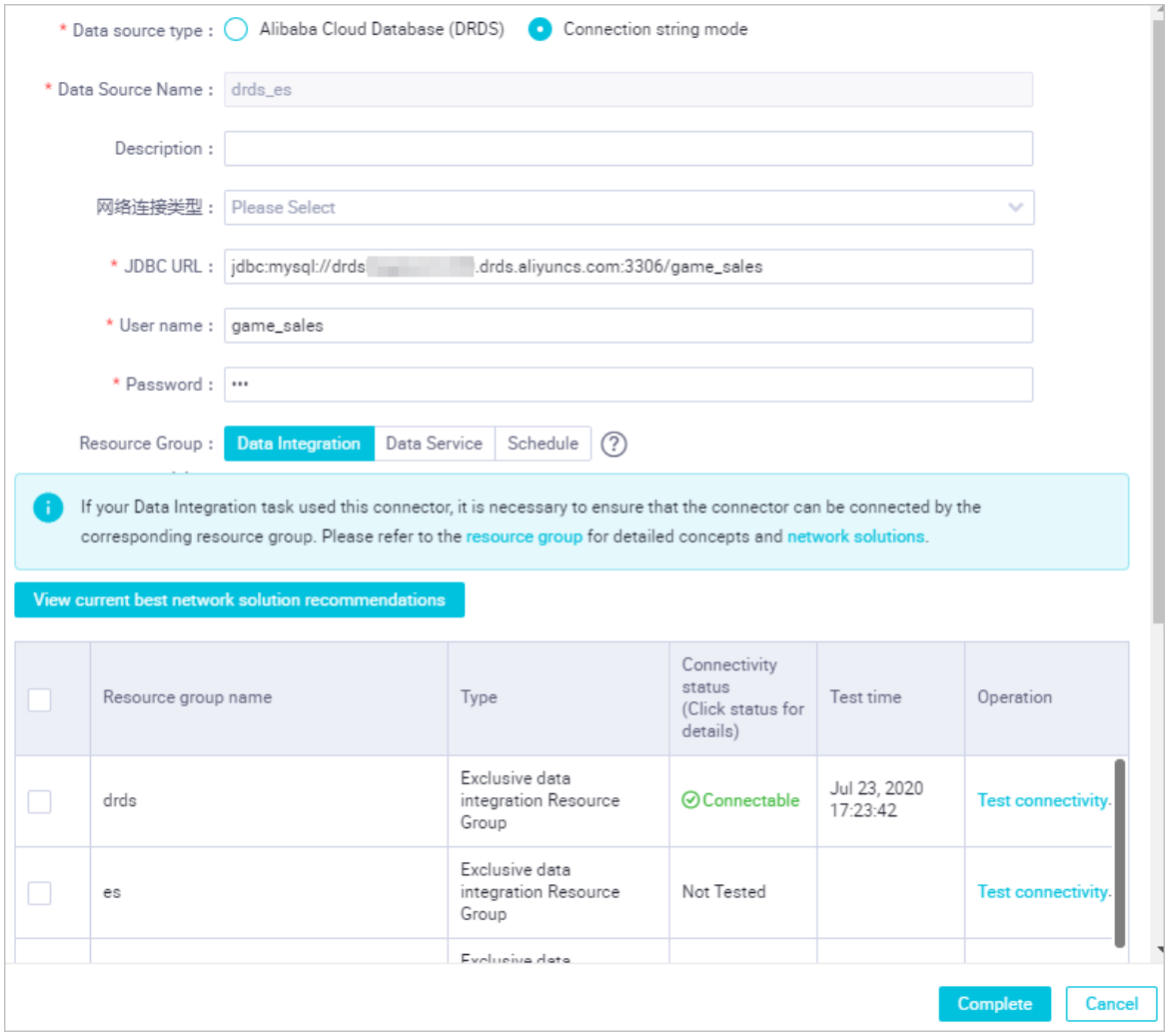
Exclusive resources are deployed in the VPC where DataWorks resides. You can use DataWorks to synchronize data from the DRDS database to the Elasticsearch cluster only after DataWorks connects to the VPCs where the database and cluster reside. In this topic, the DRDS database and Elasticsearch cluster reside in the same VPC. Therefore, when you bind the exclusive resource group to a VPC, you need to select the **VPC** and **vSwitch** to which the DRDS instance belongs.



6.

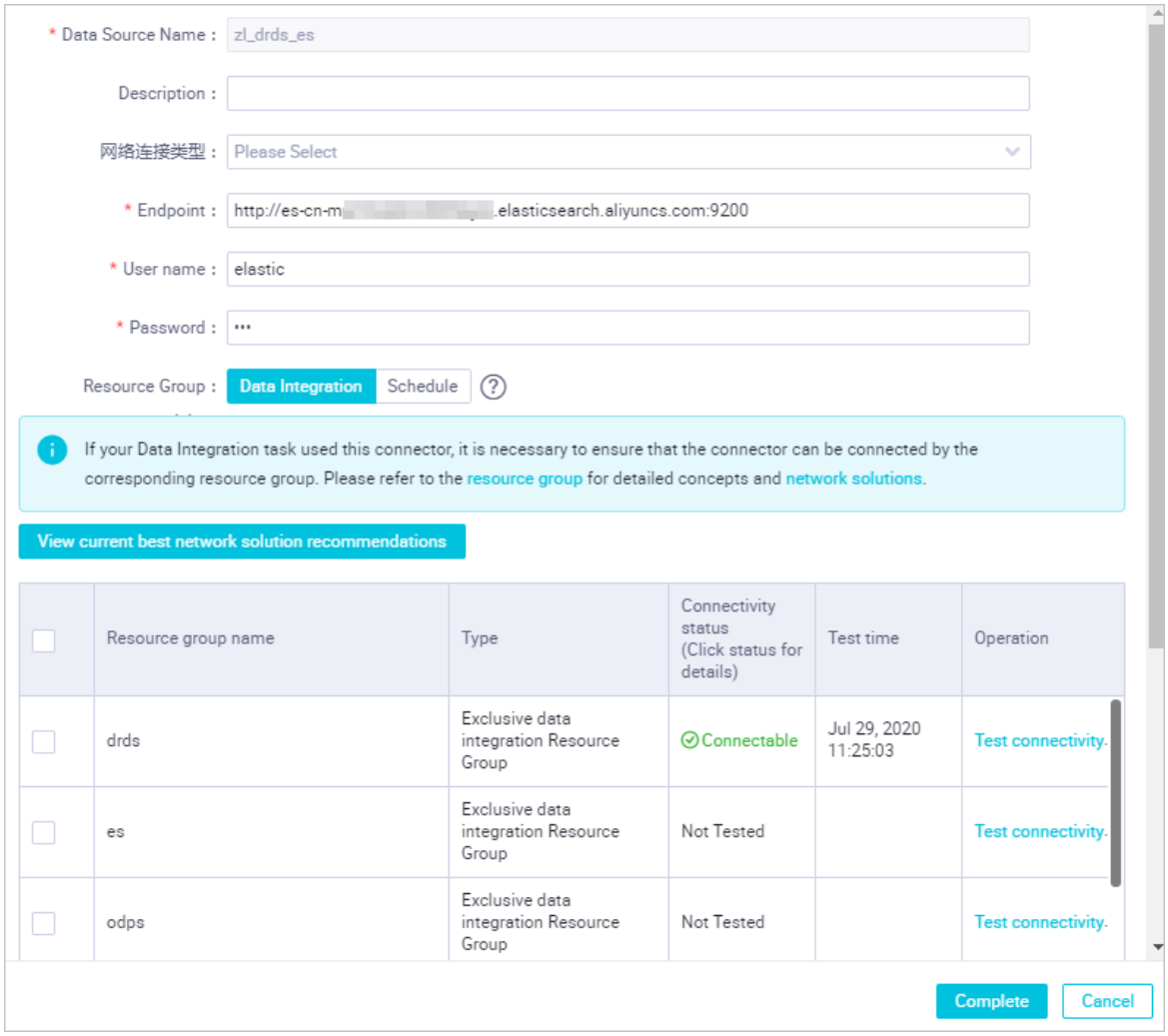
Step 2: Add data sources

1.
2. In the left-side navigation pane of the Data Integration page, choose **Data Source** > Data Sources.
3. On the Data Source page, click **Add data source** in the upper-right corner.
4. In the Relational Database section of the **Add data source** dialog box, click **DRDS**.
5. In the **Add DRDS data source** dialog box, configure the parameters and test connectivity between the DRDS data source and resource group that you created. After the connectivity test is passed, click **Complete**.



Parameter	Description
Data source type	In this topic, this parameter is set to **Connection string mode**. You can also set this parameter to **Alibaba Cloud Database (DRDS)**. For more information, see Add a DRDS data source.
Data Source Name	The name of the data source. The name must contain letters, digits, and underscores (_). It must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
JDBC URL	The Java Database Connectivity (JDBC) URL of the database, in the format of jdbc:mysql://ServerIP:Port/Database. Replace ServerIP:Port with Endpoint of the VPC where the DRDS instance resides:Port number of the VPC. Replace Database with the name of the DRDS database that you created.
User name	The username that is used to connect to the DRDS database.
Password	The password that is used to connect to the DRDS database.

6. Add an Elasticsearch data source in the same way.



Parameter	Description
Data Source Name	The name of the data source. The name must contain letters, digits, and underscores (_). It must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Endpoint	Set this parameter to a value in the format of http://<Internal endpoint of the Elasticsearch cluster>:9200. You can obtain the internal endpoint from the Basic Information page of the cluster.
Username	The username that is used to access the Elasticsearch cluster. The default username is **elastic**.
Password	The password that corresponds to the elastic username. The password of the **elastic** username is specified when you create the cluster. If you forget the password, you can reset it. For more information about the procedure and precautions for resetting a password, see Reset the access password for an Elasticsearch cluster.

Step 3: Create and run a data synchronization node

1. On the DataStudio page of the DataWorks console, create a workflow.

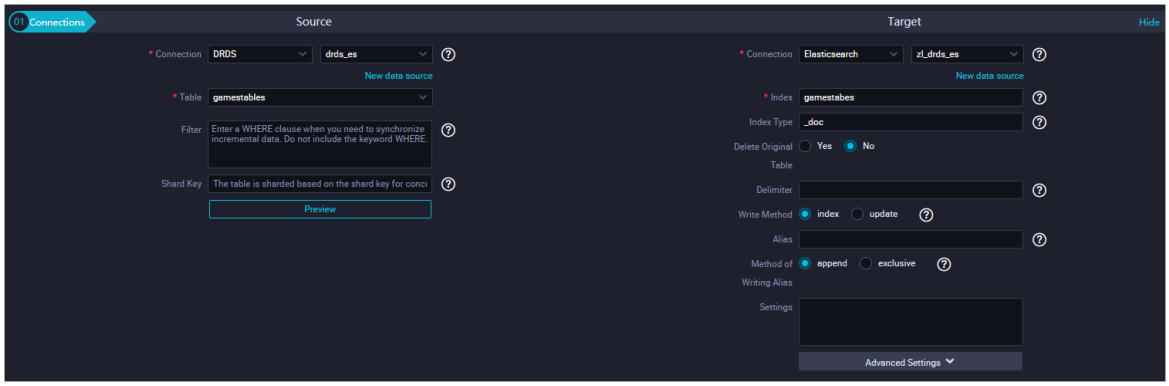
   For more information, see Manage workflows.

2. Create a batch synchronization node.

   i. In the DataStudio pane, open the newly created workflow, right-click **Data Integration**, and then choose **Create > Batch Synchronization**.

   ii. In the **Create Node** dialog box, configure the **Node Name** parameter and click **Commit**.

3. In the **Source** section of the **Connections** step, specify the DRDS data source and the name of the table that you created. In the **Target** section, specify the Elasticsearch data source, index name, and index type.



> **Note**
>
> - You can also use the code editor to configure the node. For more information, see Create a synchronization node by using the code editor, DRDS Reader, and Elasticsearch Writer.
> - We recommend that you set **Enable node discovery** to **No** in the **advanced settings** of the **Elasticsearch** data source. Otherwise, a connection timeout error occurs during data synchronization.

4. In the **Mappings** step, configure mappings between **source fields** and **destination fields**.

   In this example, the default **source fields** are used. You need only to change **destination fields**.

   Click the icon in the destination fields section on the right. In the Change Fields dialog box, enter the following information:

```
null
null
null
null
null
null
null
null
null
null
null
null
null
null
null
null
null
```

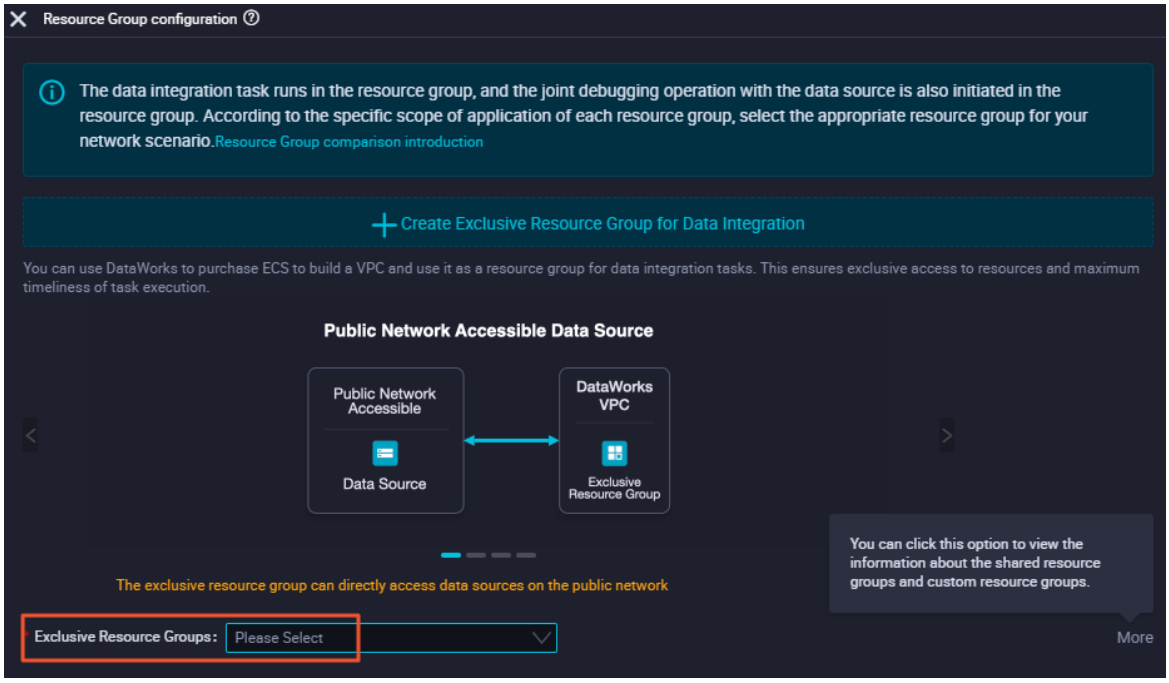
The following figure shows the configured field mappings.

5. In the **Channel** step, configure the parameters.
6. Configure properties for the node.

   In the right-side navigation pane of the configuration tab of the node, click **Properties**. On the Properties tab, configure properties for the node. For more information about the parameters, see Basic properties.

> **Notice**
>
> - Before you commit a node, you must configure a **dependent ancestor node** for the node in the Dependencies section of the Properties tab. For more information, see Configure same-cycle scheduling dependencies.
> - If you want the system to periodically run a node, you must configure time properties for the node in the **Schedule** section of the Properties tab. The time properties include Validity Period, Scheduling Cycle, Run At, and Rerun.
> - The configuration of an auto triggered node takes effect at 00:00 of the next day.

7. Configure the resource group that you want to use to run the synchronization node.



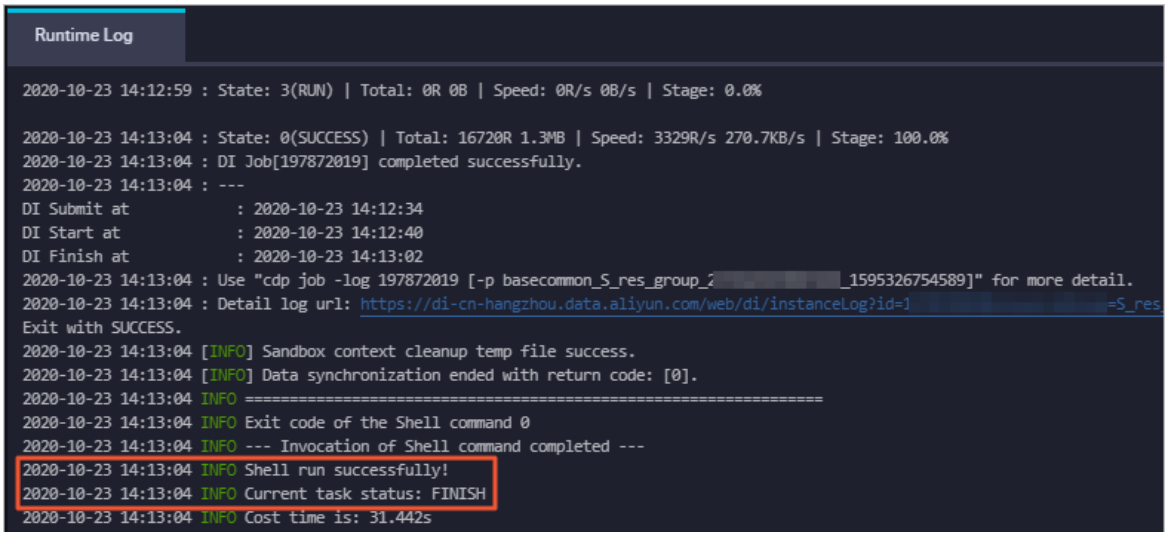
   i. In the right-side navigation pane of the configuration tab of the node, click the **Resource Group configuration** tab.
   ii. Select the exclusive resource group that you create from the **Exclusive Resource Groups** drop-down list.

8. Commit the node.

   i. Save the current configurations and click the icon in the top toolbar.
   ii. In the **Commit Node** dialog box, enter your comments in the **Change description** field.
   iii. Click **OK**.

9. Click the icon in the top navigation bar to run the node.

   You can view the operational logs of the node when the node is running. **successfully** indicates that the node is successfully run. **FINISH** indicates that the running of the node is complete.



> **Note** Before you run the node, you can configure properties for the node and select the desired resource group to run the node. For more information, see Configure basic properties .

Step 4: View synchronization results

1. Log on to the Kibana console of the destination Elasticsearch cluster.

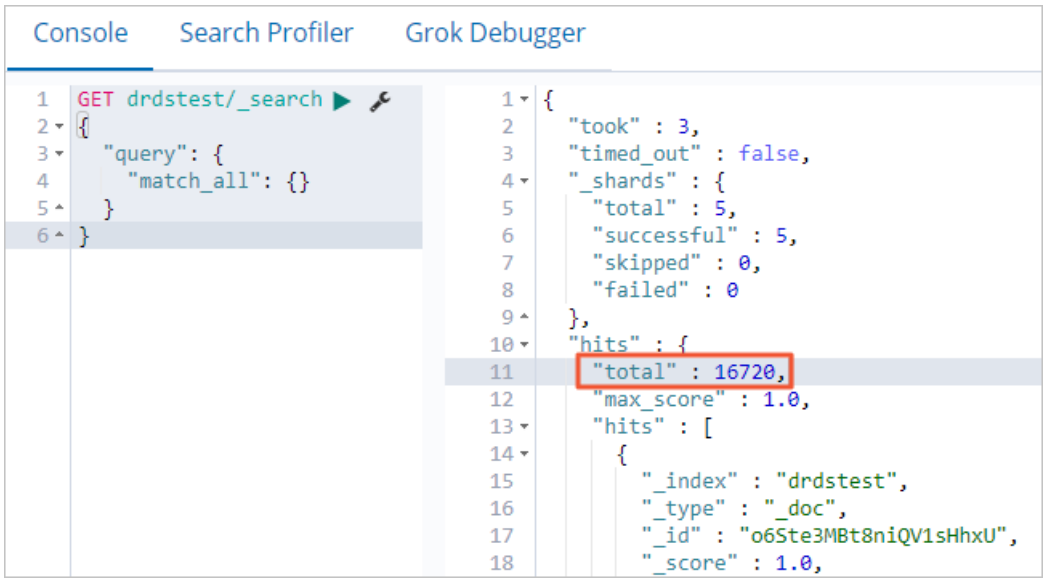
   For more information, see Log on to the Kibana console.

2. In the left-side navigation pane, click **Dev Tools**.
3. On the **Console** tab of the page that appears, run the following command to query the volume of data in the Elasticsearch cluster.

   > **Note** You can compare the queried data volume with the volume of data in the DRDS database to check whether all data is synchronized.

```
GET drdstest/_search
{
  "query": {
    "match_all": null
  }
}
```

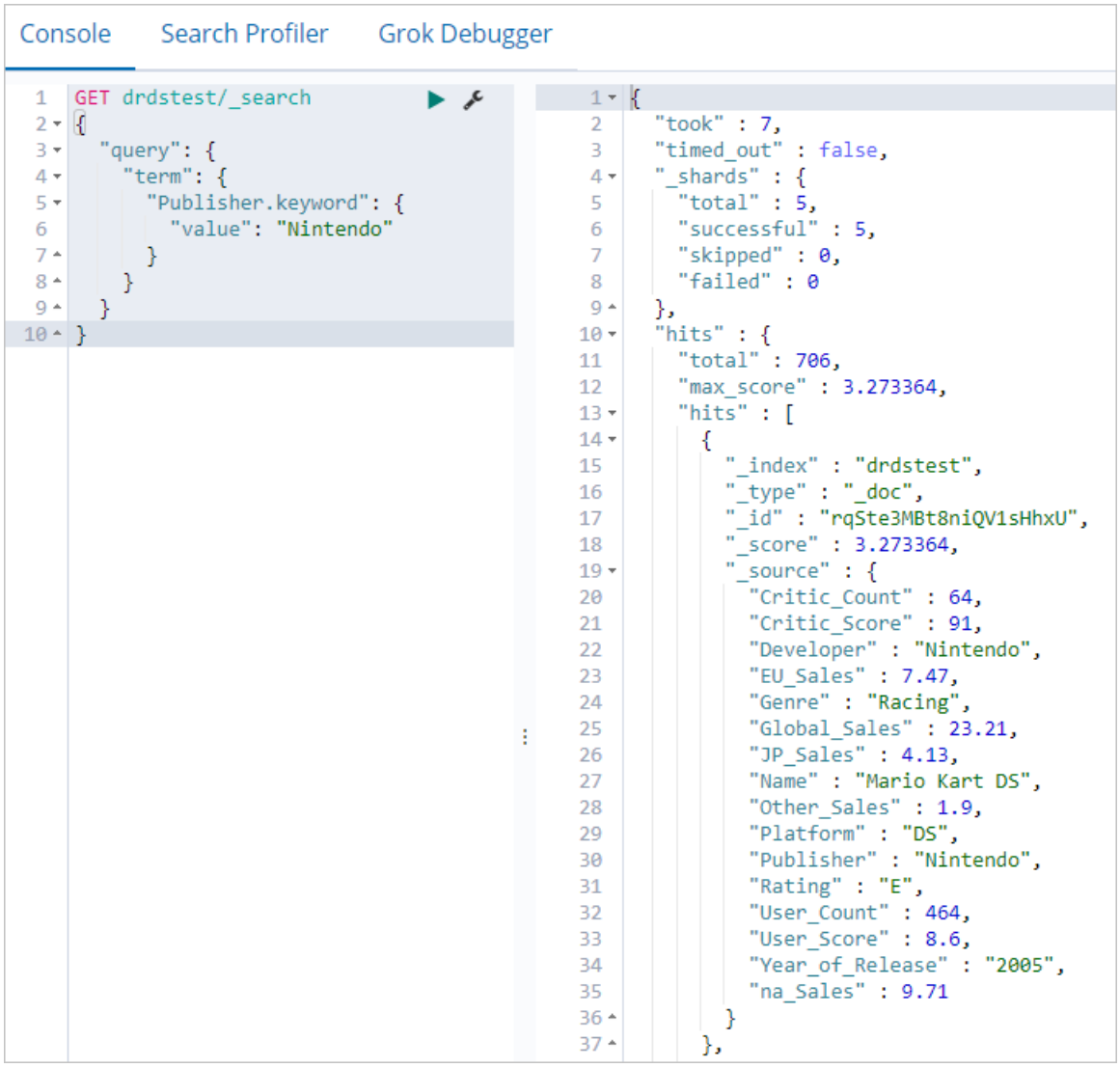
If the command is successfully run, the result shown in the following figure is returned.



4. Run the following command to search for data by using a specific field:

```
GET drdstest/_search
{
  "query": {
    "term": {
      "Publisher.keyword": {
        "value": "Nintendo"
      }
    }
  }
}
```

If the command is successfully run, the result shown in the following figure is returned.



3.3. Use DTS to synchronize data from a PolarDB for MySQL database to an Alibaba Cloud Elasticsearch cluster

If you encounter slow queries when you use a PolarDB for MySQL database, you can use Data Transmission Service (DTS) to synchronize production data from the database to an Elasticsearch cluster in real time. Then, you can search for and analyze the synchronized data in the Elasticsearch cluster. This topic describes how to synchronize data from a PolarDB for MySQL database to an Elasticsearch cluster.

Context

The following cloud services are used:

- DTS is a data transmission service that integrates data migration, data subscription, and real-time data synchronization. For more information, see DTS. You can use DTS to synchronize these SQL

statements: INSERT, DELETE, and UPDATE.

> **Notice** When you synchronize data, you must select a data source and a version that are supported by DTS.

- PolarDB is a next-generation relational database service developed by Alibaba Cloud. It is compatible with MySQL, PostgreSQL, and Oracle database engines. A PolarDB cluster can provide a maximum of 100 TB of storage space and can be scaled to a maximum of 16 nodes. PolarDB provides superior performance in storage and computing to meet diverse requirements of enterprises. For more information, see PolarDB for MySQL overview.
- Elasticsearch is a Lucene-based, distributed, real-time search and analytics engine. It allows you to store, query, and analyze large amounts of datasets in near real time. In most cases, it is used as a basic engine or technology to accommodate complex queries and high application performance. For more information, see What is Alibaba Cloud Elasticsearch?.

This topic can be used to guide real-time synchronization for data in relational databases.

Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU utilization of the source and destination databases is less than 30%.
- DTS does not synchronize data definition language (DDL) operations. If a DDL operation is performed on a table in the source database during data synchronization, you must perform the following steps: Remove the table from the required objects, remove the index for the table from the Elasticsearch cluster, and then add the table to the required objects. For more information, see Remove an object from a data synchronization task and Add an object to a data synchronization task.
- To add columns to the table that you want to synchronize, perform the following steps: Modify the mapping of the table in the Elasticsearch cluster, perform DDL operations in the source MySQL database, and then pause and start the data synchronization task.

Preparations

-
- Create a PolarDB for MySQL cluster and enable binary logging.

  For more information, see Purchase a pay-as-you-go cluster and Enable binary logging.
- Create a PolarDB for MySQL database and a table, and insert test data into the table.

  For more information, see Database Management.

- Table creation statement

```
CREATE TABLE `product` (
    `id` bigint(32) NOT NULL AUTO_INCREMENT,
    `name` varchar(32) NULL,
    `price` varchar(32) NULL,
    `code` varchar(32) NULL,
    `color` varchar(32) NULL,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB
DEFAULT CHARACTER SET=utf8;
```

- Test data

```
INSERT INTO `estest`.`product` (`id`,`name`,`price`,`code`,`color`) VALUES (1,'mobile p
hone A','2000','amp','golden');
INSERT INTO `estest`.`product` (`id`,`name`,`price`,`code`,`color`) VALUES (2,'mobile p
hone B','2200','bmp','white');
INSERT INTO `estest`.`product` (`id`,`name`,`price`,`code`,`color`) VALUES (3,'mobile p
hone C','2600','cmp','black');
INSERT INTO `estest`.`product` (`id`,`name`,`price`,`code`,`color`) VALUES (4,'mobile p
hone D','2700','dmp','red');
INSERT INTO `estest`.`product` (`id`,`name`,`price`,`code`,`color`) VALUES (5,'mobile p
hone E','2800','emp','silvery');
```

Procedure

1. Preparations

   Create an Elasticsearch cluster and a PolarDB for MySQL cluster and prepare test data.

2. Step 1: Configure and enable a data synchronization channel

   Use DTS to create and start a real-time task to synchronize data from the PolarDB for MySQL database to the Elasticsearch cluster.

3. Step 2: View the data synchronization result

   Log on to the Kibana console of the Elasticsearch cluster and query the synchronized data.

4. Step 3: Verify incremental data synchronization

   Add data to the PolarDB for MySQL database and check whether the data is synchronized to the Elasticsearch cluster.

Step 1: Configure and enable a data synchronization channel

1. Create a data synchronization task in the DTS console.
   i. Log on to the Data Transmission Service console.
   ii. In the left-side navigation pane, click **Data Synchronization**.
   iii. On the page that appears, click **Create Data Synchronization Task**. Then, purchase a data synchronization instance on the buy page as prompted.

   For more information, see Purchase a data synchronization instance. On the buy page, set Source Instance to **PolarDB**, Target Instance to **Elasticsearch**, and Synchronization Topology to **One-Way Synchronization**.

2. On the page that appears, select the region. Then, find the target instance and click **Configure**

Synchronization Channel in the **Actions** column.

3. In the **Create Data Synchronization Task** wizard, configure the PolarDB for MySQL cluster and Elasticsearch cluster for synchronization.

Field/Section	Parameter	Description
Synchronization Task Name	None	◦ DTS automatically generates a task name. You do not need to use a unique task name. ◦ We recommend that you use an informative name for easy identification.
Source Instance Details	**Instance Type**	The value of this parameter is **PolarDB Instance** and cannot be changed.
	Instance Region	The value of this parameter is the region that you selected for the PolarDB for MySQL cluster when you purchased the data synchronization instance. The value cannot be changed.
	PolarDB Instance ID	The ID of the PolarDB for MySQL cluster.
	Database Account	The account of the PolarDB for MySQL database from which you want to synchronize data. **Note** The account must have the read permissions on the database.
	Database Password	The password for the account of the PolarDB for MySQL database.
Destination Instance Details	**Instance Type**	The value of this parameter is **Elasticsearch** and cannot be changed.
	Instance Region	The value of this parameter is the region that you selected for the Elasticsearch cluster when you purchased the data synchronization instance. The value cannot be changed.
	Elasticsearch	The ID of the Elasticsearch cluster.
	Database Account	The username of the Elasticsearch cluster. Default value: elastic.
	Database Password	The password of the Elasticsearch cluster. Enter the password that corresponds to the username specified by **Database Account**.

4. Click **Set Whitelist and Next**. After the synchronization account is created, click **Next**.

> **Notice** In this step, the IP address of the DTS server is automatically added to the whitelists of the PolarDB for MySQL cluster and Elasticsearch cluster. This ensures that the DTS server communicates with both clusters.

5. Select the objects that you want to synchronize.

Parameter	Description
Index Name	○ **Table Name** If you select **Table Name**, the indexes and tables created on the Elasticsearch cluster use the same names as those on the ApsaraDB RDS for MySQL instance. ○ **DatabaseName_TableName** If you select **DatabseName_TableName**, the indexes created on the Elasticsearch cluster are named in the format of Database name_Table name.
Processing Mode In Existed Target Table	○ **Pre-check and Intercept**: The system checks whether the destination Elasticsearch cluster contains indexes that have the same names as tables in the source database. If the destination Elasticsearch cluster does not contain indexes that have the same names as tables in the source database, the precheck is passed. Otherwise, the system displays an error message during the precheck and does not start the data synchronization task. **Note** If indexes in the destination Elasticsearch cluster have the same names as tables in the source database, and cannot be deleted or renamed, you can perform the operations described in Rename an object to be synchronized to avoid table name conflicts. ○ **Ignore**: The system skips the precheck for identical table names in the source database and destination Elasticsearch cluster. **Warning** If you select **Ignore**, data inconsistency may occur and your business may be affected. ■ The source database and destination Elasticsearch cluster have the same mappings. If the primary key of a record in the destination Elasticsearch cluster is the same as that in the source database, the record remains unchanged during initial data synchronization. However, the record is overwritten during incremental data synchronization. ■ The source database and destination Elasticsearch cluster have different mappings. This may cause initial data synchronization to fail, only some columns to be synchronized, or the entire data synchronization to fail.

Parameter	Description
Objects to be synchronized	Select objects from the **Available** section and click the › button to move the objects to the **Selected** section.

6. In the **Selected** section, move the pointer over the name of the table whose data you want to synchronize and click **Edit**. In the Edit Table dialog box, configure parameters for the table in the Elasticsearch cluster, such as Index Name and Type Name. Then, click **OK**.

Parameter	Description
Index Name	For more information, see Terms.
Type Name	For more information, see Terms.
Filter	Specifies SQL filter conditions to filter data. Only data that meets the specific conditions is synchronized to the Elasticsearch cluster. For more information, see Use SQL conditions to filter data.
IsPartition	Specifies whether to set partitions. If you select **Yes**, you must also specify the partition key column and number of partitions.
_id value	○ **the primary key of table** Composite primary key fields are merged into one column. ○ **Bis id** If you select a business key, you must also specify the business key column.
add param	Select the required **column param** and **column param value** parameters. For more information, see Mapping parameters.

7. In the lower-right corner of the page, click **Precheck**.

> **Notice**
> - You can start a data synchronization task only after the task passes the precheck.
> - If the task fails to pass the precheck, click the ⓘ icon next to each failed item to view the details. Troubleshoot the issues and run the precheck again.

8. After the **The precheck is passed** message appears, close the **Precheck** dialog box.

   The data synchronization task starts. Data starts to synchronize until initial synchronization is complete and the synchronization task is in the **Synchronizing** state.

> **Notice** PolarDB for MySQL and Elasticsearch support different data types. During initial schema synchronization, DTS maps the data types of the PolarDB for MySQL database to those of the Elasticsearch cluster. For more information, see Data type mappings for schema synchronization.

Step 2: View the data synchronization result

1. Log on to the Kibana console of the Elasticsearch cluster.

   For more information, see Log on to the Kibana console.

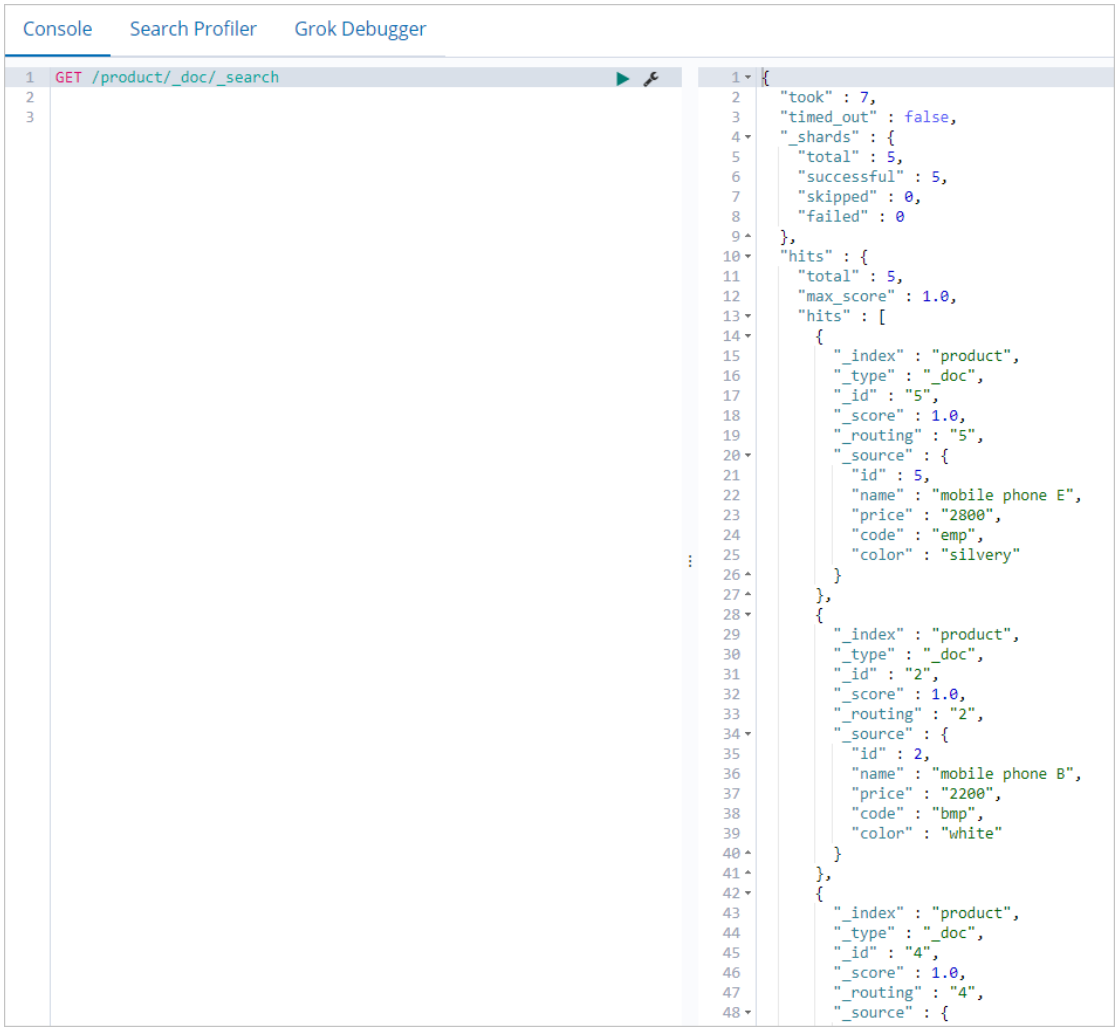
2. Use a command to query the synchronized data.

   i. In the left-side navigation pane, click **Dev Tools**.

   ii. On the **Console** tab of the page that appears, run the following command to query the synchronized data:

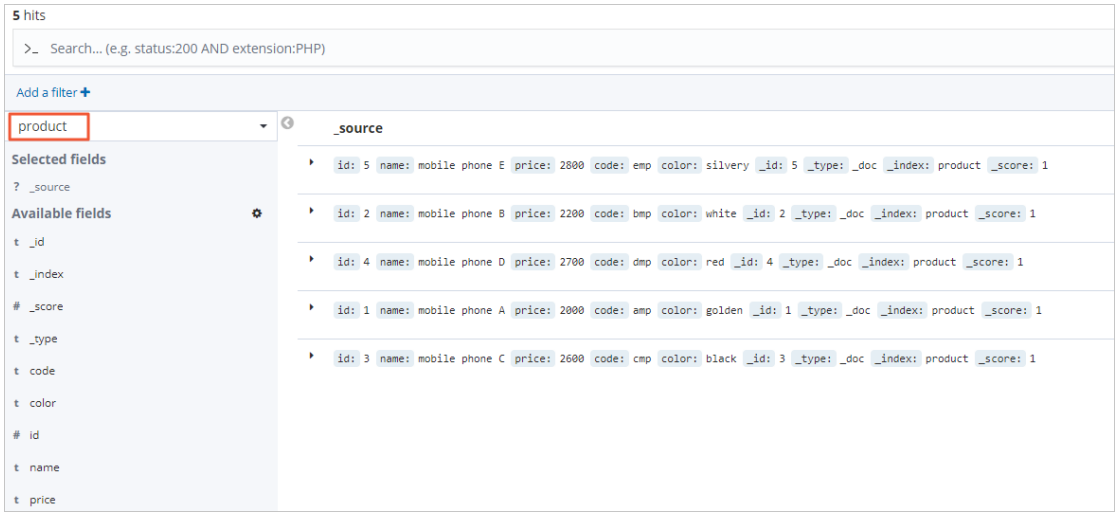
   ```
   GET /product/_doc/_search
   ```

   If the command is successfully executed, the result shown in the following figure is returned.



3. Perform operations in the console to query the synchronized data.

i. Create an index pattern for the destination index.
   a. In the left-side navigation pane, click **Management**.
   b. In the **Kibana** section, click **Index Patterns**.
   c. Click **Create index pattern**.
   d. In the **Create index pattern** section, enter a name in the **Index pattern** field.
   e. Click **Next step**.
   f. Click **Create index pattern**.
ii. In the left-side navigation pane, click **Discover**.
iii. Select the index pattern that you created to view the synchronized data.



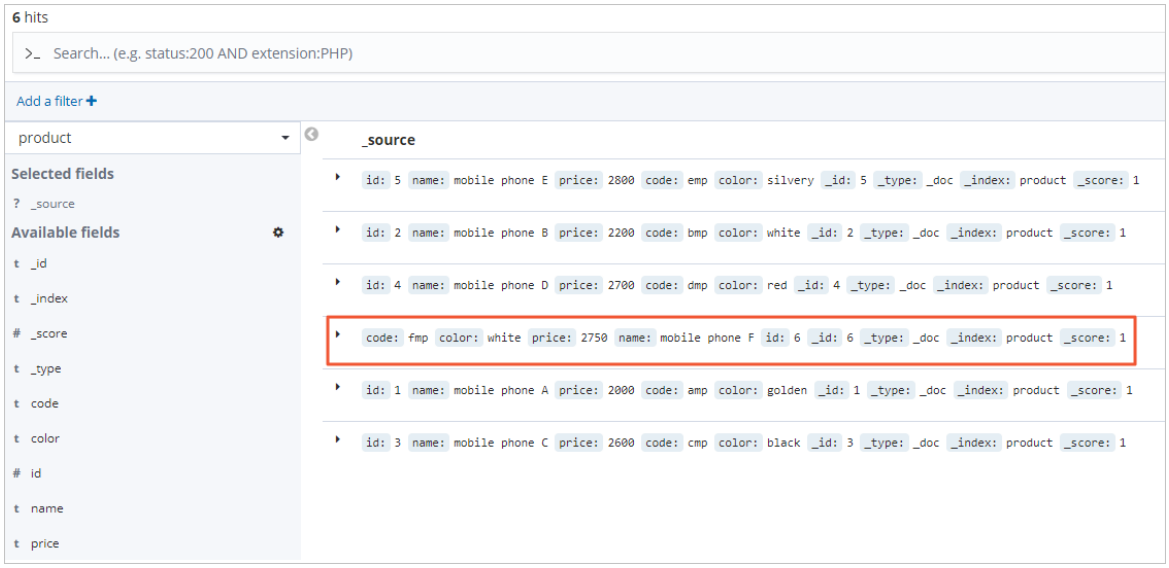
Step 3: Verify incremental data synchronization

1. Log on to the PolarDB console.
2. Execute the following statement to insert a data record into the PolarDB for MySQL database:

```
INSERT INTO `estest`.`product` (`id`,`name`,`price`,`code`,`color`) VALUES (6,'mobile p
hone F','2750','fmp','white');
```

3. Log on to the Kibana console.

   For more information, see Log on to the Kibana console.
4. In the left-side navigation pane, click **Discover**.
5. Select the index pattern that you created to view the synchronized incremental data.



Note After you delete or modify data in the source PolarDB for MySQL database, you can use the same method to verify data synchronization.

3.4. Use Monstache to synchronize data from a MongoDB database to an Alibaba Cloud Elasticsearch cluster in real time

Use Monstache to synchronize data from a MongoDB database to an Alibaba Cloud Elasticsearch cluster in real time

Alibaba Cloud Elasticsearch allows you to analyze semantics and displays analysis results in large charts for your business data that is stored in a MongoDB database. This topic describes how to use Monstache to synchronize data from a MongoDB database to an Elasticsearch cluster in real time. It also describes how to analyze the data and display analysis results.

Context

The example in this topic demonstrates how to parse and collect statistics on data of popular movies. You can perform the following operations:

- Use Monstache to quickly synchronize and subscribe to full or incremental data.
- Synchronize data from your MongoDB database to an Elasticsearch cluster of a later version in real time.
- Familiarize yourself with the common configuration parameters of Monstache.

Benefits

- The MongoDB database, Elasticsearch cluster, and Monstache are deployed in virtual private clouds (VPCs). Data can be securely transmitted over internal networks at a high speed.
- Monstache synchronizes and subscribes to data in real time based on MongoDB oplogs. It allows you

to synchronize data between MongoDB databases and later versions of Elasticsearch clusters. Monstache supports the change streams and aggregation pipelines of MongoDB. For more information about Monstache features, see Features.

- Monstache supports logical and physical deletion. You can also use it to delete databases and collections. It can ensure data consistency between the Elasticsearch cluster and MongoDB database in real time.

Procedure

1. Preparations

   Create an ApsaraDB for MongoDB instance, an Elasticsearch cluster, and an Elastic Compute Service (ECS) instance in the same VPC. The ECS instance is used to install Monstache.

   > **Notice** Make sure that the version of Monstache you installed is compatible with the versions of the ApsaraDB for MongoDB instance and Elasticsearch cluster. For more information about version compatibility of Monstache, see Monstache version.

2. Step 1: Build a Monstache environment

   Install Monstache on the ECS instance. Before you install Monstache, make sure that you have configured Go environment variables.

3. Step 2: Configure a real-time data synchronization task

   Modify information in the default configuration file of Monstache. The information includes the endpoints of the ApsaraDB for MongoDB instance and Elasticsearch cluster, the collections you want to synchronize, and the username and password of the Elasticsearch cluster. After you modify the preceding information, run Monstache to synchronize data from the ApsaraDB for MongoDB instance to the Elasticsearch cluster in real time.

4. Step 3: Verify the data synchronization result

   Add, update, or remove data in an ApsaraDB for MongoDB instance. Then, check whether data is synchronized in real time.

5. Step 4: Analyze data and display analysis results in the Kibana console

   In the Kibana console, analyze data and display analysis results in pie charts.

Preparations

1. Create an Elasticsearch cluster and enable the Auto Indexing feature for the cluster.

   For more information, see Create an Alibaba Cloud Elasticsearch cluster and Access and configure an Elasticsearch cluster. In this topic, an Elasticsearch V6.7.0 cluster of the Standard Edition is used.

2. Create an ApsaraDB for MongoDB instance and prepare test data.

   For more information, see Quick start. In this topic, an ApsaraDB for MongoDB replica set instance of V4.2 is used. The following figure shows part of the test data.

Notice The ApsaraDB for MongoDB instance you use must be a replica set instance or sharded cluster instance.

3. Create an ECS instance.

   For more information, see Create an instance by using the wizard. The ECS instance is used to install Monstache and must reside in the same VPC as the Elasticsearch cluster.

Step 1: Build a Monstache environment

1. Connect to the ECS instance.

   For more information, see 连接ECS实例.

2. Install SDK for Go and configure environment variables.

   Note Monstache-based data synchronization depends on the Go language. Therefore, before you install Monstache, you must prepare the Go environment on the ECS instance.

   i. Download and decompress the installation package of SDK for Go.

   ```
   wget https://dl.google.com/go/go1.14.4.linux-amd64.tar.gz
   tar -C /usr/local -xzf go1.14.4.linux-amd64.tar.gz
   ```

   ii. Configure environment variables.

   Run the `vim /etc/profile` command to open the configuration file for environment variables. Then, add the following content to the file. `GOPROXY` specifies a proxy for the modules of Alibaba Cloud SDK for Go.

   ```
   export GOROOT=/usr/local/go
   export GOPATH=/home/go/
   export PATH=$PATH:$GOROOT/bin:$GOPATH/bin
   export GOPROXY=https://mirrors.aliyun.com/goproxy/
   ```

   iii. Apply the environment variables.

   ```
   source /etc/profile
   ```

3. Install Monstache.
   i. Go to the installation path.
   ```
   cd /usr/local/
   ```
   ii. Download the installation package from the GitHub repository.
   ```
   git clone https://github.com/rwynn/monstache.git
   ```
   iii. Go to the *monstache* directory.
   ```
   cd monstache
   ```
   iv. Switch the version.

   In this topic, rel5 is used.
   ```
   git checkout rel5
   ```
   v. Install Monstache.
   ```
   go install
   ```
   vi. View the version of Monstache.
   ```
   monstache -v
   ```
   If the running of the command is successful, the following result is returned:
   ```
   5.5.5
   ```

Step 2: Configure a real-time data synchronization task

Monstache uses the TOML format for its configuration. In most cases, Monstache uses the default port to connect to the Elasticsearch cluster and ApsaraDB for MongoDB instance on your on-premises host and tracks the oplogs of the ApsaraDB for MongoDB instance. During the running of Monstache, all changes to data in your ApsaraDB for MongoDB database are synchronized to the Elasticsearch cluster.

In this topic, ApsaraDB for MongoDB and Alibaba Cloud Elasticsearch are used, and you need to specify the objects that you want to synchronize. Therefore, you must modify the default configuration file of Monstache. The objects that are used in this topic are the hotmovies and col collections in the mydb database. To modify the configuration file, perform the following steps:

1. Go to the Monstache installation directory, create a configuration file, and then edit the file.
   ```
   cd /usr/local/monstache/
   vim config.toml
   ```
2. Modify the configuration file.

   The following example demonstrates how to modify the configuration file. For more information, see Monstache Usage.
   ```
   # connection settings
   # connect to MongoDB using the following URL
   mongo-url = "mongodb://root:<your_mongodb_password>@dds-bp1aadcc629******.mongodb.rds.aliyuncs.com:3717"
   # connect to the Elasticsearch REST API at the following node URLs
   elasticsearch-urls = ["http://es-cn-mp91kzb8m00******.elasticsearch.aliyuncs.com:9200"]
   ```

```
# frequently required settings
# if you need to seed an index from a collection and not just listen and sync changes e
vents
# you can copy entire collections or views from MongoDB to Elasticsearch
direct-read-namespaces = ["mydb.hotmovies","mydb.col"]
# if you want to use MongoDB change streams instead of legacy oplog tailing use change-
stream-namespaces
# change streams require at least MongoDB API 3.6+
# if you have MongoDB 4+ you can listen for changes to an entire database or entire dep
loyment
# in this case you usually don't need regexes in your config to filter collections unle
ss you target the deployment.
# to listen to an entire db use only the database name.   For a deployment use an empty
string.
#change-stream-namespaces = ["mydb.col"]
# additional settings
# if you don't want to listen for changes to all collections in MongoDB but only a few
# e.g. only listen for inserts, updates, deletes, and drops from mydb.mycollection
# this setting does not initiate a copy, it is only a filter on the change event listen
er
#namespace-regex = '^mydb\.col$'
# compress requests to Elasticsearch
#gzip = true
# generate indexing statistics
#stats = true
# index statistics into Elasticsearch
#index-stats = true
# use the following PEM file for connections to MongoDB
#mongo-pem-file = "/path/to/mongoCert.pem"
# disable PEM validation
#mongo-validate-pem-file = false
# use the following user name for Elasticsearch basic auth
elasticsearch-user = "elastic"
# use the following password for Elasticsearch basic auth
elasticsearch-password = "<your_es_password>"
# use 4 go routines concurrently pushing documents to Elasticsearch
elasticsearch-max-conns = 4
# use the following PEM file to connections to Elasticsearch
#elasticsearch-pem-file = "/path/to/elasticCert.pem"
# validate connections to Elasticsearch
#elastic-validate-pem-file = true
# propogate dropped collections in MongoDB as index deletes in Elasticsearch
dropped-collections = true
# propogate dropped databases in MongoDB as index deletes in Elasticsearch
dropped-databases = true
# do not start processing at the beginning of the MongoDB oplog
# if you set the replay to true you may see version conflict messages
# in the log if you had synced previously. This just means that you are replaying old d
ocs which are already
# in Elasticsearch with a newer version. Elasticsearch is preventing the old docs from
overwriting new ones.
#replay = false
# resume processing from a timestamp saved in a previous run
resume = true
```

```
# do not validate that progress timestamps have been saved
#resume-write-unsafe = false
# override the name under which resume state is saved
#resume-name = "default"
# use a custom resume strategy (tokens) instead of the default strategy (timestamps)
# tokens work with MongoDB API 3.6+ while timestamps work only with MongoDB API 4.0+
resume-strategy = 0
# exclude documents whose namespace matches the following pattern
#namespace-exclude-regex = '^mydb\.ignorecollection$'
# turn on indexing of GridFS file content
#index-files = true
# turn on search result highlighting of GridFS content
#file-highlighting = true
# index GridFS files inserted into the following collections
#file-namespaces = ["users.fs.files"]
# print detailed information including request traces
verbose = true
# enable clustering mode
cluster-name = 'es-cn-mp91kzb8m00******'
# do not exit after full-sync, rather continue tailing the oplog
#exit-after-direct-reads = false
[[mapping]]
namespace = "mydb.hotmovies"
index = "hotmovies"
type = "movies"
[[mapping]]
namespace = "mydb.col"
index = "mydbcol"
type = "collection"
```

Parameter	Description
mongo-url	The connection string of the primary node in the ApsaraDB for MongoDB instance. You can obtain the connection string from the details page of the ApsaraDB for MongoDB instance. Before you obtain the connection string, add the private IP address of the ECS instance where Monstache is installed to the whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist for a sharded cluster instance.
elasticsearch-urls	The URL that is used to access the Elasticsearch cluster. Specify the URL in the format of `http://<Internal endpoint of the Elasticsearch cluster>:9200`. You can obtain the internal endpoint from the details page of the cluster. For more information, see View the basic information of a cluster.
direct-read-namespaces	The collections that you want to synchronize. For more information, see direct-read-namespaces. The collections used in this topic are hotmovies and col in the mydb database.
change-stream-namespaces	You must specify this parameter if you want to use the change streams of the ApsaraDB for MongoDB instance. If you specify this parameter, oplog tracking becomes invalid. For more information, see change-stream-namespaces.

Parameter	Description
namespace-regex	The regular expression that is used to specify the collections you want to monitor. After you specify a regular expression, the system can monitor changes to data in collections that match the regular expression.
elasticsearch-user	The username that is used to access the Elasticsearch cluster. Default value: elastic. **Notice** To ensure system security, we recommend that you do not use the elastic username. You can use a custom username instead. Before you use a custom username, you must create a role for it and grant the required permissions to the role. For more information, see Use the RBAC mechanism provided by Elasticsearch X-Pack to implement access control.
elasticsearch-password	The password of the elastic account. The password that corresponds to the elastic username is specified when you create your Elasticsearch cluster. If you forget the password, you can reset it. For more information about the precautions and procedures for resetting a password, see Reset the access password for an Elasticsearch cluster.
elasticsearch-max-conns	The number of threads that are used to connect to the Elasticsearch cluster. The default value is 4. This value indicates that four Go threads are used at a time to synchronize data to the Elasticsearch cluster.
dropped-collections	The default value is true. This value indicates that Monstache deletes a mapped index in the Elasticsearch cluster if a collection in your ApsaraDB for MongoDB database is deleted.
dropped-databases	The default value is true. This value indicates that Monstache deletes mapped indexes in the Elasticsearch cluster if your ApsaraDB for MongoDB database is deleted.
resume	The default value is false. If you set this parameter to true, Monstache writes the timestamps of operations that are synchronized to the Elasticsearch cluster to the monstache.monstache collection. If Monstache fails, you can use the timestamps to resume the synchronization task. This avoids data loss. If Monstache starts with the cluster-name parameter specified, the resume parameter is automatically set to true. For more information, see resume.
resume-strategy	The resuming policy. This parameter is valid only when resume is set to true. For more information, see resume-strategy.
verbose	The default value is false. This value indicates that log debugging is disabled.

Parameter	Description
cluster-name	The name of the cluster. If you specify this parameter, Monstache runs in high availability mode. Processes that have the same cluster name cooperate with each other. For more information, see cluster-name.
mapping	The mapping of the index in the Elasticsearch cluster. By default, when data is synchronized from a MongoDB database to an Elasticsearch cluster, the index name is automatically mapped to `Database name.Collection name`. You can change this index name by setting this parameter. For more information, see Index Mapping.

> **Note** Monstache supports a large number of parameters. The preceding table describes only some parameters that are used for real-time data synchronization. For more information about how to configure parameters that are used for complex data synchronization, see Monstache config and Advanced.

3. Run Monstache.

```
monstache -f config.toml
```

> **Note** The -f parameter is used to explicitly run Monstache. In this case, the system will log all debugging operations, including request tracking for the Elasticsearch cluster.

Step 3: Verify the data synchronization result

1. Log on to the Data Management (DMS) console of the ApsaraDB for MongoDB instance and the Kibana console of the Elasticsearch cluster. Query the number of documents before the synchronization and that after the synchronization.

   > **Note**
   > - For more information about how to log on to the DMS console, see Connect to an ApsaraDB for MongoDB replica set instance by using DMS.
   > - For more information about how to log on to the Kibana console, see Log on to the Kibana console.

   - MongoDB

     ```
     db.hotmovies.find().count()
     ```

     If the running of the command is successful, the following result is returned:

     ```
     [
     10000
     ]
     ```

   - Elasticsearch cluster

```
GET hotmovies/_count
```

If the running of the command is successful, the following result is returned:

```
{
  "count" : 10000,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  }
}
```

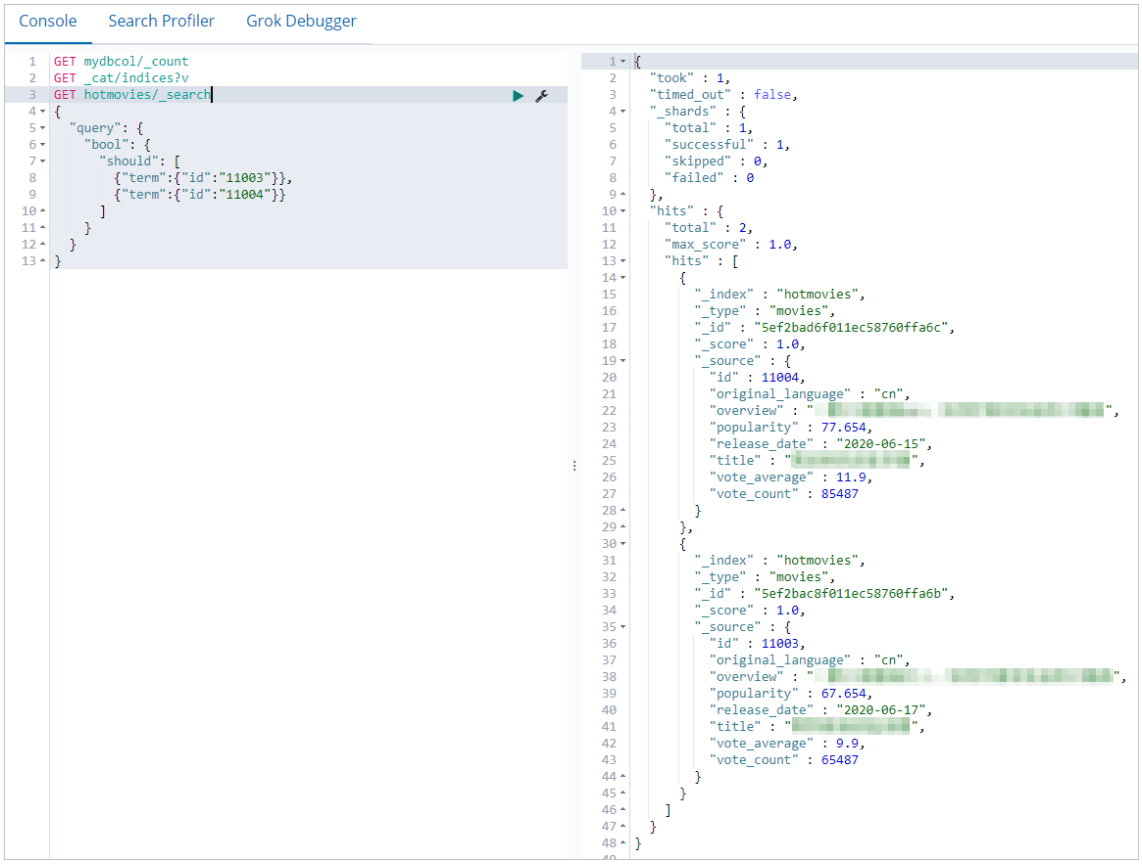
2. Insert data into the ApsaraDB for MongoDB database and check whether the data is synchronized to the Elasticsearch cluster.
   - MongoDB

```
db.hotmovies.insert({id: 11003,title: "Beauty",overview: "How a group of IT women with high IQ become outstanding",original_language:"cn",release_date:"2020-06-17",popularity:67.654,vote_count:65487,vote_average:9.9})
```

```
db.hotmovies.insert({id: 11004,title: "Heroic Programmers",overview: "How a group of IT men with high IQ become outstanding",original_language:"cn",release_date:"2020-06-15",popularity:77.654,vote_count:85487,vote_average:11.9})
```

   - Elasticsearch cluster

```
GET hotmovies/_search
{
  "query": {
    "bool": {
      "should": [
        {"term":{"id":"11003"}},
        {"term":{"id":"11004"}}
      ]
    }
  }
}
```

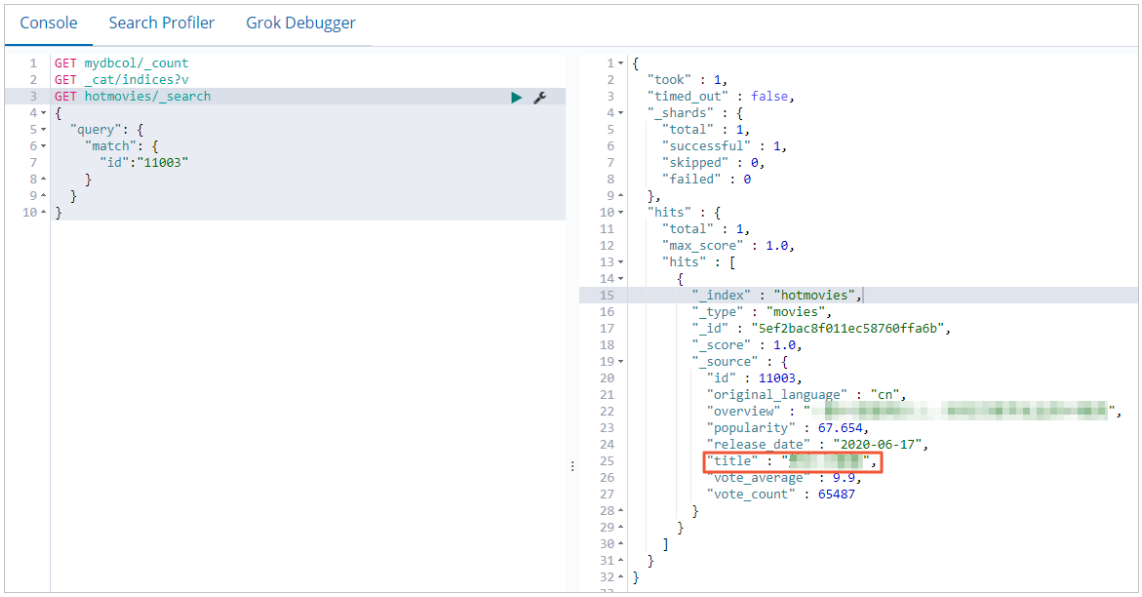


3. Update data in the ApsaraDB for MongoDB database. Then, check whether the data in the Elasticsearch cluster is also updated.
   - MongoDB

     ```
     db.hotmovies.update({'title':'Beauty'},{$set:{'title':'Beautiful Programmers'}})
     ```

   - Elasticsearch cluster

     ```
     GET hotmovies/_search
     {
       "query": {
         "match": {
           "id":"11003"
         }
       }
     }
     ```



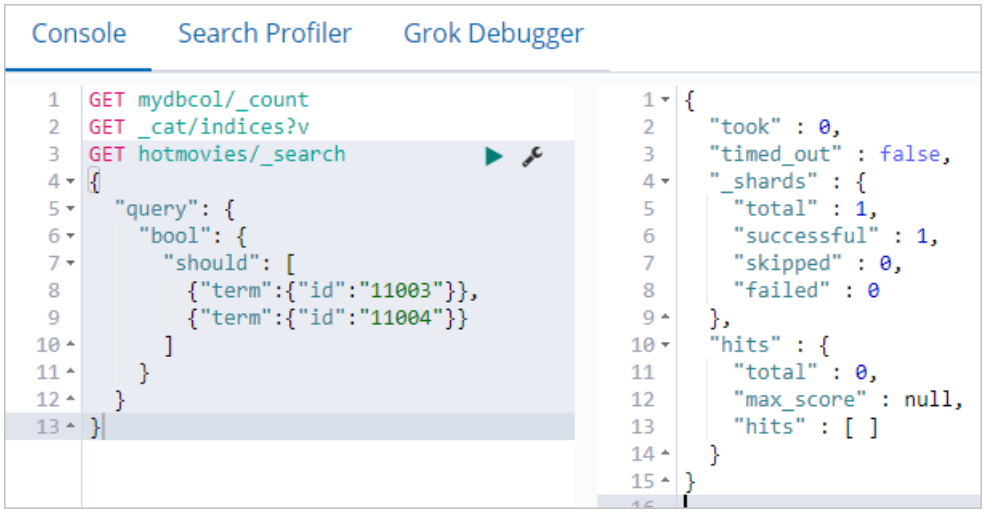
4. Remove data from the ApsaraDB for MongoDB database. Then, check whether the data is also removed from the Elasticsearch cluster.
   - MongoDB

     ```
     db.hotmovies.remove({id: 11003})
     ```

     ```
     db.hotmovies.remove({id: 11004})
     ```

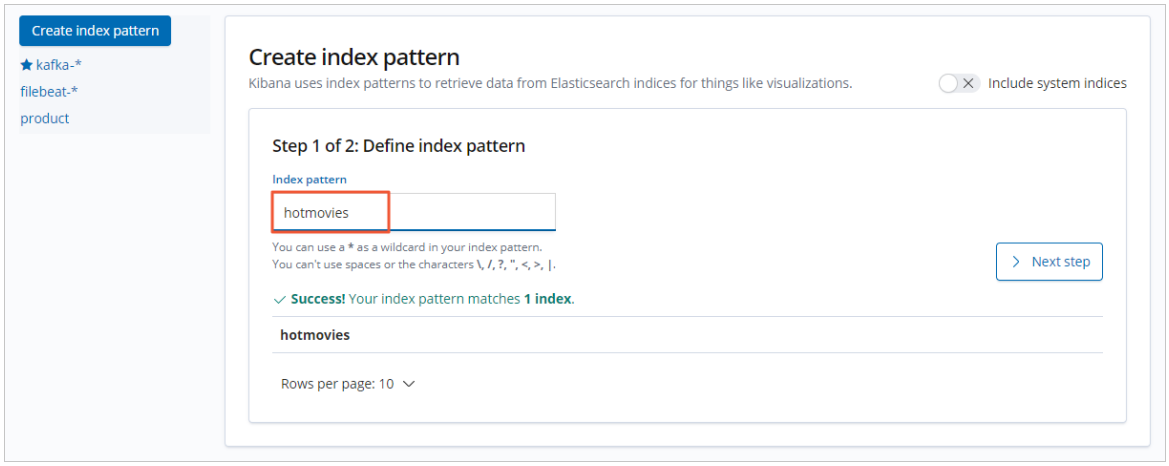
   - Elasticsearch cluster

     ```
     GET hotmovies/_search
     {
       "query": {
         "bool": {
           "should": [
             {"term":{"id":"11003"}},
             {"term":{"id":"11004"}}
           ]
         }
       }
     }
     ```



Step 4: Analyze data and display analysis results in the Kibana console

1.
2. Create an index pattern.

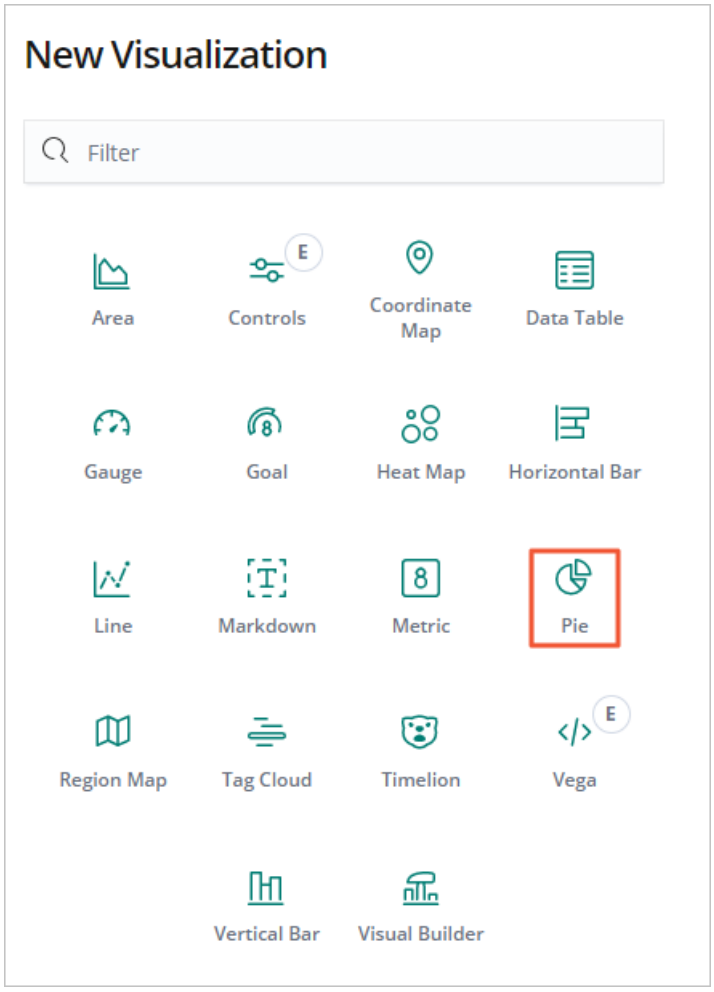


   i.
   ii.
   iii.
   iv.
   v. Specify **Time Filter field name**. In this example, Time Filter field name is set to **I don't want to use the Time Filter**.
   vi.
3. Configure a chart.

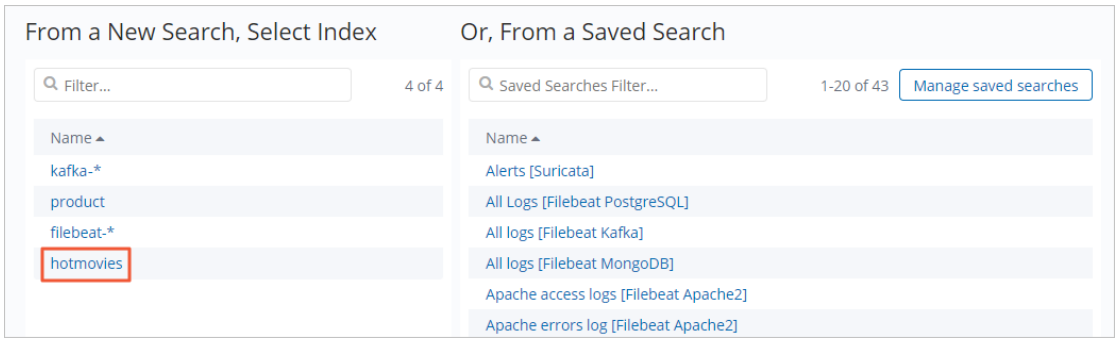
   The following example demonstrates how to configure a pie chart for the top 10 popular movies:

   i. In the left-side navigation pane, click **Visualize**.
   ii. Click **+** next to the search box.

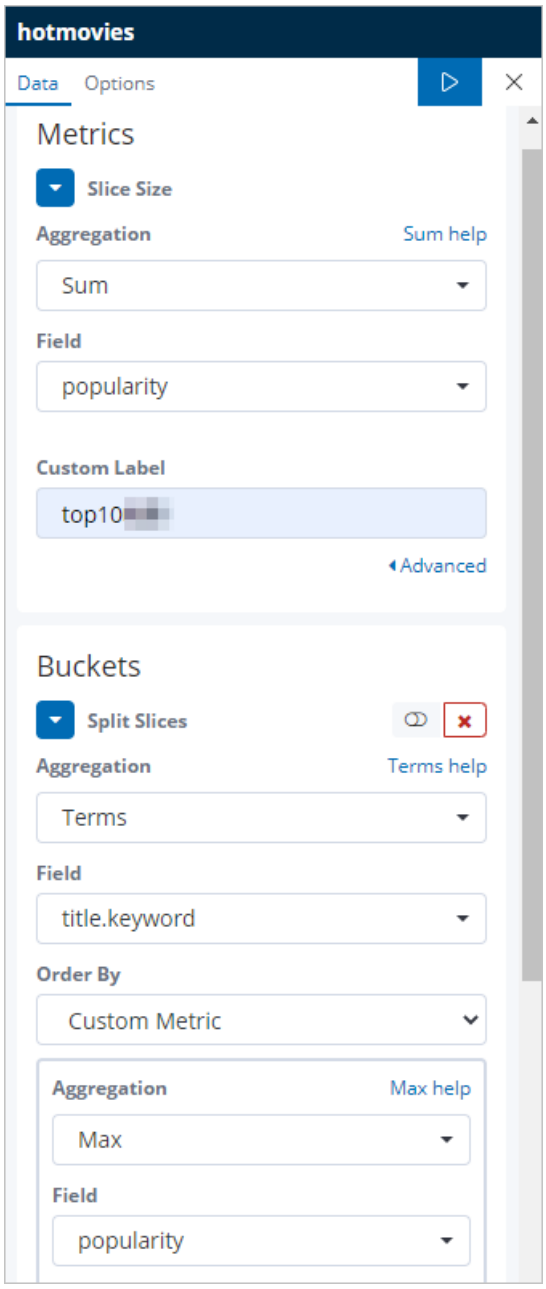
iii. In the **New Visualization** dialog box, click **Pie**.



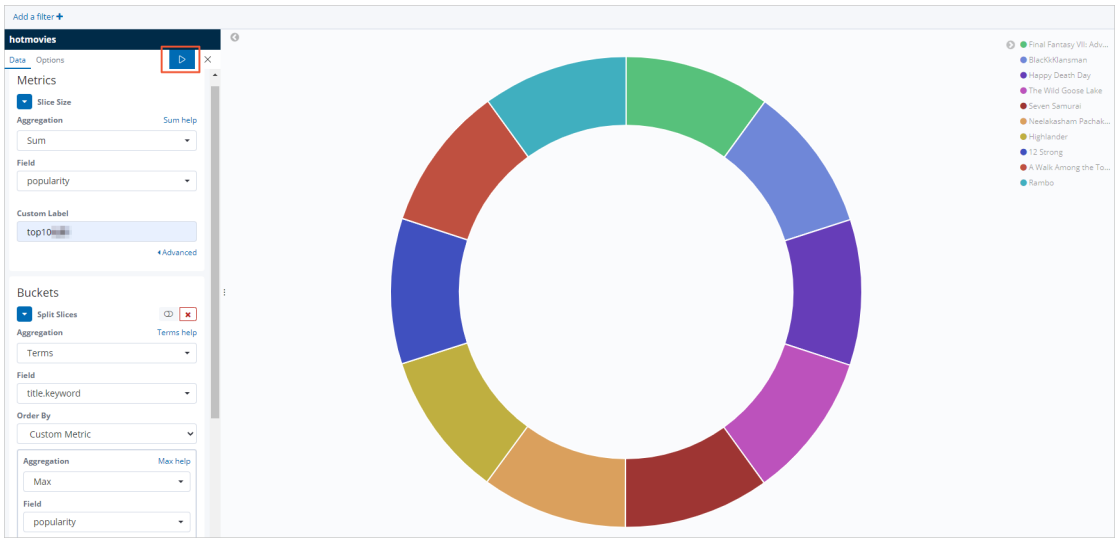
iv. Click the **hotmovies** index pattern.



v. Configure parameters in the **Metrics** and **Buckets** sections based on the following figure.



vi. Click the icon to apply the configurations and display data.



FAQ

Problem description: After I enable the high availability and high concurrency features for my Elasticsearch cluster, data loss occurs.

Solution: Check whether the cluster is normal. If the cluster is normal, you must check whether Monstache can normally provide services. For more information, visit the official Monstache website. If the cluster is abnormal, you must troubleshoot the issue from the cluster and reduce the number of concurrencies. For more information about the frequently asked questions about Elasticsearch clusters and the answers to these questions, see References. If the issue cannot be resolved, submit a ticket for consultation.

4.Big data synchronization

4.1. Use DataWorks to synchronize data from MaxCompute to an Alibaba Cloud Elasticsearch cluster

Synchronize data from MaxCompute to Alibaba Cloud Elasticsearch

Alibaba Cloud provides a variety of cloud storage and database services. To search for and analyze the data stored in these services, you can use the Data Integration feature of DataWorks to collect offline data at intervals of at least five minutes. Then, synchronize the data to an Alibaba Cloud Elasticsearch cluster. This topic describes how to synchronize data from MaxCompute to an Alibaba Cloud Elasticsearch cluster.

Context

Alibaba Cloud Elasticsearch supports the following offline data sources:

- Alibaba Cloud databases: ApsaraDB RDS for MySQL, ApsaraDB RDS for PostgreSQL, ApsaraDB RDS for SQL Server, ApsaraDB RDS for PPAS, ApsaraDB for MongoDB, and ApsaraDB for HBase
- Distributed Relational Database Service (DRDS)
- MaxCompute
- Object Storage Service (OSS)
- Tablestore
- User-created databases: HDFS, Oracle, FTP, DB2, MySQL, PostgreSQL, SQL Server, PPAS, MongoDB, and HBase

Procedure

1. Preparations

   Create a DataWorks workspace, activate MaxCompute, prepare the MaxCompute data source, and create an Alibaba Cloud Elasticsearch cluster.

2. Step 1: Purchase and create an exclusive resource group

   Purchase and create an exclusive resource group for data integration. Bind the exclusive resource group to a virtual private cloud (VPC) and the created workspace. Exclusive resource groups transmit data in a fast and stable manner.

3. Step 2: Add data sources

   Connect MaxCompute and Elasticsearch data sources to Data Integration.

4. Step 3: Create and run a data synchronization task

   Configure a data synchronization script to import the data synchronized by Data Integration into the Elasticsearch cluster. The exclusive resource group is registered with Data Integration as a resource to run tasks. This resource group retrieves data from data sources and runs the task of writing data to the Elasticsearch cluster. The task is issued by Data Integration.

5. Step 4: View synchronization results

   In the Kibana console, view the synchronized data and search for data based on specific

conditions.

Preparations

1. Create a DataWorks workspace. Select MaxCompute as the computing engine.

   For more information, see Create a MaxCompute project.

2. Create a table in MaxCompute and import data into the table.

   For more information, see Create tables and Import data to tables.

   The following figures show the table schema and a part of the table data.

   Table schema

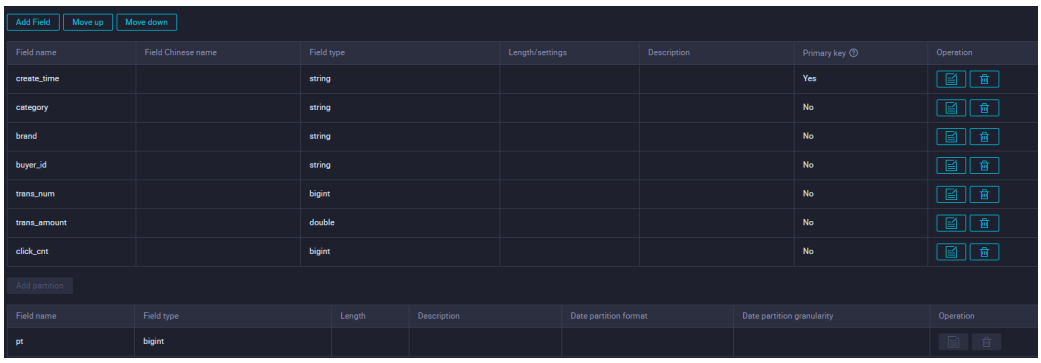


   Table data

	A	B	C	D	E	F	G	H
1	create_time	category	brand	buyer_id	trans_num	trans_amount	click_cnt	pt
2	2020/6/1 8:00		A	user1	10	150.0	50	1
3	2020/6/2 8:00		B	user2	11	180.0	60	1
4	2020/6/3 8:00		D	user3	12	150.0	50	1
5	2020/6/4 8:00		A	user4	14	150.0	50	1
6	2020/6/5 8:00		C	user5	10	1500.0	90	1
7	2020/6/6 8:00		F	user6	10	190.0	69	1
8	2020/6/7 8:00		E	user7	88	150.0	80	1
9	2020/6/8 8:00		A	user8	10	180.0	67	1
10	2020/6/9 8:00		G	user9	18	3500.0	50	1
11	2020/6/10 8:00		F	user10	10	150.0	45	1
12	2020/6/11 8:00		G	user11	22	4500.0	70	1
13	2020/6/12 8:00		A	user12	10	150.0	55	1
14	2020/6/13 8:00		B	user13	20	220.0	110	1

> **Note** The provided data is only for tests. You can migrate data from Hadoop to MaxCompute and synchronize the data to your Elasticsearch cluster. For more information, see Synchronize data from Hadoop to MaxCompute.

3. Create an Elasticsearch cluster and enable the Auto Indexing feature for the cluster.

   For more information, see Create an Alibaba Cloud Elasticsearch cluster and Access and configure an Elasticsearch cluster. The Elasticsearch cluster must reside in the same region as the DataWorks workspace you created.

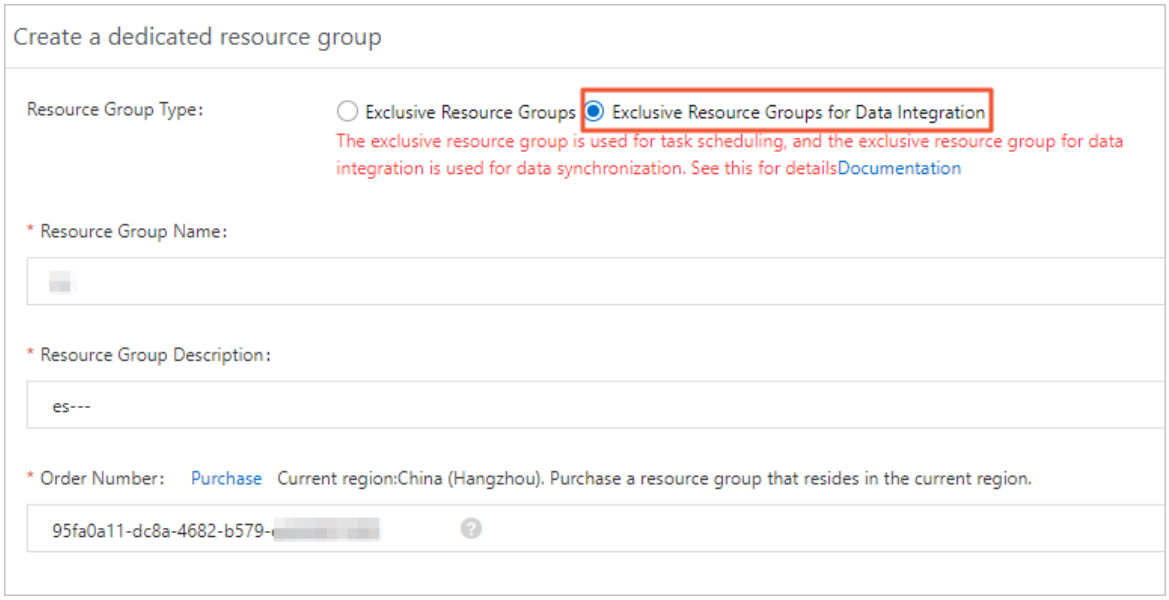
Step 1: Purchase and create an exclusive resource group

1. Log on to the DataWorks console.
2. In the top navigation bar, select a region. In the left-side navigation pane, click **Resource Groups**.
3. Purchase exclusive resources for data integration. For more information, see Purchase exclusive resources for Data Integration.

> **Notice** The exclusive resources for data integration must reside in the same region as the DataWorks workspace you created.

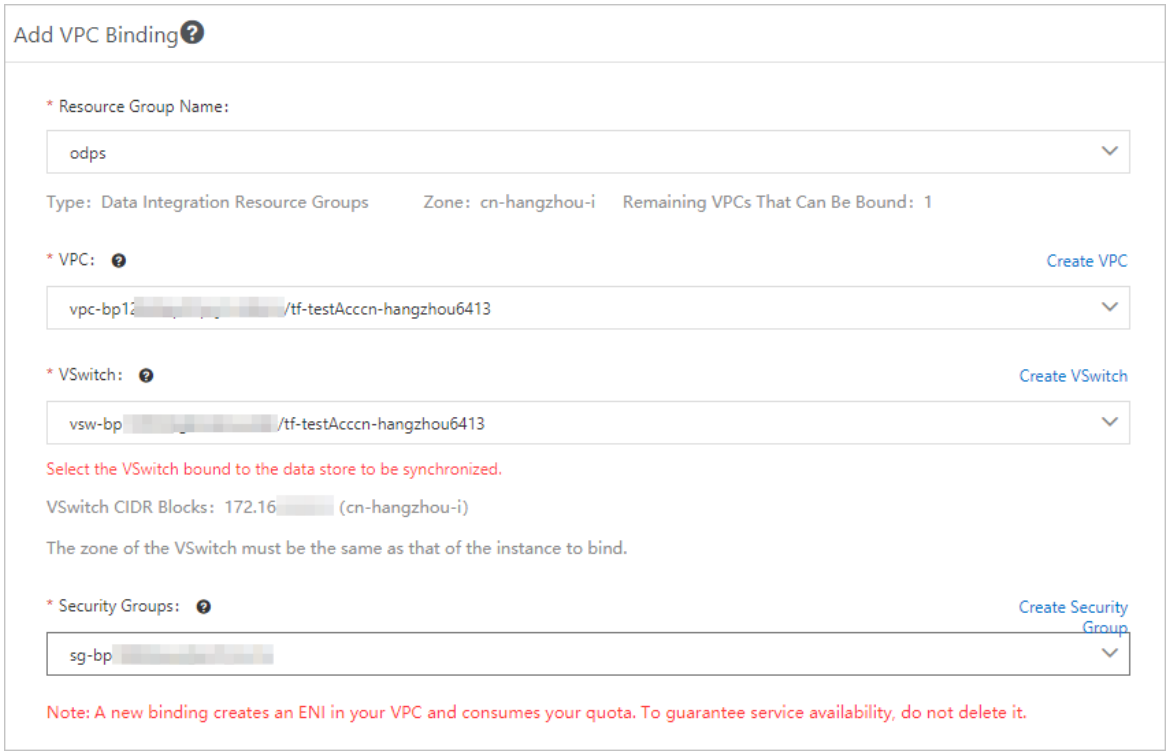
4. Create an exclusive resource group for data integration. For more information, see Create an exclusive resource group for Data Integration.

   The following figure shows the configuration used in this example. **Resource Group Type** is set to **Exclusive Resource Groups for Data Integration**.

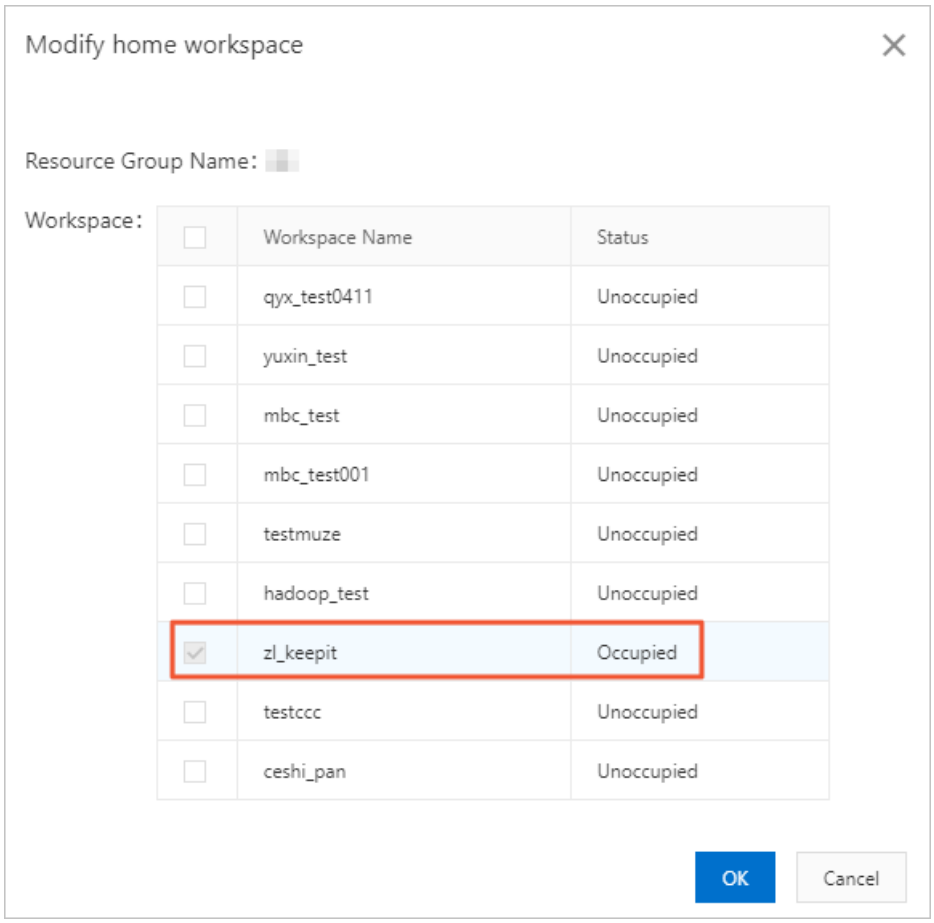


5. Find the created exclusive resource group and click **Add VPC Binding** in the Actions column to bind the exclusive resource group to a VPC. For more information, see Configure network settings.

   Exclusive resources are deployed in VPCs managed by DataWorks. DataWorks can synchronize data to an Elasticsearch cluster only after DataWorks and the cluster are connected to the same VPC. Therefore, when you bind the exclusive resource group to a VPC, you must select the **VPC** and **VSwitch** to which your Elasticsearch cluster belongs.

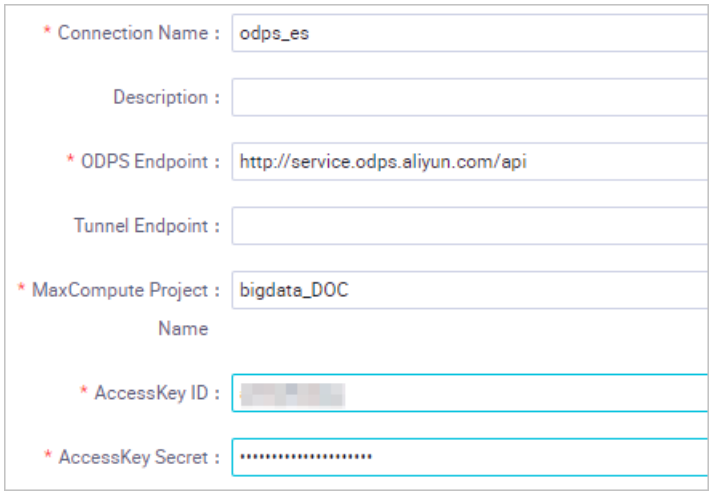


6. Click **Change Workspace** in the Actions column that corresponds to the exclusive resource group to bind it to the DataWorks workspace you created. For more information, see Associate an exclusive resource group with a workspace.



Step 2: Add data sources

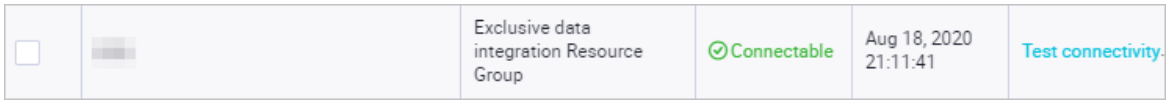
1. Go to the **Data Integration** page.
   i. In the left-side navigation pane of the DataWorks console, click **Workspaces**.
   ii. Find the workspace you created and click **Data Integration** in the **Actions** column.
2. In the left-side navigation pane of the Data Integration page, click **Connection**.
3. In the left-side navigation pane of the page that appears, click **Data Source**. Then, click **New data source** in the upper-right corner.
4. In the Big Data Storage section of the **Add data source** dialog box, click **MaxCompute (ODPS)**. In the **Add MaxCompute (ODPS) data source** dialog box, configure the parameters.



Parameter	Description
ODPS Endpoint	The endpoint of MaxCompute, which varies in different regions. For more information, see Endpoints.
ODPS project name	To obtain the project name, log on to the DataWorks console. In the left-side navigation pane, click **MaxCompute** below **Compute Engines**.
AccessKey ID	To obtain the AccessKey ID, move the pointer over your profile picture and click **AccessKey Management**.
AccessKey Secret	To obtain the AccessKey secret, move the pointer over your profile picture and click **AccessKey Management**.

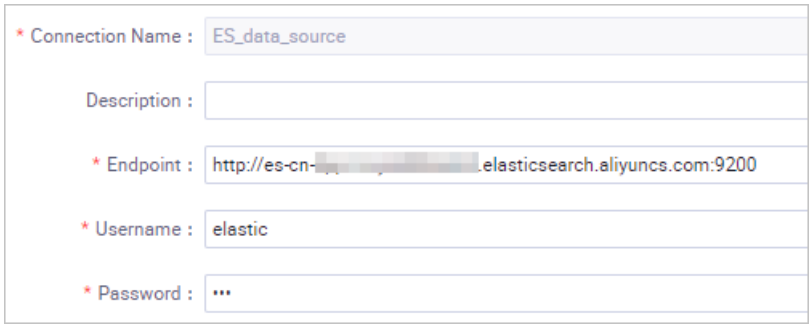
> **Note** Configure the parameters that are not listed in the preceding table as required or use their default values.

After parameters are configured, you can test the connectivity to the exclusive resource group. If the connectivity test is passed, **Connectable** appears in the **Connectivity status** column.



5. Click **Complete**.

6. Add an Elasticsearch data source in the same way.



Parameter	Description
Endpoint	The URL that is used to access the Elasticsearch cluster. Specify the URL in the following format: `http://<Internal or public endpoint of the Elasticsearch cluster>:9200`. You can obtain the endpoint from the Basic Information page of the cluster. For more information, see View the basic information of a cluster. **Notice** If you use the public endpoint of the cluster, add the elastic IP address (EIP) of the exclusive resource group to the public IP address whitelist of the cluster. For more information, see Configure a public or private IP address whitelist for an Elasticsearch cluster and Add the EIP or CIDR block of an exclusive resource group for Data Integration to the whitelist of a data source.
User name	The username that is used to access the Elasticsearch cluster. The default username is elastic.
Password	The password that is used to access the Elasticsearch cluster. The password of the elastic account is specified when you create the cluster. If you forget the password, you can reset it. For more information about the procedure and precautions for resetting a password, see Reset the access password for an Elasticsearch cluster.

Note Configure the parameters that are not listed in the preceding table as required.

Step 3: Create and run a data synchronization task

1.
2.
3.
4. In the upper part of the page, click the icon.
5. In the Tips message, click OK. Then, configure the data synchronization script.

   For more information, see Create a synchronization node by using the code editor.

> **Note** You can also click the icon in the upper part of the page to import a script configuration template. Then, modify the template as required.

The following code provides a sample script:

```
{
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "0"
        },
        "speed": {
            "concurrent": 1,
            "throttle": false
        }
    },
    "steps": [
        {
            "category": "reader",
            "name": "Reader",
            "parameter": {
                "column": [
                    "create_time",
                    "category",
                    "brand",
                    "buyer_id",
                    "trans_num",
                    "trans_amount",
                    "click_cnt"
                ],
                "datasource": "odps_es",
                "partition": "pt=1",
                "table": "hive_doc_good_sale"
            },
            "stepType": "odps"
        },
        {
            "category": "writer",
            "name": "Writer",
            "parameter": {
                "batchSize": 1000,
                "cleanup": true,
                "column": [
                    {
                        "name": "create_time",
```

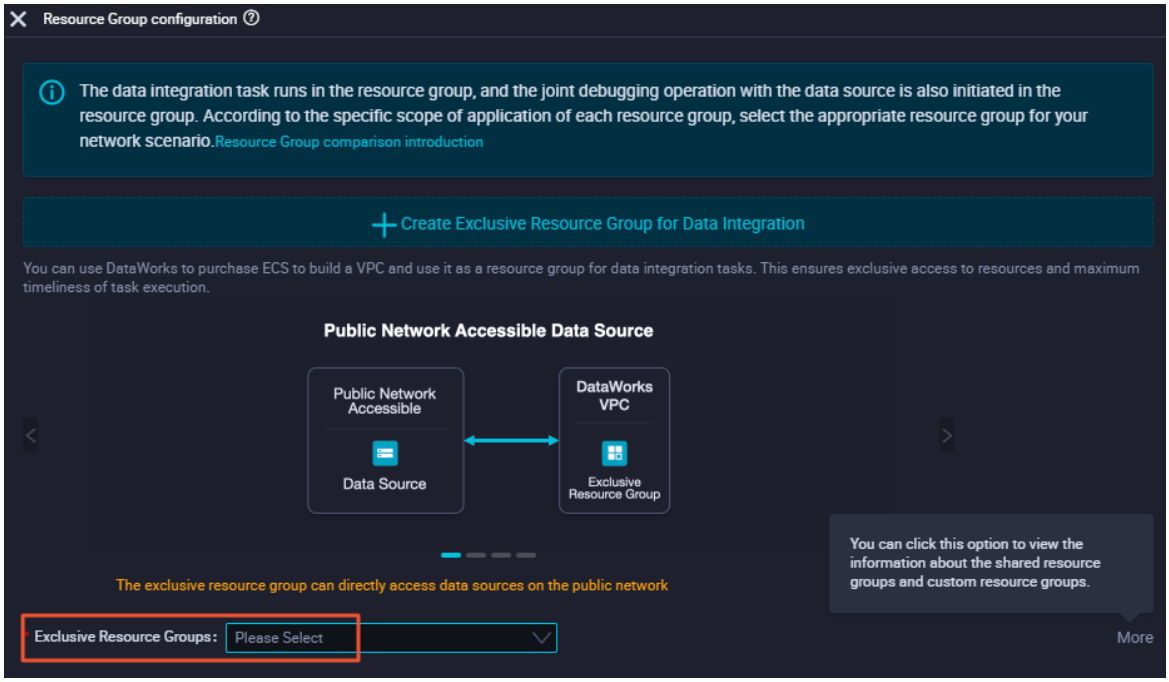
```
                        "type": "id"
                    },
                    {
                        "name": "category",
                        "type": "text"
                    },
                    {
                        "name": "brand",
                        "type": "text"
                    },
                    {
                        "name": "buyer_id",
                        "type": "text"
                    },
                    {
                        "name": "trans_num",
                        "type": "integer"
                    },
                    {
                        "name": "trans_amount",
                        "type": "double"
                    },
                    {
                        "name": "click_cnt",
                        "type": "integer"
                    }
                ],
                "datasource": "es_test",
                "discovery": false,
                "index": "odps_index",
                "indexType": "_doc",
                "splitter": ","
            },
            "stepType": "elasticsearch"
        }
    ],
    "type": "job",
    "version": "2.0"
}
```

The preceding script includes three parts.

Part	Description
`setting`	Used to configure parameters related to packet loss and the maximum concurrency during synchronization. **Note** If the volume of data you want to synchronize is large, you can increase the maximum concurrency.

Part	Description
`Reader`	Used to configure MaxCompute as the reader. If your MaxCompute table is a partitioned table, you must configure partition information by using the `partition` field. For more information, see MaxCompute Reader. The partition information in this example is `pt=1`. **Note** If the volume of data you want to synchronize is large, you can split the data into partitions and synchronize the data by partition.
`Writer`	Used to configure the Elasticsearch cluster as the writer. For more information, see Elasticsearch Writer. ○ `index`: the name of the destination index. ○ `indexType`: the type of the destination index. The index type of Elasticsearch clusters of V7.0 or later must be `_doc`.

6.
7. Configure the resource group that is used to execute the synchronization task.



   i. Click the **Data integration resource group configuration** tab on the right side of the page.
   ii. Set **Programme** to **Exclusive data integration Resource Group**.
   iii. Set **Exclusive data integration Resource Group** to the exclusive resource group you created.
8.
9.

Step 4: View synchronization results

1. Log on to the Kibana console of the Elasticsearch cluster.

   For more information, see Log on to the Kibana console.

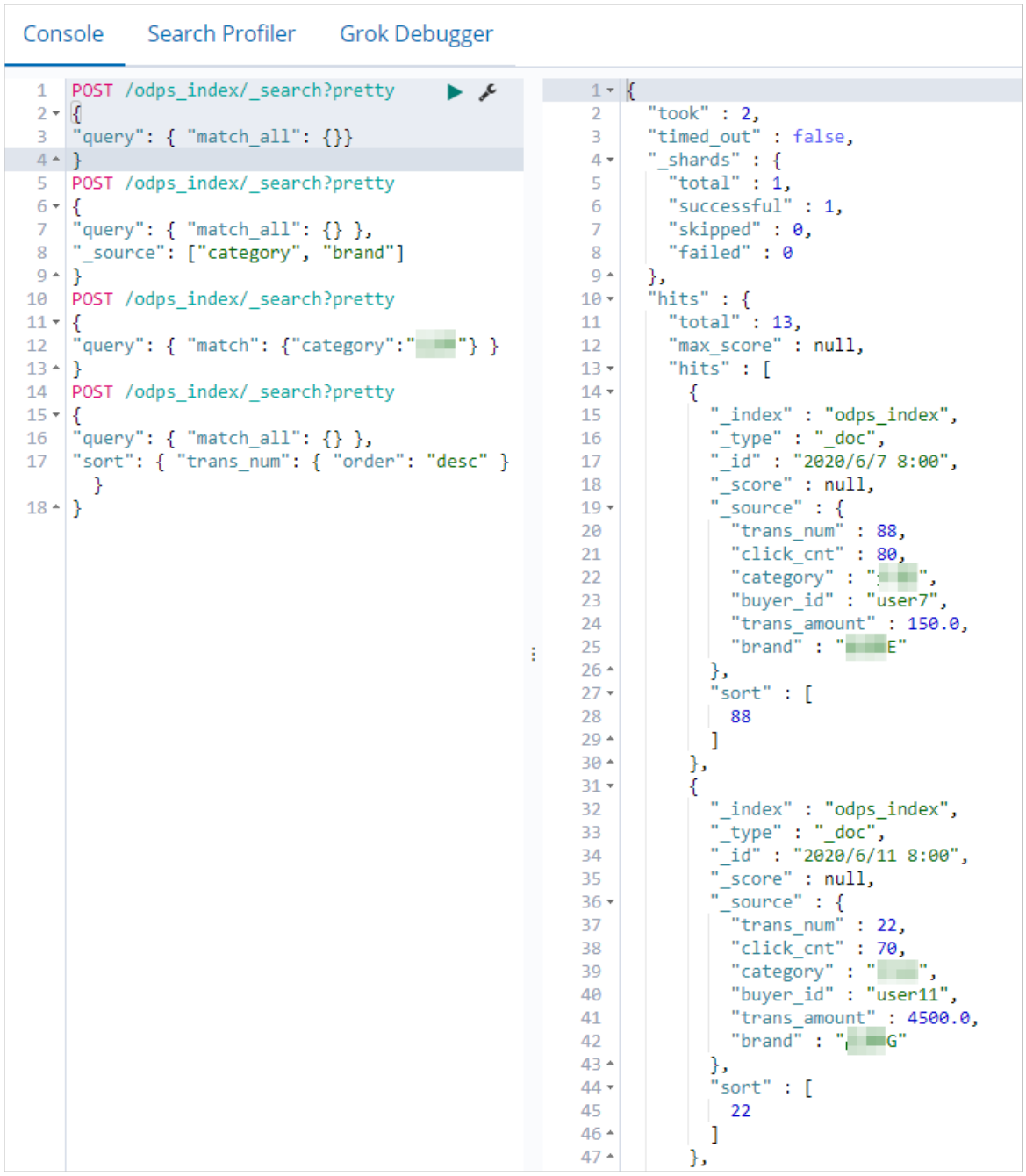
2. In the left-side navigation pane, click **Dev Tools**.

3. On the **Console** tab of the page that appears, run the following command to query the synchronized data:

   ```
   POST /odps_index/_search?pretty
   {
   "query": { "match_all": {}}
   }
   ```

   > **Note** `odps_index` is the value that you specified for the `index` field in the data synchronization script.

   If the data is synchronized, the result shown in the following figure is returned.



4. Run the following command to query the `category` and `brand` fields in the data:

```
POST /odps_index/_search?pretty
{
"query": { "match_all": {} },
"_source": ["category", "brand"]
}
```

5. Run the following command to query data entries where the value of the `category` field is `fresh`:

```
POST /odps_index/_search?pretty
{
"query": { "match": {"category":"fresh"} }
}
```

6. Run the following command to sort the data based on the `trans_num` field:

```
POST /odps_index/_search?pretty
{
"query": { "match_all": {} },
"sort": { "trans_num": { "order": "desc" } }
}
```

For more information, see open source Elastic documentation.

4.2. Use Realtime Compute to process and synchronize data to an Alibaba Cloud Elasticsearch cluster

Use Realtime Compute for Apache Flink to synchronize data to Elasticsearch

If you want to build a log retrieval system, you can use Alibaba Cloud Realtime Compute for Apache Flink to compute log data and import the processed data into Alibaba Cloud Elasticsearch for searches. This topic uses log data in Log Service to describe the detailed procedure.

Prerequisites

You have completed the following operations:

- Activate Realtime Compute and create a project.

  For more information, see Activate Realtime Compute for Apache Flink and create a project.
- Create an Elasticsearch cluster.

  For more information, see Create an Alibaba Cloud Elasticsearch cluster.
- Activate Log Service, and create a project and a Logstore.

  For more information, see Quick start, Create a project, and Create a Logstore.

Context

Realtime Compute for Apache Flink is a Flink-based service provided by Alibaba Cloud. It supports various input and output systems, such as Kafka and Elasticsearch. You can use Realtime Compute for Apache Flink and Elasticsearch to retrieve logs.

Realtime Compute for Apache Flink processes logs in Kafka or Log Service by using simple or complex Flink SQL statements. Then, it imports the processed logs into an Elasticsearch cluster as source data for searches. The computing capabilities of Realtime Compute for Apache Flink and the search capabilities of Elasticsearch allow you to process and search for data in real time. This help you transform your business into real-time services.

Realtime Compute for Apache Flink provides a simple way to interact with Elasticsearch. For example, logs or data records are imported into Log Service and must be processed before they are imported into an Elasticsearch cluster. The following figure shows the data consumption pipeline.

Procedure

1. Log on to the .
2. Create a Realtime Compute job.

   For more information, see Create a job.
3. Write Flink SQL statements.

i. Create a source table for Log Service.

```
create table sls_stream(
  a int,
  b int,
  c VARCHAR
)
WITH (
  type ='sls',
  endPoint ='<yourEndpoint>',
  accessId ='<yourAccessId>',
  accessKey ='<yourAccessKey>',
  startTime = '<yourStartTime>',
  project ='<yourProjectName>',
  logStore ='<yourLogStoreName>',
  consumerGroup ='<yourConsumerGroupName>'
);
```

The following table describes the parameters in the WITH part.

Parameter	Description
endPoint	The URL that is used to access projects and logs in Log Service. For more information, see Endpoints. For example, the URL that is used to access Log Service in the China (Hangzhou) region is **http://cn-hangzhou.log.aliyuncs.com**. Make sure that the URL starts with **http://**.
accessId	The AccessKey ID of your Alibaba Cloud account.
accessKey	The AccessKey secret of your Alibaba Cloud account.
startTime	The time when logs start to be consumed. When you run a Realtime Compute for Apache Flink job, specify a time point that is later than the start time specified by this parameter.
project	The name of the Log Service project.
logStore	The name of the Logstore in the project.
consumerGroup	The name of the Log Service consumer group.

For more information about other parameters in the WITH part, see Create a Log Service source table.

ii. Create an Elasticsearch result table.

> **Notice**
>
> - Elasticsearch result tables are supported in Realtime Compute V3.2.2 and later. When you create a Realtime Compute for Apache Flink job, select a valid version.
> - Elasticsearch result tables are based on RESTful APIs and are compatible with all Elasticsearch versions.

```
CREATE TABLE es_stream_sink(
  a int,
  cnt BIGINT,
  PRIMARY KEY(a)
)
WITH(
  type ='elasticsearch',
  endPoint = 'http://<instanceid>.public.elasticsearch.aliyuncs.com:<port>',
  accessId = '<yourAccessId>',
  accessKey = '<yourAccessSecret>',
  index = '<yourIndex>',
  typeName = '<yourTypeName>'
);
```

The following table describes the parameters in the WITH part.

Parameter	Description
endPoint	The URL that is used to access the Elasticsearch cluster over the Internet. Specify the URL in the format of http://<instanceid>.public.elasticsearch.aliyuncs.com:9200. You can obtain the public endpoint of the cluster from the Basic Information page of the cluster. For more information, see View the basic information of a cluster.
accessId	The username that is used to access the Elasticsearch cluster. The default username is elastic.
accessKey	The password that is used to access the Elasticsearch cluster. The password of the elastic account is specified when you create the cluster. If you forget the password, you can reset it. For more information about the procedure and precautions for resetting a password, see Reset the access password for an Elasticsearch cluster.
index	The name of the destination index. If no indexes are created in the Elasticsearch cluster, create one first. For more information, see Step 3: Create an index. You can also enable the Auto Indexing feature for the Elasticsearch cluster to automatically create indexes. For more information, see Configure the YML file.
typeName	The type of the destination index. The index type of Elasticsearch clusters of V7.0 or later must be _doc.

For more information about other parameters in the WITH part, see Create an Elasticsearch result table.

> **Note**
>
> - Elasticsearch allows you to update documents based on the `PRIMARY KEY` field. Only one field can be specified as the `PRIMARY KEY` field. If you specify the `PRIMARY KEY` field, values of the `PRIMARY KEY` field are used as document IDs. If the `PRIMARY KEY` field is not specified, the system generates random IDs for documents. For more information, see Index API.
> - Elasticsearch supports multiple update modes. You can set the `updateMode` parameter to specify the update mode.
>   - If `updateMode` is set to full, new documents overwrite existing documents.
>   - If `updateMode` is set to inc, new values overwrite existing values of the related fields.
> - All updates in Elasticsearch are performed by using INSERT or UPDATE statements that follow the UPSERT syntax.

iii. Create data consumption logic and synchronize data.

```
INSERT INTO es_stream_sink
SELECT
  a,
  count(*) as cnt
FROM sls_stream GROUP BY a
```

4. Submit and run the job.

   For more information, see Publish a job and Start a job.

   After you submit and run the job, data stored in Log Service is aggregated and imported into the Elasticsearch cluster. Realtime Compute for Apache Flink also supports other compute operations. For more information, see Overview.

Summary

Realtime Compute for Apache Flink and Elasticsearch allow you to quickly create your own real-time search services. If more complex logic is required to import data into an Elasticsearch cluster, use the user-defined sinks of Realtime Compute for Apache Flink. For more information, see Create a custom result table.

4.3. Use DataWorks to synchronize data from a Hadoop cluster to an Alibaba Cloud Elasticsearch cluster

When you use a Hadoop cluster to perform interactive big data analytics and queries, the process may be time-consuming. To address this issue, you can synchronize data from the Hadoop cluster to an Alibaba Cloud Elasticsearch cluster for analytics and queries. Elasticsearch can respond to multiple types of queries within seconds, especially ad hoc queries. This topic describes how to synchronize data from a Hadoop cluster to an Elasticsearch cluster by using the data synchronization feature of DataWorks.

Procedure

1. Preparations

   Create a Hadoop cluster, a DataWorks workspace, and an Elasticsearch cluster. Configure the Elasticsearch cluster.

2. Step 1: Prepare data

   Create test data in the Hadoop cluster.

3. Step 2: Purchase and create an exclusive resource group
4. Step 3: Add data sources

   Connect the Elasticsearch cluster and the HDFS of the Hadoop cluster to the Data Integration service of DataWorks.

5. Step 4: Create and run a data synchronization task
6. Step 5: View synchronization results

   In the Kibana console, view the synchronized data and search for data based on specific conditions.

Preparations

1. Create a Hadoop cluster.

   Before you synchronize data, make sure that your Hadoop cluster runs normally. In this step, the Alibaba Cloud E-MapReduce (EMR) service is used to automatically create a Hadoop cluster. For more information, see Create a cluster.

   Sample configurations of the EMR Hadoop cluster: (Default configurations are used for items that are not listed. You can also modify the default configurations based on your business needs.)

   - **Cluster Type**: Hadoop
   - **EMR Version**: EMR-3.26.3
   - **Assign Public IP Address**: turned on

2. Create an Elasticsearch cluster and enable the Auto Indexing feature for the cluster.

   For more information, see Create an Alibaba Cloud Elasticsearch cluster and Access and configure an Elasticsearch cluster. Make sure that the Elasticsearch cluster resides in the same virtual private cloud (VPC), region, and zone as the EMR Hadoop cluster. In this step, an Elasticsearch V6.7.0 cluster of the Standard Edition is created.

3. Create a DataWorks workspace.

   Make sure that the workspace resides in the same region as the Elasticsearch cluster. For more information, see Create a workspace.

Step 1: Prepare data

1. Log on to the EMR console.

2. In the top navigation bar, select the region where your EMR Hadoop cluster resides.
3. In the **Clusters** section of the page that appears, find your EMR Hadoop cluster and click its ID.
4. In the upper part of the page, click the **Data Platform** tab.
5. Click **Create Project** to create a data development project. In this step, set Select Resource Group to Default Resource Group.

   For more information, see Manage projects.
6. In the **Projects** section, find the created project and click **Edit Job** in the **Actions** column to create a job.

   For more information, see Edit jobs. In this step, set Job Type to Hive.
7. Create a data table and insert data into the table.
   i. In the code editor, enter a statement to create a Hive table. Then, click **Run**.

      In this step, the following statement is used:

      ```
      CREATE TABLE IF NOT
      EXISTS hive_esdoc_good_sale(
       create_time timestamp,
       category STRING,
       brand STRING,
       buyer_id STRING,
       trans_num BIGINT,
       trans_amount DOUBLE,
       click_cnt BIGINT
       )
       PARTITIONED BY (pt string) ROW FORMAT
       DELIMITED FIELDS TERMINATED BY ',' lines terminated by '\n'
      ```

   ii. In the **Run Job** dialog box, configure the parameters and click **OK**.
      - Set Select Resource Group to Default Resource Group.
      - Set Target Cluster to the cluster you created.

   iii. Create another job. In the code editor, enter the following SQL statement to insert test data.

      You can import data from Object Storage Service (OSS) or other data sources. You can also manually insert data. In this step, data is manually inserted.

      ```
      insert into
      hive_esdoc_good_sale PARTITION(pt =1 ) values('2018-08-21','Coat','Brand A','lilei'
      ,3,500.6,7),('2018-08-22','Fresh','Brand B','lilei',1,303,8),('2018-08-22','Coat','
      Brand C','hanmeimei',2,510,2),(2018-08-22,'Bathroom','Brand A','hanmeimei',1,442.5,
      1),('2018-08-22','Fresh','Brand D','hanmeimei',2,234,3),('2018-08-23','Coat','Brand
      B','jimmy',9,2000,7),('2018-08-23','Fresh','Brand A','jimmy',5,45.1,5),('2018-08-23
      ','Coat','Brand E','jimmy',5,100.2,4),('2018-08-24','Fresh','Brand G','peiqi',10,55
      60,7),('2018-08-24','Bathroom','Brand F','peiqi',1,445.6,2),('2018-08-24','Coat','B
      rand A','ray',3,777,3),('2018-08-24','Bathroom','Brand G','ray',3,122,3),('2018-08-
      24','Coat','Brand C','ray',1,62,7) ;
      ```
8. Check whether the data is inserted.
   i. Create a job for an ad hoc query.

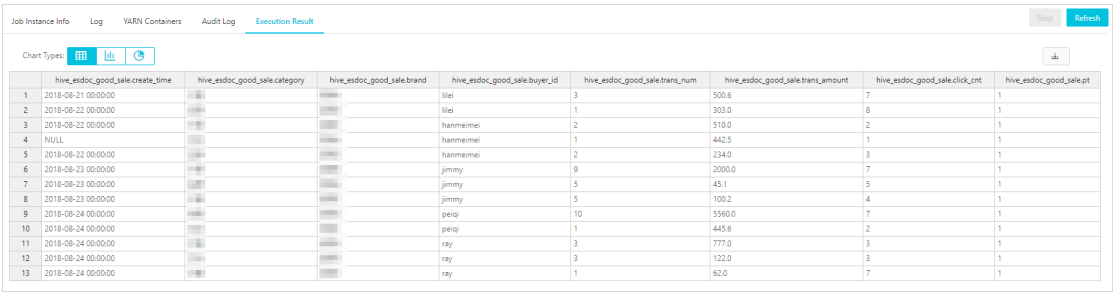
      For more information, see Perform ad hoc queries.

ii. Enter the following SQL statement and click **Run**:

```
select * from hive_esdoc_good_sale where pt =1;
```

iii. In the lower part of the page, click the **Records** tab. On this tab, click **Details** in the **Action** column.

iv. In the upper part of the page, click the **Scheduling Center** tab. On this tab, click the **Execution Result** tab.
Then, you can check whether the data is inserted into the Hive table of the Hadoop cluster for synchronization. The following figure shows the inserted data.

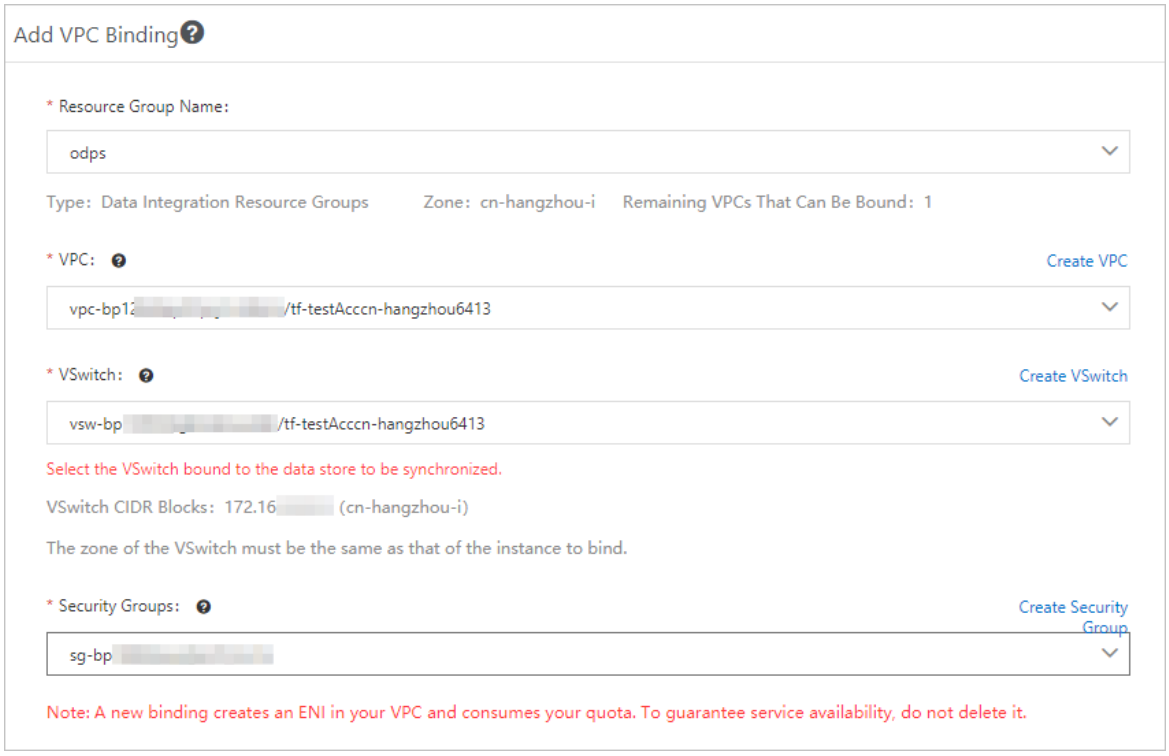


	hive_esdoc_good_sale.create_time	hive_esdoc_good_sale.category	hive_esdoc_good_sale.brand	hive_esdoc_good_sale.buyer_id	hive_esdoc_good_sale.trans_num	hive_esdoc_good_sale.trans_amount	hive_esdoc_good_sale.click_cnt	hive_esdoc_good_sale.pt
1	2018-08-21 00:00:00			lilei	3	500.6	7	1
2	2018-08-22 00:00:00			lilei	1	303.0	8	1
3	2018-08-22 00:00:00			hanmeimei	2	510.0	2	1
4	NULL			hanmeimei	1	442.5	1	1
5	2018-08-22 00:00:00			hanmeimei	2	234.0	3	1
6	2018-08-23 00:00:00			jimmy	9	2000.0	7	1
7	2018-08-23 00:00:00			jimmy	5	45.1	5	1
8	2018-08-23 00:00:00			jimmy	5	100.2	4	1
9	2018-08-24 00:00:00			peiqi	10	5560.0	7	1
10	2018-08-24 00:00:00			peiqi	1	445.6	2	1
11	2018-08-24 00:00:00			ray	3	777.0	3	1
12	2018-08-24 00:00:00			ray	3	122.0	3	1
13	2018-08-24 00:00:00			ray	1	62.0	7	1

Step 2: Purchase and create an exclusive resource group

1.
2.
3.
4.
5. Find the created exclusive resource group. Then, click **Add VPC Binding** in the Actions column to bind the exclusive resource group to a VPC. For more information, see Configure network settings.

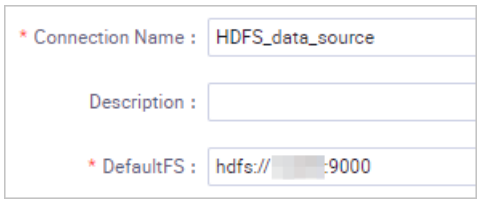
   Exclusive resources are deployed in a VPC managed by DataWorks. DataWorks can be used to synchronize data from the Hadoop cluster to the Elasticsearch cluster only after it connects to the VPCs where the clusters reside. In this topic, the Hadoop cluster and Elasticsearch cluster reside in the same VPC. Therefore, you only need to select the **VPC** and **VSwitch** of the Elasticsearch cluster for the binding.



6.

Step 3: Add data sources

1.
2.
3.
4. In the **Semi-structured storage** section of the **Add data source** dialog box, click **HDFS**.
5. In the **Add HDFS data source** dialog box, specify **Data Source Name** and **DefaultFS**.



DefaultFS: If your EMR Hadoop cluster is in non-HA mode, set this parameter to `hdfs://Internal IP address of emr-header-1:9000`. If your EMR Hadoop cluster is in HA mode, set this parameter to `hdfs://Internal IP address of emr-header-1:8020`. The internal IP address of emr-header-1 is used because emr-header-1 communicates with DataWorks over a VPC.

6. Click **Complete**.
7.

Step 4: Create and run a data synchronization task

1.
2.
3.

4. In the upper part of the page, click the  icon.
5. In the Tips message, click OK. Then, configure the data synchronization script.
   For more information, see Create a synchronization node by using the code editor.

   > **Note** You can also click the  icon in the upper part of the page to import a script configuration template. Then, modify the template as required.

   The following code provides a sample script:

```
{
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "10"
        },
        "speed": {
            "concurrent": 3,
            "throttle": false
        }
    },
    "steps": [
        {
            "category": "reader",
            "name": "Reader",
            "parameter": {
                "column": [
                    {
                        "format": "yyyy-MM-dd HH:mm:ss",
                        "index": 0,
                        "type": "date"
                    },
                    {
                        "index": 1,
                        "type": "string"
                    },
                    {
                        "index": 2,
                        "type": "string"
                    },
                    {
                        "index": 3,
                        "type": "string"
                    },
                    {
```

```
                        "index": 4,
                        "type": "long"
                    },
                    {
                        "index": 5,
                        "type": "double"
                    },
                    {
                        "index": 6,
                        "type": "long"
                    }
                ],
                "datasource": "HDFS_data_source",
                "encoding": "UTF-8",
                "fieldDelimiter": ",",
                "fileType": "text",
                "path": "/user/hive/warehouse/hive_esdoc_good_sale"
            },
            "stepType": "hdfs"
        },
        {
            "category": "writer",
            "name": "Writer",
            "parameter": {
                "batchSize": 1000,
                "cleanup": true,
                "column": [
                    {
                        "name": "create_time",
                        "type": "id"
                    },
                    {
                        "name": "category",
                        "type": "text"
                    },
                    {
                        "name": "brand",
                        "type": "text"
                    },
                    {
                        "name": "buyer_id",
                        "type": "text"
                    },
                    {
                        "name": "trans_num",
                        "type": "integer"
                    },
                    {
                        "name": "trans_amount",
                        "type": "double"
                    },
                    {
                        "name": "click_cnt",
                        "type": "integer"
```

```
                    }
                ],
                "datasource": "ES_data_source",
                "discovery": false,
                "index": "hive_esdoc_good_sale",
                "indexType": "_doc",
                "splitter": ","
            },
            "stepType": "elasticsearch"
        }
    ],
    "type": "job",
    "version": "2.0"
}
```

The preceding script includes three parts.

Part	Description
`setting`	Used to configure parameters related to packet loss and the maximum concurrency during synchronization. The default value of the `record` field in the `errorLimit` parameter is 0. You must set the field to a larger value, such as 10.
`Reader`	Used to configure the Hadoop cluster as the reader. `path` specifies the location of the data that is stored in the Hadoop cluster. To obtain the location, log on to the master node of the Hadoop cluster and run the `hdfs dfs -ls /user/hive/warehouse/hive_esdoc_good_sale` command. For a partitioned table, the data synchronization feature of DataWorks can automatically recurse to the partition where the data is stored. For more information, see HDFS Reader.
`Writer`	Used to configure the Elasticsearch cluster as the writer. For more information, see Elasticsearch Writer. ○ `index`: the name of the destination index. ○ `indexType`: the type of the destination index. The index type of Elasticsearch clusters of V7.0 or later must be `_doc`.

6.
7.
8.
9.

Step 5: View synchronization results

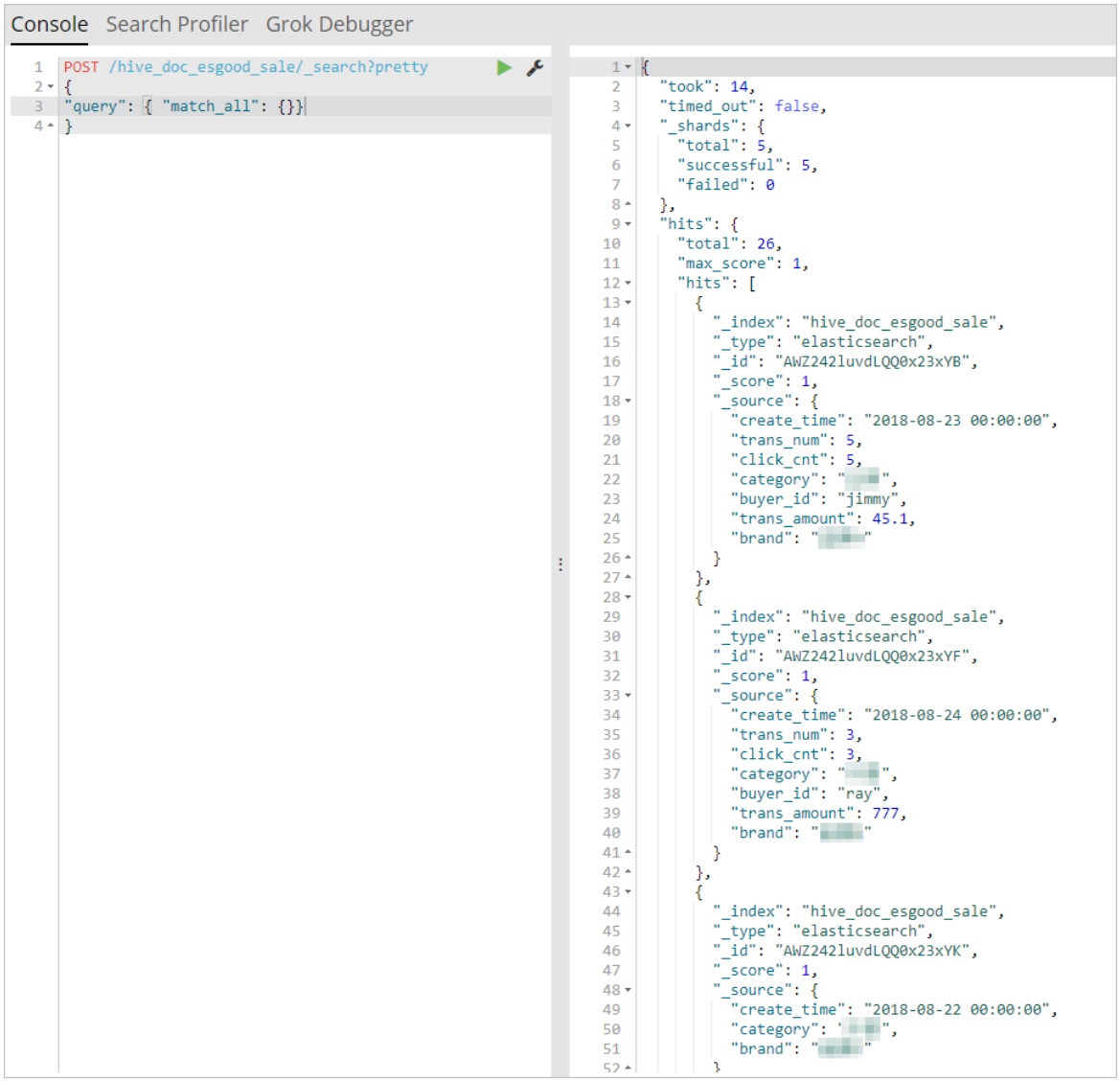
1. Log on to the Kibana console of the destination Elasticsearch cluster.

   For more information, see Log on to the Kibana console.
2. In the left-side navigation pane, click **Dev Tools**.
3. On the **Console** tab of the page that appears, run the following command to query the synchronized data:

```
POST /hive_esdoc_good_sale/_search?pretty
{
"query": { "match_all": {}}
}
```

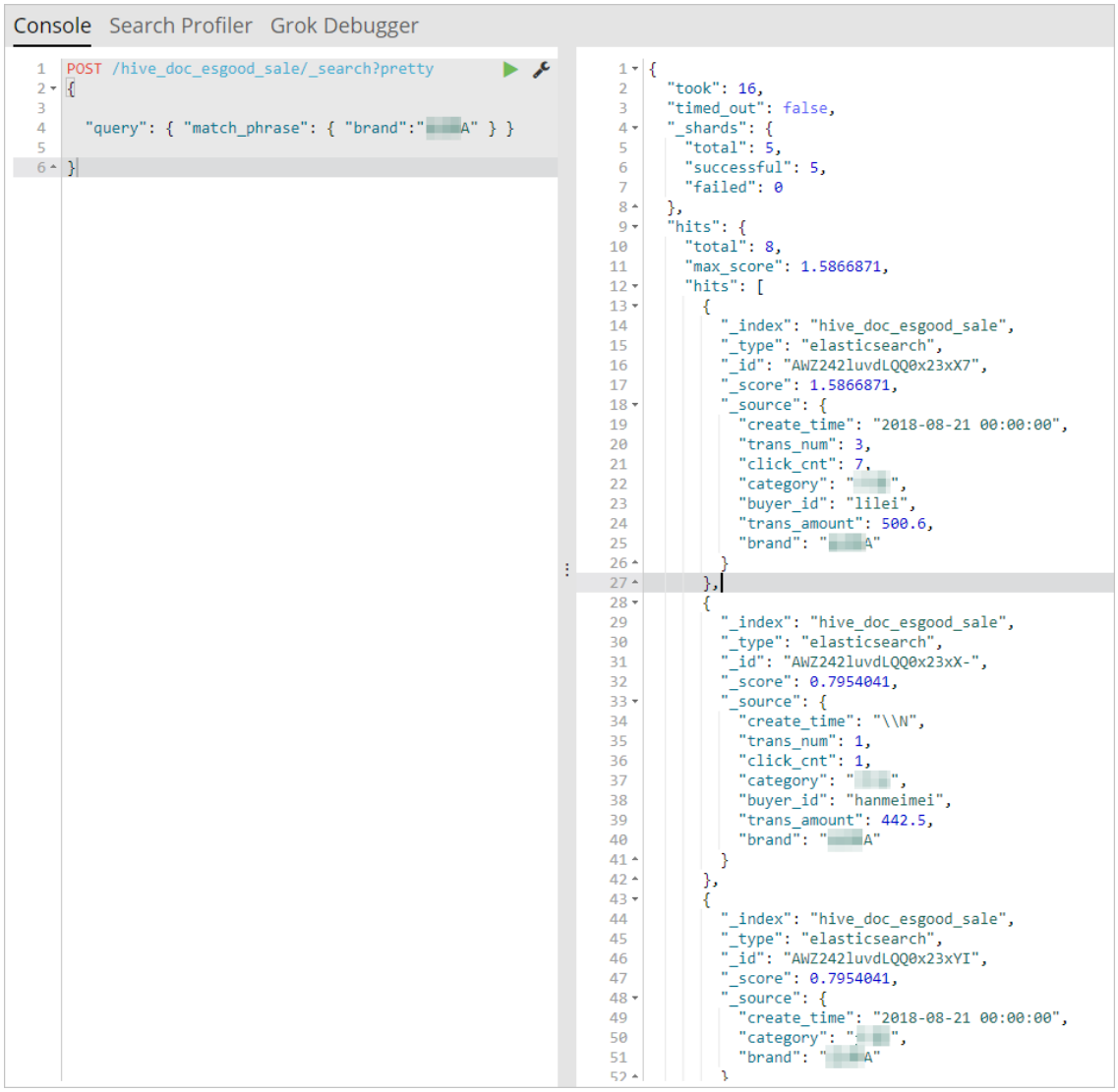
Note `hive_esdoc_good_sale` is the index name that is specified by the `index` field in the data synchronization script.

If the data is synchronized, the result shown in the following figure is returned.



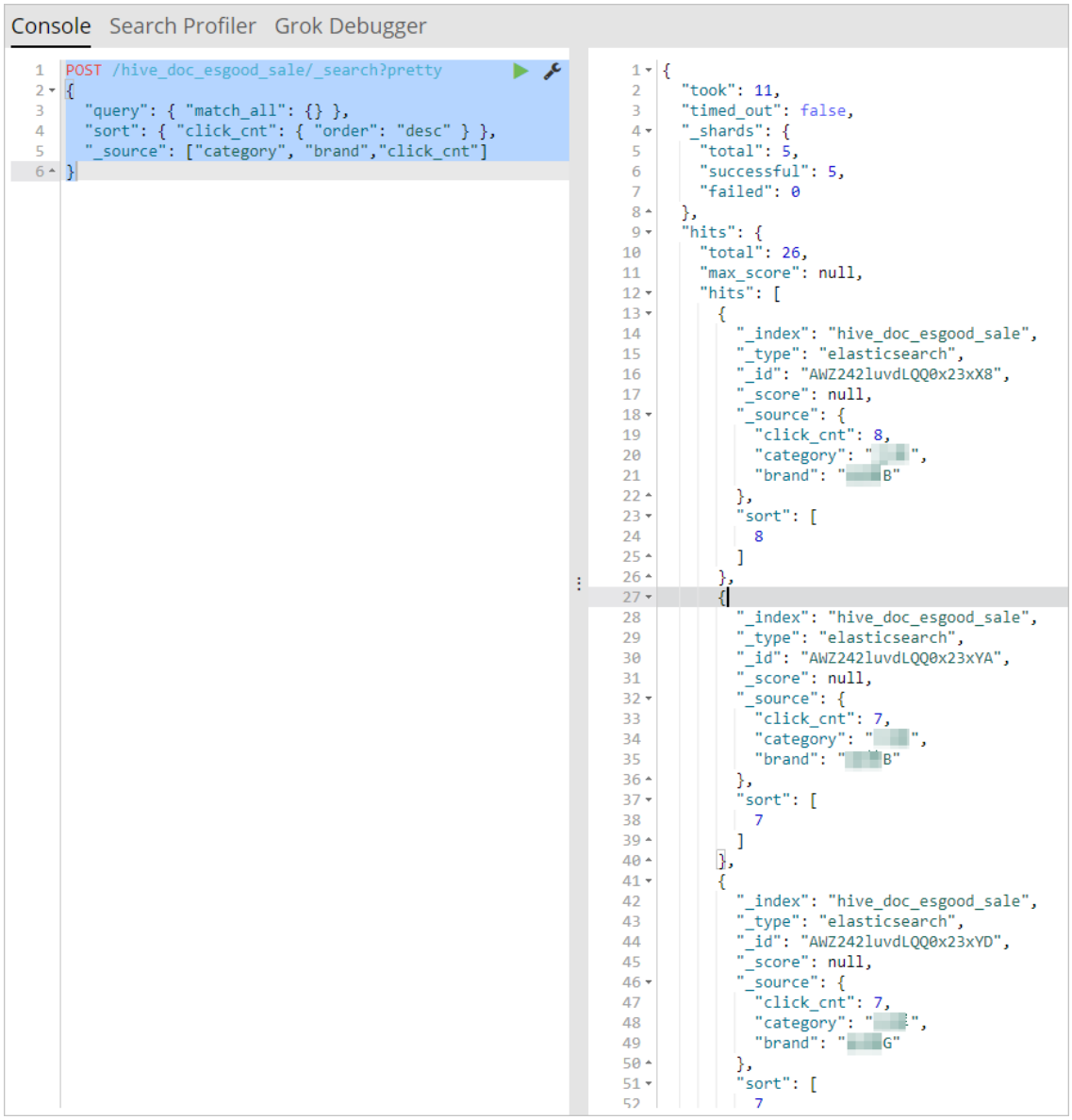
4. Run the following command to search for all documents that contain Brand A:

```
POST /hive_esdoc_good_sale/_search?pretty
{
  "query": { "match_phrase": { "brand":"Brand A" } }
}
```



5. Run the following command to sort products of each brand based on the **number of clicks**. Then, determine the popularity of the products:

```
POST /hive_esdoc_good_sale/_search?pretty
{
"query": { "match_all": {} },
"sort": { "click_cnt": { "order": "desc" } },
"_source": ["category", "brand","click_cnt"]
}
```



For more information about other commands and their use scenarios, see Alibaba Cloud Elasticsearch documentation and open source Elasticsearch documentation.

5.Data migration

5.1. Migrate documents from a Solr cluster to an Alibaba Cloud Elasticsearch cluster

This topic describes how to use the solr-to-es tool to migrate documents from a Solr cluster to an Alibaba Cloud Elasticsearch cluster. The tool is provided by a third-party community.

Preparations

1. Create an Alibaba Cloud Elasticsearch V6.X cluster. This topic uses an Elasticsearch V6.3.2 cluster as an example. For more information, see Create an Alibaba Cloud Elasticsearch cluster.

   > **Notice** The solr-to-es tool used in this topic supports only Elasticsearch V6.X clusters. If you want to use an Elasticsearch cluster of another version, first perform a compatibility test.

2. Enable the Auto Indexing feature for the cluster. For more information, see Access and configure an Elasticsearch cluster.
3. Create an Alibaba Cloud Elastic Compute Service (ECS) instance. For more information, see Step 1: Create an ECS instance. In this topic, the ECS instance runs CentOS 7.3.

   > **Notice** The ECS instance must reside in the same region, zone, and Virtual Private Cloud (VPC) as the Elasticsearch cluster.

4. Install Solr on the ECS instance. This topic uses Solr 5.0.0 as an example. For more information, see Official Solr documentation.
5. Install Python on the ECS instance. The version must be 3.0 or later. This topic uses Python 3.6.2 as an example.
6. Install pysolr on the ECS instance. The version must be 3.3.3 or later but earlier than 4.0.

Install solr-to-es

1. Connect to the ECS instance and download solr-to-es.
2. Navigate to the directory where *setup.py* is stored and run the `python setup.py install` command to install solr-to-es.
3. After solr-to-es is installed, run the following command to migrate documents:

   ```
   python __main__.py <solr_url>:8983/solr/<my_core>/select http://<username>:<password>
   @<elasticsearch_url>:9200 <elasticsearch_index> <doc_type>
   ```

   Parameters

Parameter	Description
`<solr_url>`	The endpoint of your Solr cluster. Example: http://116.62.**.**.

Parameter	Description
`<my_core>`	The name of the Solr Core that contains the documents you want to migrate.
`<username>`	The username that is used to access your Elasticsearch cluster. The default username is elastic.
`<password>`	The password that is used to access your Elasticsearch cluster. The password is specified when you create the cluster.
`<elasticsearch_url>`	The internal or public endpoint of your Elasticsearch cluster. You can obtain the endpoint from the Basic Information page of your cluster. For more information, see View the basic information of a cluster.
`<elasticsearch_index>`	The name of the index to which documents will be migrated.
`<doc_type>`	The type of the index.

Notice If you are using solr-to-es of a version that is different from the one described in this topic, you can try the following command to migrate documents. For more information, see solr-to-es.

```
solr-to-es [-h] [--solr-query SOLR_QUERY] [--solr-fields COMMA_SEP_FIELDS]
                [--rows-per-page ROWS_PER_PAGE] [--es-timeout ES_TIMEOUT]
                solr_url elasticsearch_url elasticsearch_index doc_type
```

If you use the preceding command in the environment described in this topic, the `-bash: solr-to-es.py: command not found` error is returned.

Procedure

Query all documents in the `my_core` Solr Core and write these documents to the index on your Elasticsearch cluster. The name of the index is `elasticsearch_index`, and the type of the index is `doc_type`.

1. In the Solr environment, navigate to the *solr-to-es-master/solr_to_es* directory.
2. Run the following command:

```
python __main__.py 'http://116.62.**.**:8983/solr/my_core/select?q=*%3A*&wt=json&indent=true' 'http://elastic:Your password@es-cn-so4lwf40ubsrf****.public.elasticsearch.aliyuncs.com:9200' elasticsearch_index doc_type
```

Parameter	Description
`q`	Required. This parameter defines a query that uses the standard query syntax in Solr. Operators are supported. The value `*%3A*` indicates that all documents will be queried.
`wt`	The type of the data to return. Valid values: JSON, XML, PY, RB, and CSV.

Parameter	Description
`indent`	Specifies whether to use indentations to ensure that the returned data is easier to read. Default value: `false`.

For information about other parameters, see Parameters.

3. Log on to the Kibana console of your Elasticsearch cluster.

   For more information, see Log on to the Kibana console.

4. In the left-side navigation pane, click **Dev Tools**. On the **Console** tab of the page that appears, run the following command to check whether the `elasticsearch_index` index is created on the Elasticsearch cluster:

   ```
   GET _cat/indices?v
   ```

5. Run the following command to query details about the migrated documents:

   ```
   GET /elasticsearch_index/doc_type/_search
   ```

   If the command is executed successfully, the following result is returned:

```
{
  "took" : 12,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 2,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "elasticsearch_index",
        "_type" : "doc_type",
        "_id" : "Tz8WNW4BwRjcQciJ****",
        "_score" : 1.0,
        "_source" : {
          "id" : "2",
          "title" : [
            "test"
          ],
          "_version_" : 1648195017403006976
        }
      },
      {
        "_index" : "elasticsearch_index",
        "_type" : "doc_type",
        "_id" : "Tj8WNW4BwRjcQciJ****",
        "_score" : 1.0,
        "_source" : {
          "id" : "1",
          "title" : [
            "change.me"
          ],
          "_version_" : 1648195007391203328
        }
      }
    ]
  }
}
```

6.Using ES-Hadoop

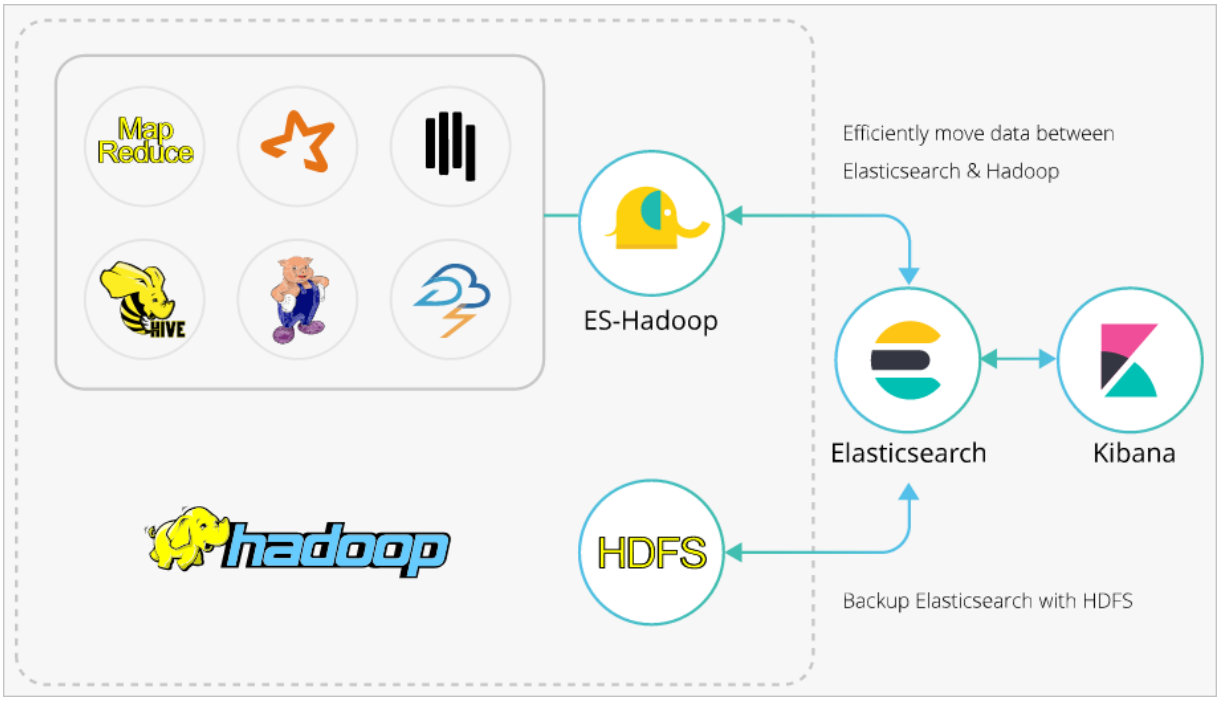
6.1. Use ES-Hadoop to enable Hive to write data to and read data from Alibaba Cloud Elasticsearch

Elasticsearch-Hadoop (ES-Hadoop) is a tool developed by open source Elasticsearch. It connects Elasticsearch to Apache Hadoop and enables data transmission between them. ES-Hadoop combines the quick search capability of Elasticsearch and the batch processing capability of Hadoop to achieve interactive data processing. This topic describes how to use ES-Hadoop to enable Hive to write data to and read data from Alibaba Cloud Elasticsearch.

Context

Hadoop can handle large datasets. However, when it is used for interactive analytics, a high latency occurs. Elasticsearch has an advantage over Hadoop in interactive analytics. It can respond to queries, especially ad hoc queries, within seconds. ES-Hadoop combines the advantages of Hadoop and Elasticsearch. ES-Hadoop allows you to make only a few code modifications to process the data that is stored in Elasticsearch. ES-Hadoop also provides an accelerated query experience.

ES-Hadoop uses Elasticsearch as a data source of data processing engines, such as MapReduce, Spark, and Hive. ES-Hadoop also uses Elasticsearch as storage in a computing-storage separation architecture. Elasticsearch works in a similar way to other data sources of MapReduce, Spark, and Hive. However, Elasticsearch can select and filter data in a more rapid manner. This is critical to an analytics engine.



Procedure

1. Preparations

   Create an Alibaba Cloud Elasticsearch cluster and an E-MapReduce (EMR) cluster in the same virtual

private cloud (VPC). Disable the Auto Indexing feature for the Elasticsearch cluster. Create an index in the Elasticsearch cluster and configure mappings for the index. Download the ES-Hadoop package that is compatible with the version of the Elasticsearch cluster.

2. Step 1: Upload the ES-Hadoop JAR package to HDFS

   Upload the ES-Hadoop package to the HDFS directory on the master node of the EMR cluster.

3. Step 2: Create a Hive external table

   Create a Hive external table and map the fields in the table with those in the index of the Elasticsearch cluster.

4. Step 3: Use Hive to write data to the index

   Use HiveSQL to write data to the index of the Elasticsearch cluster.

5. Step 4: Use Hive to read data from the index

   Use HiveSQL to read data from the index of the Elasticsearch cluster.

Preparations

1. Create an Alibaba Cloud Elasticsearch cluster.

   In this topic, an Elasticsearch V6.7.0 cluster is created. For more information, see Create an Alibaba Cloud Elasticsearch cluster.

2. Disable the Auto Indexing feature for the cluster. Create an index in the cluster and configure mappings for the index.

   If you enable the Auto Indexing feature for the cluster, the index that is automatically created by the Elasticsearch cluster may not meet your requirements. For example, you define the age field of the INT data type and enable the Auto Indexing feature. In this case, the data type of the age field may become LONG in the index. Therefore, we recommend that you disable the Auto Indexing feature. An index named company is created in this topic. The following code shows this index and its mappings:

```
PUT company
{
  "mappings": {
    "_doc": {
      "properties": {
        "id": {
          "type": "long"
        },
        "name": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword",
              "ignore_above": 256
            }
          }
        },
        "birth": {
          "type": "text"
        },
        "addr": {
          "type": "text"
        }
      }
    }
  },
  "settings": {
    "index": {
      "number_of_shards": "5",
      "number_of_replicas": "1"
    }
  }
}
```

3. Create an EMR cluster that resides in the same VPC as the Elasticsearch cluster.

   Notice By default, 0.0.0.0/0 is specified in the private IP address whitelist of the Elasticsearch cluster. You can view the whitelist configuration on the cluster security configuration page. If the default setting is not used, you must add the private IP address of the EMR cluster to the whitelist.

   - For more information about how to obtain the private IP address of the EMR cluster, see View the cluster list and cluster details.
   - For more information about how to configure the private IP address whitelist of the Elasticsearch cluster, see Configure a public or private IP address whitelist for an Elasticsearch cluster. The IP addresses in the whitelist can be used to access the Elasticsearch cluster over a VPC.

4. Download an ES-Hadoop package that is compatible with the version of the Elasticsearch cluster.

   The elasticsearch-hadoop-6.7.0.zip package is used in this topic.

Step 1: Upload the ES-Hadoop JAR package to HDFS

1. Log on to the EMR console and obtain the IP address of the master node of the EMR cluster. Then, use SSH to log on to the Elastic Compute Service (ECS) instance that is indicated by the IP address.

   For more information, see Log on to a cluster.

2. Upload the elasticsearch-hadoop-6.7.0.zip package to the master node. Decompress the package to obtain the elasticsearch-hadoop-hive-6.7.0.jar file.

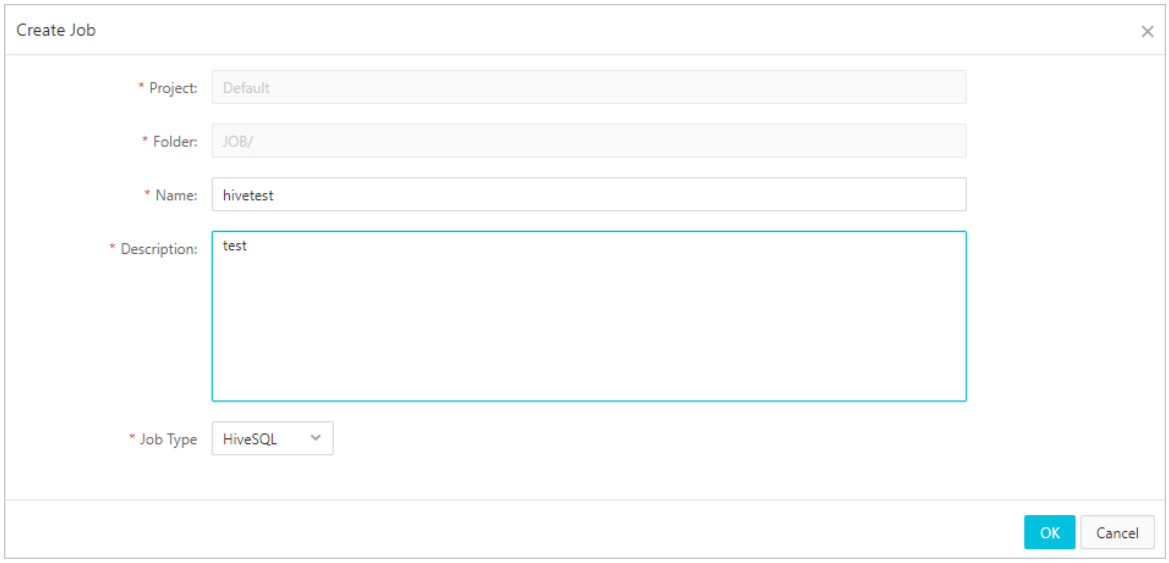
3. Create an HDFS directory and upload the elasticsearch-hadoop-hive-6.7.0.jar file to the directory.

```
hadoop fs -mkdir /tmp/hadoop-es
hadoop fs -put elasticsearch-hadoop-6.7.0/dist/elasticsearch-hadoop-hive-6.7.0.jar /tmp
/hadoop-es
```

Step 2: Create a Hive external table

1. On the **Data Platform** tab of the EMR console, create a **HiveSQL** job.

   For more information, see Configure a Hive SQL job.



2. Configure the job and create a Hive external table.

   The following code shows the configuration of the job:

```
####Add a JAR file, which is valid only for the current session.########
add jar hdfs:///tmp/hadoop-es/elasticsearch-hadoop-hive-6.7.0.jar;
####Create a Hive external table and map the table with the index of the Elasticsearch
cluster.####
CREATE EXTERNAL table IF NOT EXISTS company(
   id BIGINT,
   name STRING,
   birth STRING,
   addr STRING
)
STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'
TBLPROPERTIES(
    'es.nodes' = 'http://es-cn-mp91kzb8m0009****.elasticsearch.aliyuncs.com',
    'es.port' = '9200',
    'es.net.ssl' = 'true',
    'es.nodes.wan.only' = 'true',
    'es.nodes.discovery'='false',
    'es.input.use.sliced.partitions'='false',
    'es.input.json' = 'false',
    'es.resource' = 'company/_doc',
    'es.net.http.auth.user' = 'elastic',
    'es.net.http.auth.pass' = 'xxxxxx'
);
```

ES-Hadoop parameters

Parameter	Default value	Description
es.nodes	localhost	The endpoint that is used to access the Elasticsearch cluster. We recommend that you use the internal endpoint. You can obtain the internal endpoint on the Basic Information page of the Elasticsearch cluster. For more information, see View the basic information of a cluster.
es.port	9200	The port number that is used to access the Elasticsearch cluster.

Parameter	Default value	Description
es.net.http.auth.user	elastic	The username that is used to access the Elasticsearch cluster. **Note** If you use the elastic account to access your Elasticsearch cluster and then reset the password of the account, it may require some time for the new password to take effect. During this period, you cannot use the elastic account to access the cluster. Therefore, we recommend that you do not use the elastic account to access an Elasticsearch cluster. You can log on to the Kibana console and create a user with the required role to access an Elasticsearch cluster. For more information, see Use the RBAC mechanism provided by Elasticsearch X-Pack to implement access control.
es.net.http.auth.pass	/	The password that is used to access the Elasticsearch cluster.
es.nodes.wan.only	false	Specifies whether to enable node sniffing when the Elasticsearch cluster uses a virtual IP address for connections. Valid values: ○ true: indicates that node sniffing is enabled. ○ false: indicates that node sniff is disabled.
es.nodes.discovery	true	Specifies whether to prohibit the node discovery mechanism. Valid values: ○ true: indicates that the node discovery mechanism is prohibited. ○ false: indicates that the node discovery mechanism is not prohibited. **Notice** If you use Alibaba Cloud Elasticsearch, you must set this parameter to false.

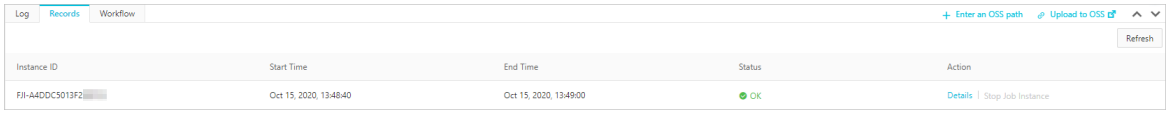
Parameter	Default value	Description
es.input.use.sliced.partitions	true	Specifies whether to use partitions. Valid values: ○ true: indicates that partitions are used. In this case, more time may be required for the index read-ahead phase. The time required for this phase may be longer than the time required for data queries. To improve query efficiency, we recommend that you set this parameter to false. ○ false: indicates that partitions are not used.
es.index.auto.create	true	Specifies whether the system creates an index in the Elasticsearch cluster when you use ES-Hadoop to write data to the cluster. Valid values: ○ true: indicates that the system creates an index in the Elasticsearch cluster. ○ false: indicates that the system does not create an index in the Elasticsearch cluster.
es.resource	/	The name and type of the index on which data read or write operations are performed.
es.mapping.names	/	The mappings between the field names in the table and those in the index of the Elasticsearch cluster.
es.read.metadata	false	Specifies whether to include the document metadata such as **_id** in the results. To include the document metadata, set the value to true.

For more information about the configuration items of ES-Hadoop, see open source ES-Hadoop configuration.

3. Save and run the job.

```
HIVE_SQL FJ-36DA8E09A86AA0D9 :
1  add jar hdfs:///tmp/hadoop-es/elasticsearch-hadoop-hive-6.7.0.jar;
2
3  CREATE EXTERNAL table IF NOT EXISTS company(
4     id BIGINT,
5     name STRING,
6     birth STRING,
7     addr STRING
8  )
9  STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'
10 TBLPROPERTIES(
11    'es.nodes' = 'http://es-cn-n6w.elasticsearch.aliyuncs.com',
12    'es.port' = '9200',
13    'es.net.ssl' = 'true',
14    'es.nodes.wan.only' = 'true',
15    'es.nodes.discovery'='false',
16    'es.input.json' = 'false',
17    'es.resource' = 'company/_doc',
18    'es.net.http.auth.user' = 'elastic',
19    'es.net.http.auth.pass' = '
20 );
```

If the job is successfully run, the result shown in the following figure is returned.



Step 3: Use Hive to write data to the index

1. Create a **HiveSQL** data write job.

   The following code shows the configuration of the job:

```
add jar hdfs:///tmp/hadoop-es/elasticsearch-hadoop-hive-6.7.0.jar;
INSERT INTO TABLE company VALUES (1, "zhangsan", "1990-01-01","No.969, wenyixi Rd, yuha
ng, hangzhou");
INSERT INTO TABLE company VALUES (2, "lisi", "1991-01-01", "No.556, xixi Rd, xihu, hang
zhou");
INSERT INTO TABLE company VALUES (3, "wangwu", "1992-01-01", "No.699 wangshang Rd, binj
iang, hangzhou");
```

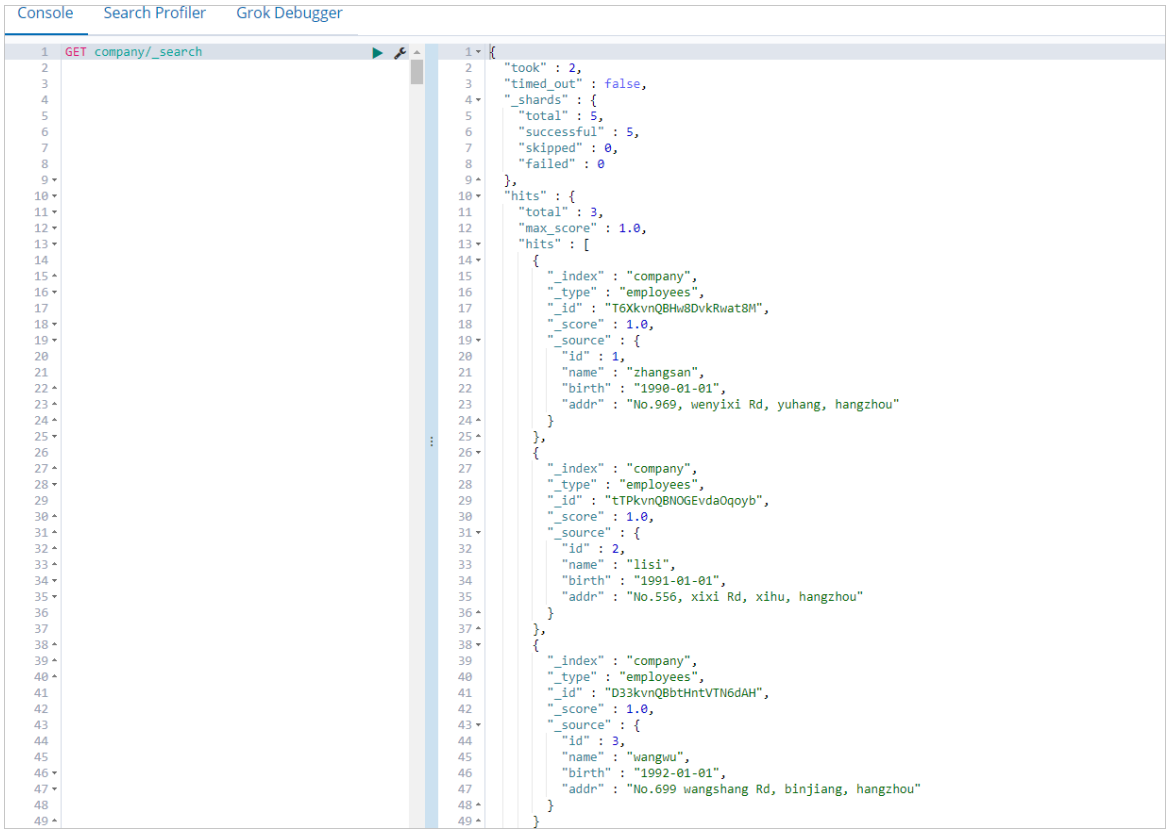
2. Save and run the job.

3. If the job is successfully run, log on to the Kibana console of the Elasticsearch cluster and query the data in the company index.

   For more information about how to log on to the Kibana console, see Log on to the Kibana console. You can run the following command to query the data in the company index:

```
GET company/_search
```

If the command is successfully run, the result shown in the following figure is returned.



Step 4: Use Hive to read data from the index

1. Create a **HiveSQL** data read job.

   The following code shows the configuration of the job:

   ```
   add jar hdfs:///tmp/hadoop-es/elasticsearch-hadoop-hive-6.7.0.jar;
   select * from company;
   ```

2. Save and run the job.

Summary

This topic describes how to enable Hive to read and write data by using ES-Hadoop. Alibaba Cloud EMR and Elasticsearch are used in this topic. Data read and write by using Hive achieve more flexible data analytics. For more information about the advanced configurations of ES-Hadoop and Hive, see open source Elasticsearch documentation.

6.2. Use ES-Hadoop to write HDFS data to Elasticsearch

ES-Hadoop is a tool developed by open source Elasticsearch. It connects Elasticsearch to Apache Hadoop and enables data transmission between them. ES-Hadoop combines the quick search capability of Elasticsearch and the batch processing capability of Hadoop to achieve interactive data processing. For some complex data analytics tasks, you must run a MapReduce task to read data from the JSON files stored in Hadoop Distributed File System (HDFS) and write the data to an Elasticsearch cluster. This topic describes how to use ES-Hadoop to run such a MapReduce task.

Procedure

1. Preparations

   Create an Alibaba Cloud Elasticsearch cluster and an E-MapReduce (EMR) cluster in the same virtual private cloud (VPC). Then, enable the Auto Indexing feature for the Elasticsearch cluster, and prepare test data and a Java environment.

2. Step 1: Upload the ES-Hadoop JAR package to HDFS

   Download the ES-Hadoop package and upload the package to the HDFS directory on the master node in the EMR cluster.

3. Step 2: Configure POM dependencies

   Create a Java Maven project and configure POM dependencies.

4. Step 3: Compile code and run a MapReduce task

   Compile the Java code that is used to write data to the Elasticsearch cluster. Compress the code into a JAR package and upload the package to the EMR cluster. Then, run the code in a MapReduce task to write data.

5. Step 4: Verify the results

   Log on to the Kibana console of the Elasticsearch cluster. Then, query the data that is written by the MapReduce task.

Preparations

1. Create an Alibaba Cloud Elasticsearch cluster and enable the Auto Indexing feature for the cluster.

   For more information, see Create an Alibaba Cloud Elasticsearch cluster and Access and configure an Elasticsearch cluster. In this topic, an Elasticsearch V6.7.0 cluster is created.

   > **Notice** In a production environment, we recommend that you disable the Auto Indexing feature. You must create an index and configure mappings for the index in advance. The Elasticsearch cluster used in this topic is only for tests. Therefore, the Auto Indexing feature is enabled.

2. Create an EMR cluster that resides in the same VPC as the Elasticsearch cluster.

   EMR cluster configuration:

   - EMR Version: Select EMR-3.29.0.
   - Required Services: HDFS (2.8.5) is one of the required services. Default settings are retained for other services.

   For more information, see Create a cluster.

> **Notice** By default, 0.0.0.0/0 is specified in the private IP address whitelist of the Elasticsearch cluster. You can view the whitelist configuration on the cluster security configuration page. If the default setting is not used, you must add the private IP address of the EMR cluster to the whitelist.
>
> - For more information about how to obtain the private IP address of the EMR cluster, see View the cluster list and cluster details.
> - For more information about how to configure the private IP address whitelist of the Elasticsearch cluster, see Configure a public or private IP address whitelist for an Elasticsearch cluster. The IP addresses in the whitelist can be used to access the Elasticsearch cluster over a VPC.

3. Prepare JSON-formatted test data and write the data to the *map.json* file. Upload the file to the */tmp/hadoop-es* directory of HDFS.

   The following test data is used in this topic:

```
{"id": 1, "name": "zhangsan", "birth": "1990-01-01", "addr": "No.969, wenyixi Rd, yuhan
g, hangzhou"}
{"id": 2, "name": "lisi", "birth": "1991-01-01", "addr": "No.556, xixi Rd, xihu, hangzh
ou"}
{"id": 3, "name": "wangwu", "birth": "1992-01-01", "addr": "No.699 wangshang Rd, binjia
ng, hangzhou"}
```

4. Prepare a Java environment. The JDK version must be 1.8.0 or later.

Step 1: Upload the ES-Hadoop JAR package to HDFS

1. Download an ES-Hadoop package that is compatible with the version of the Elasticsearch cluster.

   The elasticsearch-hadoop-6.7.0.zip package is used in this topic.

2. Log on to the EMR console and obtain the IP address of the master node of the EMR cluster. Then, use SSH to log on to the Elastic Compute Service (ECS) instance that is indicated by the IP address.

   For more information, see Log on to a cluster.

3. Upload the elasticsearch-hadoop-6.7.0.zip package to the master node in the EMR cluster. Decompress the package to obtain the elasticsearch-hadoop-6.7.0.jar file.

4. Create an HDFS directory and upload the elasticsearch-hadoop-6.7.0.jar file to the directory.

```
hadoop fs -mkdir /tmp/hadoop-es
hadoop fs -put elasticsearch-hadoop-6.7.0/dist/elasticsearch-hadoop-6.7.0.jar /tmp/hado
op-es
```

Step 2: Configure POM dependencies

Create a Java Maven project and add the following POM dependencies to the pom.xml file of the project.

```
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-shade-plugin</artifactId>
            <version>2.4.1</version>
```

```
            <executions>
                <execution>
                    <phase>package</phase>
                    <goals>
                        <goal>shade</goal>
                    </goals>
                    <configuration>
                        <transformers>
                            <transformer
                                    implementation="org.apache.maven.plugins.shade.resource
.ManifestResourceTransformer">
                                <mainClass>WriteToEsWithMR</mainClass>
                            </transformer>
                        </transformers>
                    </configuration>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>
<dependencies>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-hdfs</artifactId>
        <version>2.8.5</version>
    </dependency>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-mapreduce-client-jobclient</artifactId>
        <version>2.8.5</version>
    </dependency>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-common</artifactId>
        <version>2.8.5</version>
    </dependency>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-auth</artifactId>
        <version>2.8.5</version>
    </dependency>
    <dependency>
        <groupId>org.elasticsearch</groupId>
        <artifactId>elasticsearch-hadoop-mr</artifactId>
        <version>6.7.0</version>
    </dependency>
    <dependency>
        <groupId>commons-httpclient</groupId>
        <artifactId>commons-httpclient</artifactId>
        <version>3.1</version>
    </dependency>
</dependencies>
```

> **Notice** Make sure that the versions of POM dependencies are consistent with those of the related Alibaba Cloud services. For example, the version of elasticsearch-hadoop-mr is consistent with that of Alibaba Cloud Elasticsearch, and the version of hadoop-hdfs is consistent with that of HDFS.

Step 3: Compile code and run a MapReduce task

1. Compile code.

   The following code reads data from the JSON files in the */tmp/hadoop-es* directory of HDFS. The code also writes each row of data in these JSON files as a document to the Elasticsearch cluster. Data write is finished by EsOutputFormat in the map stage.

```
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.util.GenericOptionsParser;
import org.elasticsearch.hadoop.mr.EsOutputFormat;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;
public class WriteToEsWithMR extends Configured implements Tool {
    public static class EsMapper extends Mapper<Object, Text, NullWritable, Text> {
        private Text doc = new Text();
        @Override
        protected void map(Object key, Text value, Context context) throws IOException,
InterruptedException {
            if (value.getLength() > 0) {
                doc.set(value);
                System.out.println(value);
                context.write(NullWritable.get(), doc);
            }
        }
    }
    public int run(String[] args) throws Exception {
        Configuration conf = new Configuration();
        String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
        conf.setBoolean("mapreduce.map.speculative", false);
        conf.setBoolean("mapreduce.reduce.speculative", false);
        conf.set("es.nodes", "es-cn-4591jumei000u****.elasticsearch.aliyuncs.com");
        conf.set("es.port","9200");
        conf.set("es.net.http.auth.user", "elastic");
        conf.set("es.net.http.auth.pass", "xxxxxx");
        conf.set("es.nodes.wan.only", "true");
        conf.set("es.nodes.discovery","false");
        conf.set("es.input.use.sliced.partitions","false");
        conf.set("es.resource", "maptest/_doc");
        conf.set("es.input.json", "true");
```

```
        Job job = Job.getInstance(conf);
        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(EsOutputFormat.class);
        job.setMapOutputKeyClass(NullWritable.class);
        job.setMapOutputValueClass(Text.class);
        job.setJarByClass(WriteToEsWithMR.class);
        job.setMapperClass(EsMapper.class);
        FileInputFormat.setInputPaths(job, new Path(otherArgs[0]));
        return job.waitForCompletion(true) ? 0 : 1;
    }
    public static void main(String[] args) throws Exception {
        int ret = ToolRunner.run(new WriteToEsWithMR(), args);
        System.exit(ret);
    }
}
```

ES-Hadoop parameters

Parameter	Default value	Description
es.nodes	localhost	The endpoint that is used to access the Elasticsearch cluster. We recommend that you use the internal endpoint. You can obtain the internal endpoint on the Basic Information page of the Elasticsearch cluster. For more information, see View the basic information of a cluster.
es.port	9200	The port number that is used to access the Elasticsearch cluster.
es.net.http.auth.user	elastic	The username that is used to access the Elasticsearch cluster. **Note** If you use the elastic account to access your Elasticsearch cluster and then reset the password of the account, it may require some time for the new password to take effect. During this period, you cannot use the elastic account to access the cluster. Therefore, we recommend that you do not use the elastic account to access an Elasticsearch cluster. You can log on to the Kibana console and create a user with the required role to access an Elasticsearch cluster. For more information, see Use the RBAC mechanism provided by Elasticsearch X-Pack to implement access control.
es.net.http.auth.pass	/	The password that is used to access the Elasticsearch cluster.

Parameter	Default value	Description
es.nodes.wan.only	false	Specifies whether to enable node sniffing when the Elasticsearch cluster uses a virtual IP address for connections. Valid values: ○ true: indicates that node sniffing is enabled. ○ false: indicates that node sniff is disabled.
es.nodes.discovery	true	Specifies whether to prohibit the node discovery mechanism. Valid values: ○ true: indicates that the node discovery mechanism is prohibited. ○ false: indicates that the node discovery mechanism is not prohibited. **Notice** If you use Alibaba Cloud Elasticsearch, you must set this parameter to false.
es.input.use.sliced.partitions	true	Specifies whether to use partitions. Valid values: ○ true: indicates that partitions are used. In this case, more time may be required for the index read-ahead phase. The time required for this phase may be longer than the time required for data queries. To improve query efficiency, we recommend that you set this parameter to false. ○ false: indicates that partitions are not used.
es.index.auto.create	true	Specifies whether the system creates an index in the Elasticsearch cluster when you use ES-Hadoop to write data to the cluster. Valid values: ○ true: indicates that the system creates an index in the Elasticsearch cluster. ○ false: indicates that the system does not create an index in the Elasticsearch cluster.
es.resource	/	The name and type of the index on which data read or write operations are performed.
es.input.json	false	Specifies whether the input data is in the JSON format.
es.mapping.names	/	The mappings between the field names in the table and those in the index of the Elasticsearch cluster.

Parameter	Default value	Description
es.read.metadata	false	Specifies whether to include the document metadata such as **_id** in the results. To include the document metadata, set the value to true.

For more information about the configuration items of ES-Hadoop, see open source ES-Hadoop configuration.

2. Compress the code into a JAR package and upload it to an EMR client, such as the master node in the EMR cluster or the gateway cluster that is associated with this EMR cluster.
3. On the EMR client, run the following command to run the MapReduce task:

```
hadoop jar es-mapreduce-1.0-SNAPSHOT.jar /tmp/hadoop-es/map.json
```

> **Note** Replace es-mapreduce-1.0-SNAPSHOT.jar with the name of the uploaded JAR file.

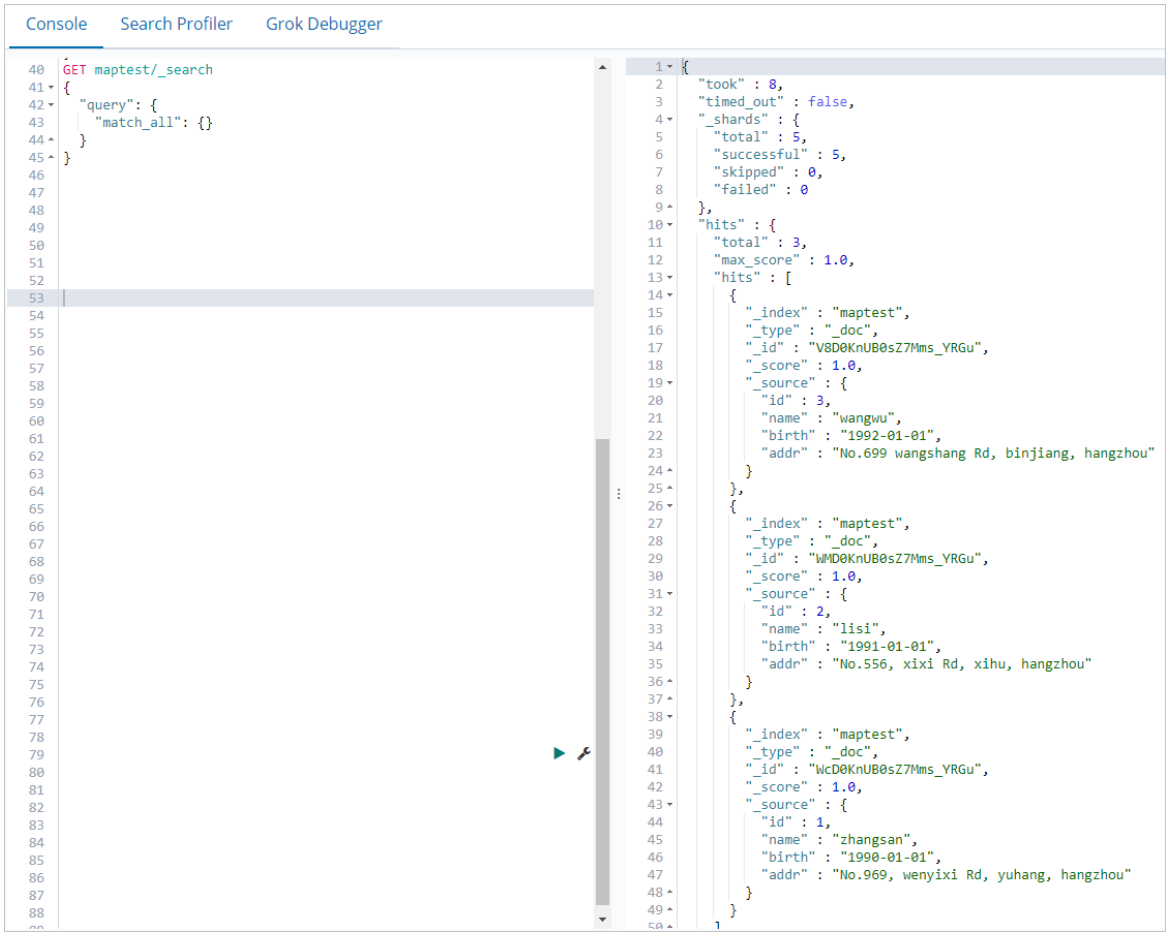
Step 4: Verify the results

1. Log on to the Kibana console of the Elasticsearch cluster.

   For more information, see Log on to the Kibana console.
2. In the left-side navigation pane, click **Dev Tools**.
3. On the **Console** tab of the page that appears, run the following command to query the data that is written by the MapReduce task:

```
GET maptest/_search
{
  "query": {
    "match_all": {}
  }
}
```

If the command is successfully run, the result shown in the following figure is returned.



Summary

This topic describes how to use ES-Hadoop to write data to Elasticsearch by running a MapReduce task in an EMR cluster. You can also run a MapReduce task to read data from Elasticsearch. The configurations for data read operations are similar to those for data write operations. For more information, see Reading data from Elasticsearch in open source Elasticsearch documentation.

6.3. Use ES-Hadoop to enable Apache Spark to write data to and read data from Alibaba Cloud Elasticsearch

Apache Spark is a general-purpose framework for big data computing and has all the computing advantages of Hadoop MapReduce. The difference is that Spark caches data in memory to enable fast iterations of large datasets. This way, data can be directly read from the cache instead of disks. This enables Spark to provide higher processing performance than MapReduce. This topic describes how to enable Spark to write data to and read data from Alibaba Cloud Elasticsearch by using Elasticsearch-Hadoop (ES-Hadoop).

Preparations

1. Create an Alibaba Cloud Elasticsearch cluster and enable the Auto Indexing feature for the cluster.

For more information, see Create an Alibaba Cloud Elasticsearch cluster and Access and configure an Elasticsearch cluster. In this topic, an Elasticsearch V6.7.0 cluster is created.

> **Notice** In a production environment, we recommend that you disable the Auto Indexing feature. You must create an index and configure mappings for the index in advance. The Elasticsearch cluster used in this topic is only for tests. Therefore, the Auto Indexing feature is enabled.

2. Create an E-MapReduce (EMR) cluster in the virtual private cloud (VPC) where the Elasticsearch cluster resides.

   EMR cluster configuration:

   - EMR Version: Select EMR-3.29.0.
   - Required Services: Spark (2.4.5) is one of the required services. Default settings are retained for other services.

   For more information, see Create a cluster.

   > **Notice** By default, 0.0.0.0/0 is specified in the private IP address whitelist of the Elasticsearch cluster. You can view the whitelist configuration on the cluster security configuration page. If the default setting is not used, you must add the private IP address of the EMR cluster to the whitelist.
   >
   > - For more information about how to obtain the private IP address of the EMR cluster, see View the cluster list and cluster details.
   > - For more information about how to configure the private IP address whitelist of the Elasticsearch cluster, see Configure a public or private IP address whitelist for an Elasticsearch cluster. The IP addresses in the whitelist can be used to access the Elasticsearch cluster over a VPC.

3. Prepare a Java environment. The JDK version must be 1.8.0 or later.

Compile and run a Spark job

1. Prepare test data.

   i. Log on to the EMR console and obtain the IP address of the master node of the EMR cluster. Then, use SSH to log on to the Elastic Compute Service (ECS) instance that is indicated by the IP address.

   For more information, see Log on to a cluster.

   ii. Write the test data to a file.

   In this example, the following JSON-formatted test data is written to the *http_log.txt* file:

```
{"id": 1, "name": "zhangsan", "birth": "1990-01-01", "addr": "No.969, wenyixi Rd, yuhang, hangzhou"}
{"id": 2, "name": "lisi", "birth": "1991-01-01", "addr": "No.556, xixi Rd, xihu, hangzhou"}
{"id": 3, "name": "wangwu", "birth": "1992-01-01", "addr": "No.699 wangshang Rd, binjiang, hangzhou"}
```

iii. Run the following command to upload the file to the *tmp/hadoop-es* directory on the master node of the EMR cluster:

```
hadoop fs -put http_log.txt /tmp/hadoop-es
```

2. Add POM dependencies.

   Create a Java Maven project and add the following POM dependencies to the pom.xml file of the project:

```
<dependencies>
    <dependency>
        <groupId>org.apache.spark</groupId>
        <artifactId>spark-core_2.12</artifactId>
        <version>2.4.5</version>
    </dependency>
    <dependency>
        <groupId>org.apache.spark</groupId>
        <artifactId>spark-sql_2.11</artifactId>
        <version>2.4.5</version>
    </dependency>
    <dependency>
        <groupId>org.elasticsearch</groupId>
        <artifactId>elasticsearch-spark-20_2.11</artifactId>
        <version>6.7.0</version>
    </dependency>
</dependencies>
```

   Notice Make sure that the versions of POM dependencies are consistent with those of the related Alibaba Cloud services. For example, the version of elasticsearch-spark-20_2.11 is consistent with that of your Elasticsearch cluster, and the version of spark-core_2.12 is consistent with that of HDFS.

3. Compile code.

i. Write data

The following sample code is used to write the test data to the company index of the Elasticsearch cluster:

```
import java.util.Map;
import java.util.concurrent.atomic.AtomicInteger;
import org.apache.spark.SparkConf;
import org.apache.spark.SparkContext;
import org.apache.spark.api.java.JavaRDD;
import org.apache.spark.api.java.function.Function;
import org.apache.spark.sql.Row;
import org.apache.spark.sql.SparkSession;
import org.elasticsearch.spark.rdd.api.java.JavaEsSpark;
import org.spark_project.guava.collect.ImmutableMap;
public class SparkWriteEs {
    public static void main(String[] args) {
        SparkConf conf = new SparkConf();
        conf.setAppName("Es-write");
        conf.set("es.nodes", "es-cn-n6w1o1x0w001c****.elasticsearch.aliyuncs.com");
        conf.set("es.net.http.auth.user", "elastic");
        conf.set("es.net.http.auth.pass", "xxxxxx");
        conf.set("es.nodes.wan.only", "true");
        conf.set("es.nodes.discovery","false");
        conf.set("es.input.use.sliced.partitions","false");
        SparkSession ss = new SparkSession(new SparkContext(conf));
        final AtomicInteger employeesNo = new AtomicInteger(0);
        //Replace /tmp/hadoop-es/http_log.txt with the actual path of your test dat
a.
        JavaRDD<Map<Object, ?>> javaRDD = ss.read().text("/tmp/hadoop-es/http_log.t
xt")
                .javaRDD().map((Function<Row, Map<Object, ?>>) row -> ImmutableMap.
of("employees"   employeesNo.getAndAdd(1), row.mkString()));
        JavaEsSpark.saveToEs(javaRDD, "company/_doc");
    }
}
```

ii. Read data

The following sample code is used to read and display the test data that is written to the Elasticsearch cluster:

```
import org.apache.spark.SparkConf;
import org.apache.spark.api.java.JavaPairRDD;
import org.apache.spark.api.java.JavaSparkContext;
import org.elasticsearch.spark.rdd.api.java.JavaEsSpark;
import  java.util.Map;
public class ReadES {
    public static void main(String[] args) {
        SparkConf  conf = new SparkConf().setAppName("readEs").setMaster("local[*]"
)
                .set("es.nodes", "es-cn-n6w1o1x0w001c****.elasticsearch.aliyuncs.co
m")
                .set("es.port", "9200")
                .set("es.net.http.auth.user", "elastic")
                .set("es.net.http.auth.pass", "xxxxxx")
                .set("es.nodes.wan.only", "true")
                .set("es.nodes.discovery","false")
                .set("es.input.use.sliced.partitions","false")
                .set("es.resource", "company/_doc")
                .set("es.scroll.size","500");
        JavaSparkContext sc = new JavaSparkContext(conf);
        JavaPairRDD<String, Map<String, Object>> rdd = JavaEsSpark.esRDD(sc);
        for ( Map<String, Object> item : rdd.values().collect()) {
            System.out.println(item);
        }
        sc.stop();
    }
}
```

Parameters

Parameter	Default value	Description
es.nodes	localhost	The endpoint that is used to access the Elasticsearch cluster. We recommend that you use the internal endpoint. You can obtain the internal endpoint on the Basic Information page of the Elasticsearch cluster. For more information, see View the basic information of a cluster.
es.port	9200	The port number that is used to access the Elasticsearch cluster.

Parameter	Default value	Description
es.net.http.auth.user	elastic	The username that is used to access the Elasticsearch cluster. **Note** If you use the elastic account to access your Elasticsearch cluster and then reset the password of the account, it may require some time for the new password to take effect. During this period, you cannot use the elastic account to access the cluster. Therefore, we recommend that you do not use the elastic account to access an Elasticsearch cluster. You can log on to the Kibana console and create a user with the required role to access an Elasticsearch cluster. For more information, see Use the RBAC mechanism provided by Elasticsearch X-Pack to implement access control.
es.net.http.auth.pass	/	The password that corresponds to the elastic username. The password is specified when you create the Elasticsearch cluster. If you forget the password, you can reset it. For more information, see Reset the access password for an Elasticsearch cluster.
es.nodes.wan.only	false	Specifies whether to enable node sniffing when the Elasticsearch cluster uses a virtual IP address for connections. Valid values: ○ true: indicates that node sniffing is enabled. ○ false: indicates that node sniff is disabled.
es.nodes.discovery	true	Specifies whether to prohibit the node discovery mechanism. Valid values: ○ true: indicates that the node discovery mechanism is prohibited. ○ false: indicates that the node discovery mechanism is not prohibited. **Notice** If you use Alibaba Cloud Elasticsearch, you must set this parameter to false.

Parameter	Default value	Description
es.input.use.sliced.partitions	true	Specifies whether to use partitions. Valid values: ○ true: indicates that partitions are used. In this case, more time may be required for the index read-ahead phase. The time required for this phase may be longer than the time required for data queries. To improve query efficiency, we recommend that you set this parameter to false. ○ false: indicates that partitions are not used.
es.index.auto.create	true	Specifies whether the system creates an index in the Elasticsearch cluster when you use ES-Hadoop to write data to the cluster. Valid values: ○ true: indicates that the system creates an index in the Elasticsearch cluster. ○ false: indicates that the system does not create an index in the Elasticsearch cluster.
es.resource	/	The name and type of the index on which data read or write operations are performed.
es.mapping.names	/	The mappings between the field names in the table and those in the index of the Elasticsearch cluster.

For more information about the configuration items of ES-Hadoop, see open source ES-Hadoop configuration.

4. Compress the code into a JAR package and upload it to an EMR client, such as the master node in the EMR cluster or the gateway cluster that is associated with this EMR cluster.
5. On the EMR client, run the following Spark jobs:
   - Write data

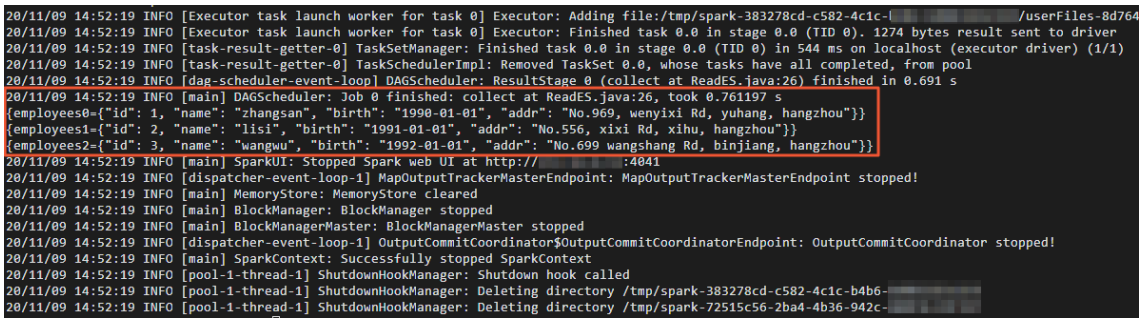
```
cd /usr/lib/spark-current
./bin/spark-submit  --master yarn --executor-cores 1 --class "SparkWriteEs" /root/spark_es.jar
```

> **Notice** Replace */root/spark_es.jar* with the path to which you have uploaded your JAR package.

   - Read data

```
cd /usr/lib/spark-current
./bin/spark-submit  --master yarn --executor-cores 1 --class "ReadES"  /root/spark_es.jar
```

After the data is read, the result shown in the following figure is returned.



Verify results

1. Log on to the Kibana console of the Elasticsearch cluster.

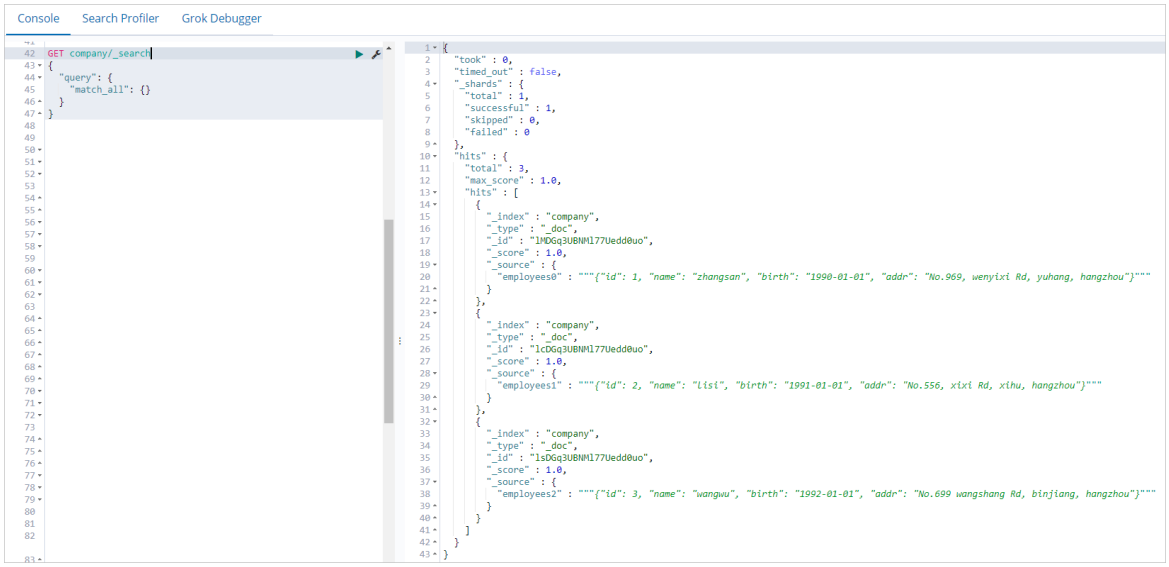
   For more information, see Log on to the Kibana console.

2. In the left-side navigation pane, click **Dev Tools**.

3. On the **Console** tab of the page that appears, run the following command to query the data that is written by a Spark job:

```
GET company/_search
{
  "query": {
    "match_all": {}
  }
}
```

If the command is successfully run, the result shown in the following figure is returned.



Summary

This topic describes how to use ES-Hadoop to write data to and read data from Alibaba Cloud Elasticsearch by running Spark jobs in an EMR cluster. After ES-Hadoop is integrated with Spark, ES-Hadoop supports Spark datasets, resilient distributed datasets (RDDs), Spark Streaming, Scala, and Spark SQL. You can configure ES-Hadoop based on your requirements. For more information, see Apache Spark support.

7.Log synchronization and analysis

7.1. Overview of log synchronization and analysis

When applications are running, various types of logs are generated. You can collect the desired log data and transfer the collected data to Alibaba Cloud Elasticsearch. Then, you can query and analyze the data. This topic provides an overview of best practices for log synchronization and analysis to meet your business requirements in various scenarios.

Best practice	Description
Use Filebeat to collect Apache log data	The typical log collection mode of Elastic Stack is used. Use Alibaba Cloud Filebeat to collect Apache log data. Then, use Alibaba Cloud Logstash to filter the collected data and transfer the processed data to an Alibaba Cloud Elasticsearch cluster for queries and analysis.
Use the logstash-input-sls plug-in to obtain logs from Log Service	Use the logstash-input-sls plug-in to obtain logs from Log Service and transfer them to Alibaba Cloud Elasticsearch for queries and analysis.
Use user-created Filebeat to collect MySQL logs	Use self-managed Filebeat to collect and send MySQL logs to Alibaba Cloud Elasticsearch. Then, query, analyze, and present these logs in the Kibana console in a visualized manner.
Use Alibaba Cloud Elasticsearch and Rsbeat to analyze Redis slow logs in real time	Use Rsbeat to collect and send Redis slow logs to Alibaba Cloud Elasticsearch. Then, perform graphical analysis on the logs in the Kibana console.

7.2. Use user-created Filebeat to collect MySQL logs

If you want to view and analyze MySQL logs such as slow logs and error logs, you can use Filebeat to collect MySQL logs. Filebeat then sends the logs to Alibaba Cloud Elasticsearch. You can query, analyze, and present these logs in the Kibana console in a visualized manner. This topic describes how to perform the detailed procedure.

Procedure

1. Preparations

   Create an Alibaba Cloud Elasticsearch cluster and an Elastic Compute Service (ECS) instance. The Elasticsearch cluster is used to receive the MySQL logs that are collected by Filebeat. It also provides the Kibana console to query, analyze, and present these logs in a visualized manner. The ECS instance is used to install MySQL and Filebeat.

2. Step 1: Install and configure MySQL

Install MySQL and configure error log files and slow query log files in the MySQL configuration file. Then, Filebeat can collect your desired logs.

3. Step 2: Install and configure Filebeat

   Install Filebeat. Filebeat is used to collect MySQL logs and send the logs to your Elasticsearch cluster. You must enable the MySQL module in Filebeat and specify the URLs that are used to access your Elasticsearch cluster and the Kibana console of the cluster in the Filebeat configuration file.

4. Step 3: Use the Kibana dashboard to present MySQL logs

   Perform a query test and present the error logs and slow query logs that you want to view and analyze on the dashboard of the Kibana console.

Preparations

- Create an Elasticsearch cluster.

  An Elasticsearch V6.7.0 cluster of the Standard Edition is used in this topic. For more information, see Create an Alibaba Cloud Elasticsearch cluster.

- Create an Alibaba Cloud ECS instance.

  An ECS instance that runs CentOS is used in this topic. For more information, see Create an instance by using the wizard.

Step 1: Install and configure MySQL

1. Connect to the ECS instance.

   For more information, see Connect to a Linux instance by using a password.

2. Download and install the MySQL source.

   ```
   wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm
   rpm -ivh mysql-community-release-el7-5.noarch.rpm
   ```

3. Install MySQL.

   ```
   yum install mysql-server
   ```

4. Start MySQL and check its status.

   ```
   systemctl start mysqld
   systemctl status mysqld
   ```

5. Configure error log files and slow query log files in the my.cnf file.

   > **Note** By default, the configuration of log files in MySQL is disabled. You must manually enable the log file configuration. You can also enable temporary slow logs by running a MySQL command.

   i. Open the my.cnf file.

   ```
   vim /etc/my.cnf
   ```

ii. Configure log files.

```
[mysqld]
log_queries_not_using_indexes = 1
slow_query_log=on
slow_query_log_file=/var/log/mysql/slow-mysql-query.log
long_query_time=0
[mysqld_safe]
log-error=/var/log/mysql/mysqld.log
```

Parameter	Description
`log_queries_not_using_indexes`	Specifies whether to record a query for which no indexes are specified as a slow query log. 1: indicates that the system records a query for which no indexes are specified as a slow query log. 0: indicates that the system does not record a query for which no indexes are specified as a slow query log.
`slow_query_log`	Specifies whether to enable slow query logs. on: indicates that slow query logs are enabled. off: indicates that slow query logs are disabled.
`slow_query_log_file`	Specifies the storage path of slow query logs.
`long_query_time`	Specifies the time threshold used to define a slow query log. Unit: seconds. When the query time exceeds the threshold, the MySQL database writes the query into the file that is specified by `slow_query_log_file`. **Notice** For the convenience of the test, the value of this parameter is set to 0. You can specify this parameter based on your business requirements.

iii. Create log files.

```
mkdir /var/log/mysql
touch /var/log/mysql/mysqld.log
touch /var/log/mysql/slow-mysql-query.log
```

Notice MySQL does not automatically create log files. You must manually create the log files.

iv. Grant read and write permissions on the log files to all users.

```
chmod 777 /var/log/mysql/slow-mysql-query.log /var/log/mysql/mysqld.log
```

Step 2: Install and configure Filebeat

1. Log on to the Kibana console of your Elasticsearch cluster.

   For more information, see Log on to the Kibana console.

2. In the Visualize and Explore Data section, click **Logs**.

3. On the page that appears, click **View setup instructions**.
4. On the **Add Data to Kibana** page, click **MySQL logs**.
5. In the Getting Started section, click the **RPM** tab.

   > **Note** The Linux operating system is used in this topic. Therefore, **RPM** is selected. You can select an appropriate installation method based on your operating system.

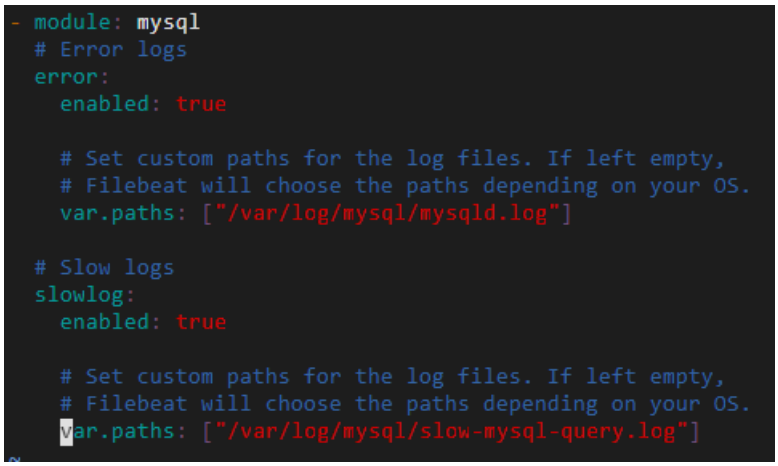
6. Install Filebeat on your ECS instance as prompted.
7. Modify the configuration of the MySQL module and specify the files of the error logs and slow logs that you want to collect.
   i. Enable the MySQL module.

   ```
   sudo filebeat modules enable mysql
   ```

   ii. Open the mysql.yml file.

   ```
   vim /etc/filebeat/modules.d/mysql.yml
   ```

iii. Modify the configuration of the MySQL module.



```
- module: mysql
  # Error logs
  error:
    enabled: true

    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    var.paths: ["/var/log/mysql/mysqld.log"]

  # Slow logs
  slowlog:
    enabled: true

    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    var.paths: ["/var/log/mysql/slow-mysql-query.log"]
```

```
- module: mysql
  # Error logs
  error:
    enabled: true
    var.paths: ["/var/log/mysql/mysqld.log"]
    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    #var.paths:
  # Slow logs
  slowlog:
    enabled: true
    var.paths: ["/var/log/mysql/slow-mysql-query.log"]
    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    #var.paths:
```

Parameter	Description
`enabled`	Set this parameter to `true`.
`var.paths`	Set this parameter to the path of the log file. The path must be the same as the path that is specified in the MySQL configuration file. For more information, see Step 1: Install and configure MySQL.

8. Configure the filebeat.yml file.

i. Open the filebeat.yml file.

```
vim /etc/filebeat/filebeat.yml
```

ii. Modify the configuration of Filebeat modules.

```
#============================== Filebeat modules ===============================

filebeat.config.modules:
  # Glob pattern for configuration loading
  path: /etc/filebeat/modules.d/mysql.yml

  # Set to true to enable config reloading
  reload.enabled: true

  # Period on which files under path should be checked for changes
  reload.period: 1s
```

```
filebeat.config.modules:
  # Glob pattern for configuration loading
  path: /etc/filebeat/modules.d/mysql.yml
  # Set to true to enable config reloading
  reload.enabled: true
  # Period on which files under path should be checked for changes
  reload.period: 1s
```

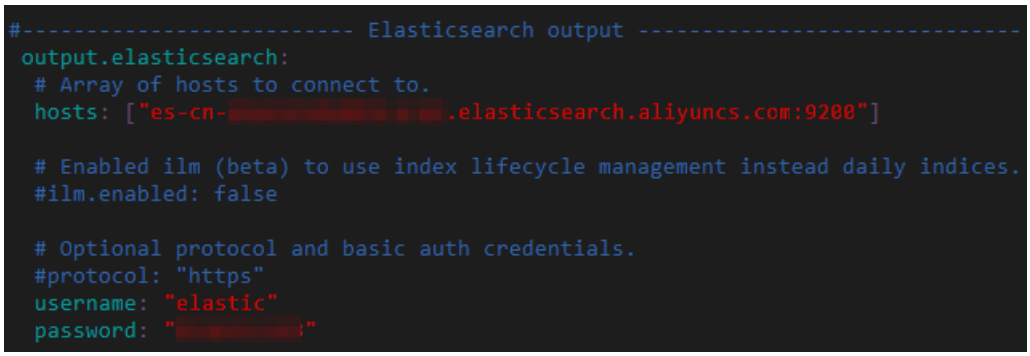
iii. Modify the configuration of Kibana.

```
#============================== Kibana =====================================

# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.
# This requires a Kibana endpoint configuration.
setup.kibana:

  # Kibana Host
  # Scheme and port can be left out and will be set to the default (http and 5601)
  # In case you specify and additional path, the scheme is required: http://localhost:5601/path
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
  host: "https://es-cn-[redacted].kibana.elasticsearch.aliyuncs.com:5601"
```

```
setup.kibana:
host: "https://es-cn-0pp1jxvcl000*****.kibana.elasticsearch.aliyuncs.com:5601"
```

`host`: the URL that is used to access the Kibana console. You can obtain the URL on the Kibana configuration page. For more information, see View the public endpoint of the Kibana console. Specify the URL in the format of `<Public endpoint of the Kibana console>:5601`.

iv. Modify the configuration of the Elasticsearch cluster.




```
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["es-cn-0pp1jxvcl000*****.elasticsearch.aliyuncs.com:9200"]
  # Optional protocol and basic auth credentials.
  #protocol: "https"
  username: "elastic"
  password: "<your_password>"
```

Parameter	Description
`hosts`	The URL that is used to access your Elasticsearch cluster. Specify the URL in the format of `<Internal or public endpoint of the Elasticsearch cluster>:9200`. You can obtain the internal or public endpoint on the Basic Information page of the cluster. For more information, see View the basic information of a cluster. **Note** If the ECS instance and Elasticsearch cluster reside in the same Virtual Private Cloud (VPC), use the internal endpoint. Otherwise, use the public endpoint. If you use the public endpoint to access the Elasticsearch cluster, you must configure a whitelist for access to the Elasticsearch cluster over the Internet. For more information, see Configure a public or private IP address whitelist for an Elasticsearch cluster.
`username`	The username that is used to access the Elasticsearch cluster. Default value: elastic.
`password`	The password that corresponds to the elastic username. The password is specified when you create your Elasticsearch cluster. If you forget the password, you can reset it. For more information about the precautions and procedures for resetting a password, see Reset the access password for an Elasticsearch cluster.

9. Run the following command to start Filebeat.

```
sudo filebeat setup
sudo service filebeat start
```

Step 3: Use the Kibana dashboard to present MySQL logs

1. Restart MySQL in the ECS instance and query logs for tests.

   Run the following command to restart MySQL:

   ```
   systemctl restart mysqld
   ```

2. View the queried logs.

   The following figures show the queried logs.

   Slow logs

   ```
   [root@zl-test003 ~]# tail -50 /var/log/mysql/slow-mysql-query.log
   # Query_time: 0.000385  Lock_time: 0.000000 Rows_sent: 0  Rows_examined: 0
   SET timestamp=1590720469;
   ;
   # User@Host: root[root] @ localhost []  Id:     2
   # Query_time: 0.000138  Lock_time: 0.000000 Rows_sent: 0  Rows_examined: 0
   SET timestamp=1590720469;
   ;
   # User@Host: root[root] @ localhost []  Id:     2
   # Query_time: 0.000214  Lock_time: 0.000048 Rows_sent: 5  Rows_examined: 5
   SET timestamp=1590720469;
   SELECT * from student;
   # Time: 200529 10:47:52
   # User@Host: root[root] @ localhost []  Id:     2
   # Query_time: 0.000231  Lock_time: 0.000117 Rows_sent: 2  Rows_examined: 10
   SET timestamp=1590720472;
   SELECT * from runoob_tbl WHERE runoob_author='菜鸟教程';
   # User@Host: root[root] @ localhost []  Id:     2
   # Query_time: 0.000171  Lock_time: 0.000075 Rows_sent: 2  Rows_examined: 10
   SET timestamp=1590720472;
   SELECT * from runoob_tbl WHERE runoob_author='菜鸟教程';
   # Time: 200529 10:47:53
   # User@Host: root[root] @ localhost []  Id:     2
   # Query_time: 0.000149  Lock_time: 0.000075 Rows_sent: 2  Rows_examined: 10
   SET timestamp=1590720473;
   SELECT * from runoob_tbl WHERE runoob_author='菜鸟教程';
   # Time: 200529 10:47:56
   # User@Host: root[root] @ localhost []  Id:     2
   # Query_time: 0.000039  Lock_time: 0.000000 Rows_sent: 0  Rows_examined: 0
   SET timestamp=1590720476;
   sleep(4);
   ```

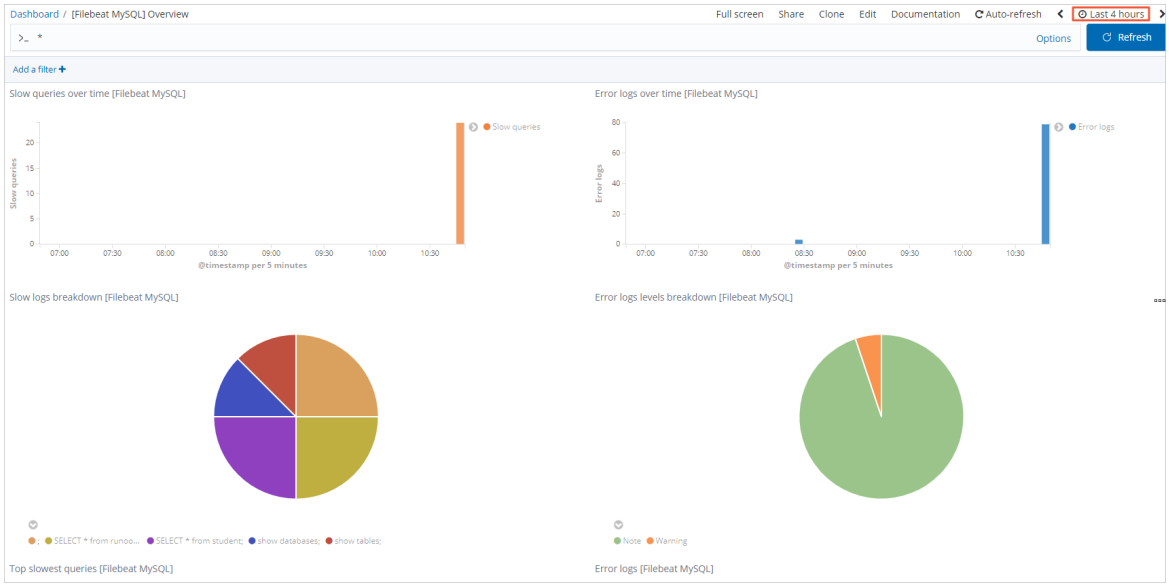
   Error logs

   ```
   [root@zl-test003 ~]# tail -50 /var/log/mysql/mysqld.log
   2020-05-29 10:47:43 4240 [Note] Shutting down plugin 'INNODB_LOCKS'
   2020-05-29 10:47:43 4240 [Note] Shutting down plugin 'INNODB_TRX'
   2020-05-29 10:47:43 4240 [Note] Shutting down plugin 'InnoDB'
   2020-05-29 10:47:43 4240 [Note] InnoDB: FTS optimize thread exiting.
   2020-05-29 10:47:43 4240 [Note] InnoDB: Starting shutdown...
   2020-05-29 10:47:45 4240 [Note] InnoDB: Shutdown completed; log sequence number 1666138
   2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'BLACKHOLE'
   2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'ARCHIVE'
   2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'MRG_MYISAM'
   2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'MyISAM'
   2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'MEMORY'
   2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'CSV'
   2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'sha256_password'
   2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'mysql_old_password'
   2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'mysql_native_password'
   2020-05-29 10:47:45 4240 [Note] Shutting down plugin 'binlog'
   2020-05-29 10:47:45 4240 [Note] /usr/sbin/mysqld: Shutdown complete

   200529 10:47:45 mysqld_safe mysqld from pid file /var/run/mysqld/mysqld.pid ended
   200529 10:47:45 mysqld_safe Logging to '/var/log/mysql/mysqld.log'.
   200529 10:47:45 mysqld_safe Starting mysqld daemon with databases from /var/lib/mysql
   2020-05-29 10:47:45 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_def
   _timestamp server option (see documentation for more details).
   2020-05-29 10:47:45 0 [Note] /usr/sbin/mysqld (mysqld 5.6.48-log) starting as process 21710 ...
   2020-05-29 10:47:45 21710 [Warning] Buffered warning: Changed limits: max_open_files: 1024 (requested 5000)
   ```

3. Log on to the Kibana console of your Elasticsearch cluster.

   For more information, see Log on to the Kibana console.

4. In the left-side navigation pane, click **Dashboard**.
5. On the **Dashboards** page, click **[Filebeat MySQL] Overview**.
6. Select a time range in the upper-right corner and view the logs in the time range.



7.3. Use Alibaba Cloud Elasticsearch and Rsbeat to analyze Redis slow logs in real time

Redis is a widely used key value database that delivers high performance. Redis is single threaded. Inappropriate use of Redis may cause slow queries. Excessive slow queries or a slow query that takes a long time, such as 20 seconds, may block an operation queue or cause services to be unavailable. In this case, you must collect and analyze Redis slow logs in real time to locate and handle exceptions. This topic describes how to use Alibaba Cloud Elasticsearch and Rsbeat to analyze Redis slow logs in real time.

Context

You can use Rsbeat to collect and send Redis slow logs to Elasticsearch. Then, use graphs to analyze the logs in the Kibana console. Terms:

- Elasticsearch: a Lucene-based, distributed, and real-time search and analytics engine. It is an open source product released under the Apache License. Elasticsearch is a popular search engine for enterprises. Elasticsearch provides distributed services and allows you to store, query, and analyze large amounts of datasets in near real time. It is typically used as a basic engine or technology to support complex queries and high-performance applications.

Alibaba Cloud Elasticsearch is compatible with open source Elasticsearch features, such as Security, Machine Learning, Graph, and Application Performance Management (APM). Alibaba Cloud Elasticsearch is released in 5.5.3, 6.3.2, 6.7.0, 6.8.0, and 7.4.0 versions. It supports the commercial plug-in X-Pack and is ideal for scenarios that involve data analytics and searches. Alibaba Cloud Elasticsearch implements enterprise-grade access control, security monitoring and alerting, and automated reporting based on the features of open source Elasticsearch. Alibaba Cloud Elasticsearch is used in this topic, For more information, see What is Alibaba Cloud Elasticsearch?.

- Rsbeat: a data shipper that is used to collect and analyze Redis slow logs. For more information, see open source Rsbeat documentation.
- Redis: an open source, in-memory data structure store. It can be used as a database, cache, and messaging middleware. For more information, see open source Redis documentation.

  ApsaraDB for Redis is a database service that is compatible with native Redis protocols. It supports hybrid storage that combines memory and hard disks. ApsaraDB for Redis provides a high-availability hot standby architecture and can scale to meet requirements for read and write operations that require high performance. ApsaraDB for Redis is used in this topic. For more information, see What is ApsaraDB for Redis?.

Procedure

1. Preparations

   Create an Alibaba Cloud Elasticsearch cluster, an ApsaraDB for Redis instance, and an Elastic Compute Service (ECS) instance in the same virtual private cloud (VPC).
2. Step 1: Configure slow query parameters for the ApsaraDB for Redis instance

   Configure the conditions to generate Redis slow logs and specify the maximum number of slow logs that can be recorded based on your requirements.
3. Step 2: Install and configure Rsbeat

   Install Rsbeat on the ECS instance and specify the ApsaraDB for Redis instance and Alibaba Cloud Elasticsearch cluster you created in the Rsbeat configuration file.
4. Step 3: Analyze the Redis slow logs in the Kibana console by using graphs

   View the details about the Redis slow logs in the Kibana console and analyze the logs as required.

Preparations

1. Create an Alibaba Cloud Elasticsearch cluster and enable the Auto Indexing feature for the cluster.

   For more information, see Create an Alibaba Cloud Elasticsearch cluster and Configure the YML file. An Elasticsearch V6.7.0 cluster of the Standard Edition is used in this topic.
2. Create an ApsaraDB for Redis instance.

   For more information, see Step 1: Create an ApsaraDB for Redis instance. An ApsaraDB for Redis V5.0 instance of the Community Edition is used in this topic. This instance resides in the same VPC as the Elasticsearch cluster. This allows you to access the Elasticsearch cluster over an internal network.
3. Create an ECS instance.

   For more information, see Create an instance by using the wizard. An ECS instance that runs an image of 64-bit CentOS 7.6 is used in this topic. This instance resides in the same VPC as the ApsaraDB for Redis instance and Elasticsearch cluster.
4. Configure a whitelist for access to the ApsaraDB for Redis instance.

Add the internal IP address of the ECS instance to the whitelist of the ApsaraDB for Redis instance. For more information, see Configure whitelists.

Step 1: Configure slow query parameters for the ApsaraDB for Redis instance

1. Log on to the ApsaraDB for Redis console.
2. In the top navigation bar, select a region.
3. On the **Instances** page, find your desired ApsaraDB for Redis instance and click its ID or choose > **Manage** in the **Actions** column.
4. In the left-side navigation pane, click **System Parameters**.
5. In the System Parameters section, find the **slowlog-log-slower-than** and **slowlog-max-len** parameters. Then, modify these parameters based on your requirements.

Parameter	Description	Example
slowlog-log-slower-than	If the runtime of a command exceeds the value of this parameter, the command is defined as a slow query and recorded as a slow log. The runtime does not include the time spent in queuing. Unit: microseconds. Default value: 10000 (10 milliseconds). **Notice** If you set this parameter to a negative number, ApsaraDB for Redis does not record slow queries as slow logs. If you set this parameter to 0, ApsaraDB for Redis records all commands.	This parameter is set to 20000 in this topic. This value indicates that a command whose runtime exceeds 20 milliseconds is recorded as a slow log.
slowlog-max-len	The maximum number of slow query commands that can be recorded as slow logs. If the number of commands that are recorded exceeds the value of this parameter, ApsaraDB for Redis deletes the earliest slow logs.	This parameter is set to 100 in this topic. This value indicates that ApsaraDB for Redis records the latest 100 slow query commands as slow logs.

Step 2: Install and configure Rsbeat

1. Connect to the ECS instance.

   For more information, see Connect to an ECS instance.
2. Install Rsbeat.

Rsbeat 5.3.2 is used in this topic.

```
wget https://github.com/Yourdream/rsbeat/archive/master.zip
unzip master.zip
```

3. Modify the configuration of Rsbeat.
   i. Run the following command to open the rsbeat.yml file.

   ```
   cd rsbeat-master
   vim rsbeat.yml
   ```

   ii. Modify the configurations in the rsbeat and output.elasticsearch sections based on the following instructions and save the modifications.

```
############################# Rsbeat ###################################

rsbeat:
  # Defines how often an event is sent to the output
  period: 1s
  redis: ["r-[redacted].redis.rds.aliyuncs.com:6379"]
  slowerThan: 100

#================================ General =====================================

# The name of the shipper that publishes the network data. It can be used to group
# all the transactions sent by a single shipper in the web interface.
#name:

# The tags of the shipper are included in their own field with each
# transaction published.
#tags: ["service-X", "web-tier"]

# Optional fields that you can specify to add additional information to the
# output.
#fields:
#  env: staging

#================================ Outputs =====================================

# Configure what outputs to use when sending the data collected by the beat.
# Multiple outputs may be used.

#-------------------------- Elasticsearch output ------------------------------
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["es-cn-[redacted].elasticsearch.aliyuncs.com:9200"]

  # Optional protocol and basic auth credentials.
  #protocol: "https"
  username: "elastic"
  password: "[redacted]"

  # Overwrite existing template
  template.overwrite: true
```

Configurations in the rsbeat section

Parameter	Description
period	The interval at which Rsbeat sends Redis slow logs to the Elasticsearch cluster.

Parameter	Description
redis	The endpoint that is used to connect to the ApsaraDB for Redis instance. For more information, see View endpoints. **Notice** The password that is used to access the ApsaraDB for Redis instance is not specified in the configuration file of Rsbeat. To enable Rsbeat to access the ApsaraDB for Redis instance, you must enable password-free access after you obtain the endpoint. For more information, see Enable password-free access.
slowerThan	The time that is required to send the `config set slowlog-log-slower-than` command to the Redis server. Unit: microseconds.

Configurations in the output.elasticsearch section

Parameter	Description
hosts	The endpoint that is used to access the Elasticsearch cluster. You can obtain the endpoint on the Basic Information page of the Elasticsearch cluster. For more information, see View the basic information of a cluster.
username	The username that is used to access the Elasticsearch cluster. The default username id elastic.
password	The password that corresponds to the username. The password is specified when you create you Elasticsearch cluster. If you forget the password, you can reset it. For more information about the precautions and procedures for resetting the password, see Reset the access password for an Elasticsearch cluster.
template.overwrite	Specifies whether the Rsbeat-created index template that has the same name as the index template of the Elasticsearch cluster overwrites the index template of the Elasticsearch cluster. Default value: true.

4. Start the Rsbeat service.

```
./rsbeat.linux.amd64 -c rsbeat.yml -e -d "*"
```

Step 3: Analyze the Redis slow logs in the Kibana console by using graphs

1. Log on to the Kibana console of your Elasticsearch cluster.

   For more information, see Log on to the Kibana console.

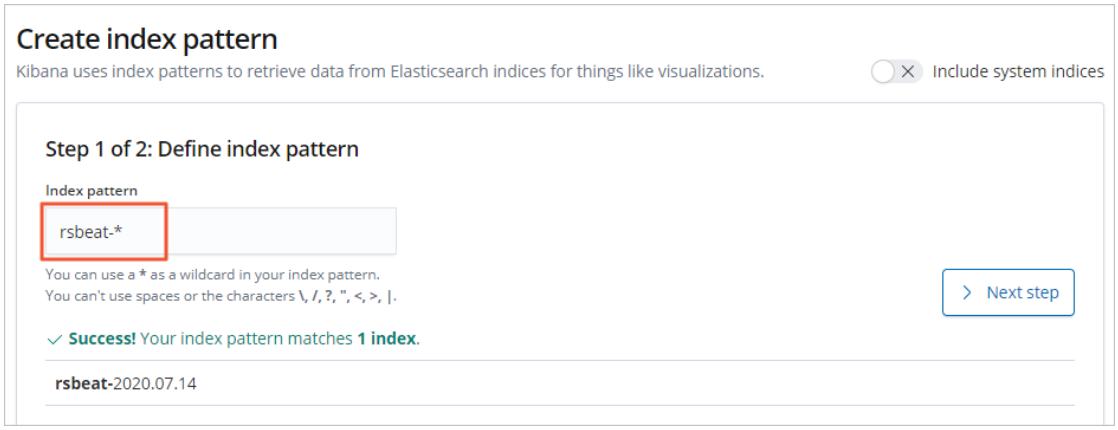
2. Create an index pattern.

   i.

ii.

iii.

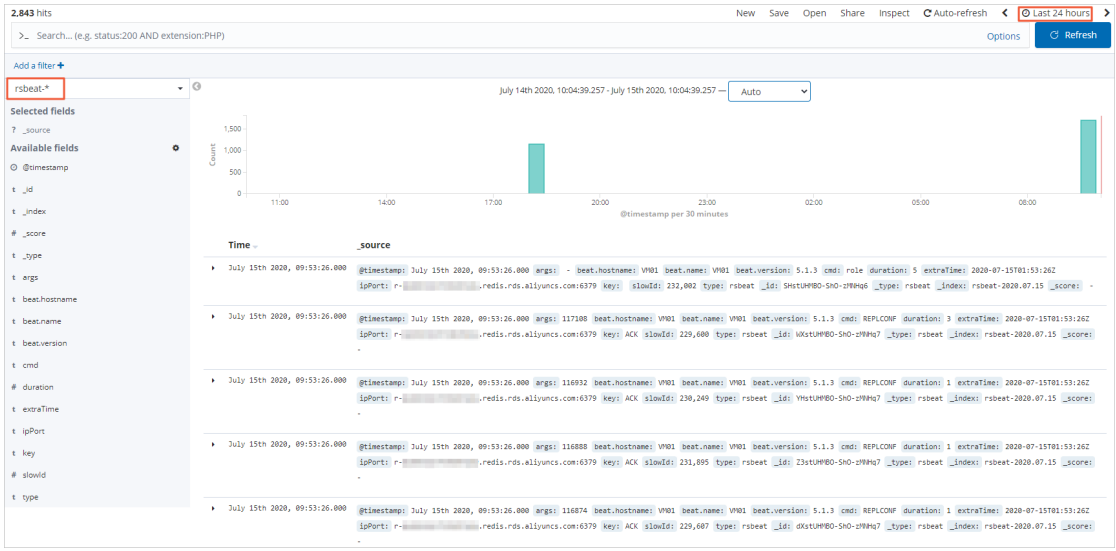
iv. In the section that appears, enter an index pattern name in the **Index pattern** field and click **Next step**.



v.

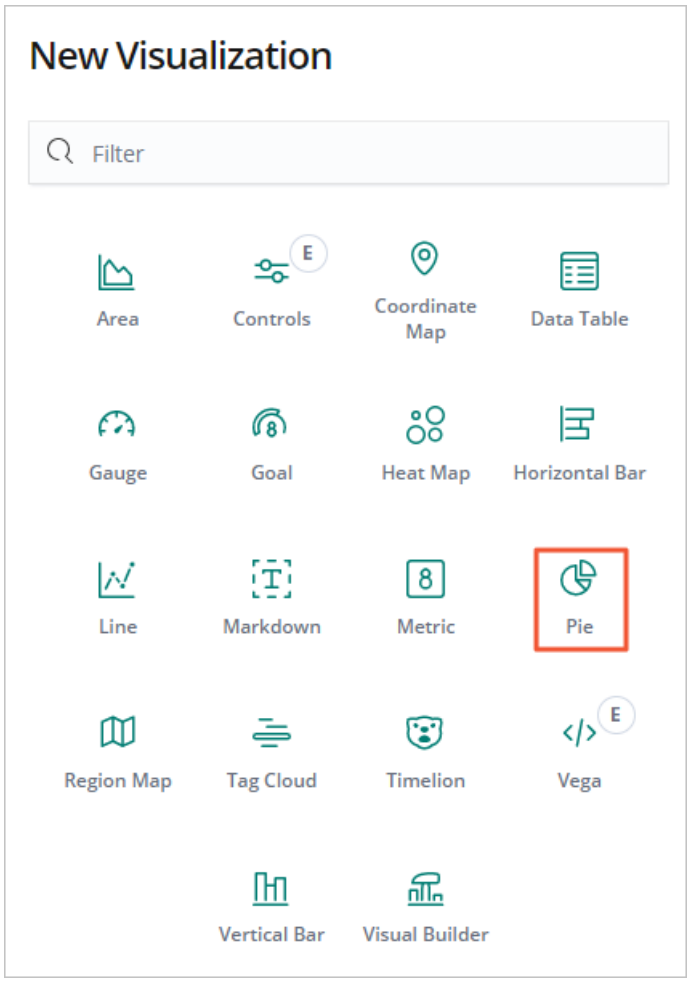
vi.

3. View the details about the Redis slow logs.
   i. In the left-side navigation pane, click **Discover**.
   ii. In the left part of the page, select rsbeat-* from the drop-down list below **Add a filter**.
   iii. In the upper-right corner of the page, select a time range and view the details about the Redis slow logs during the time range.

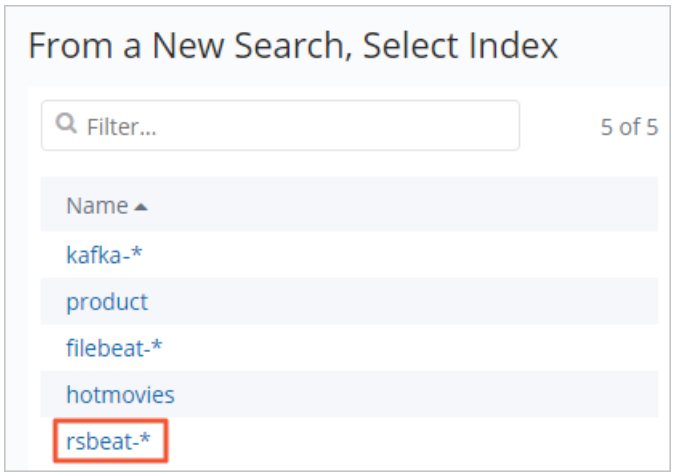


4. Record the top 10 keys with the largest number of Redis slow logs and arrange them in descending order.
   i. In the left-side navigation pane, click **Visualize**.
   ii. On the **Visualize** page, click the + icon.

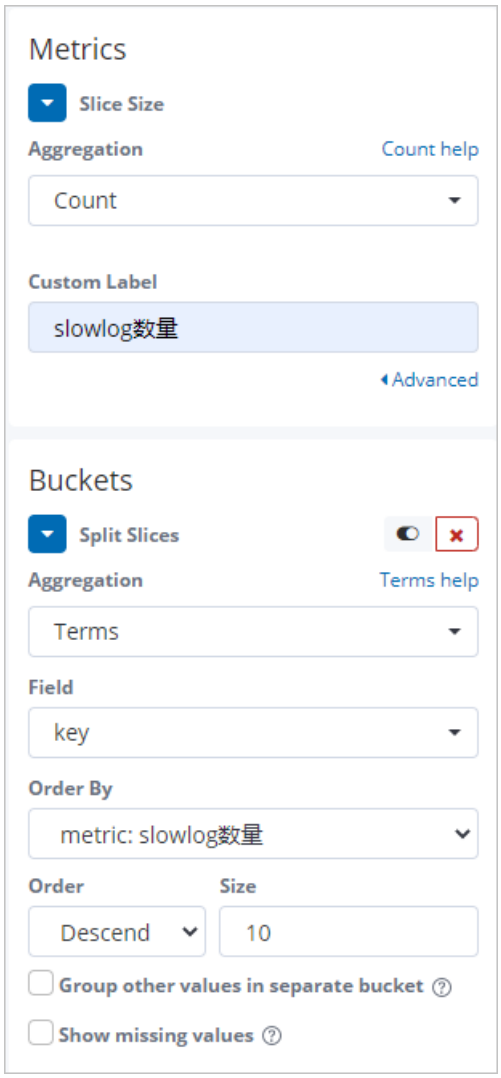
iii. In the **New Visualization** dialog box, click **Pie**.



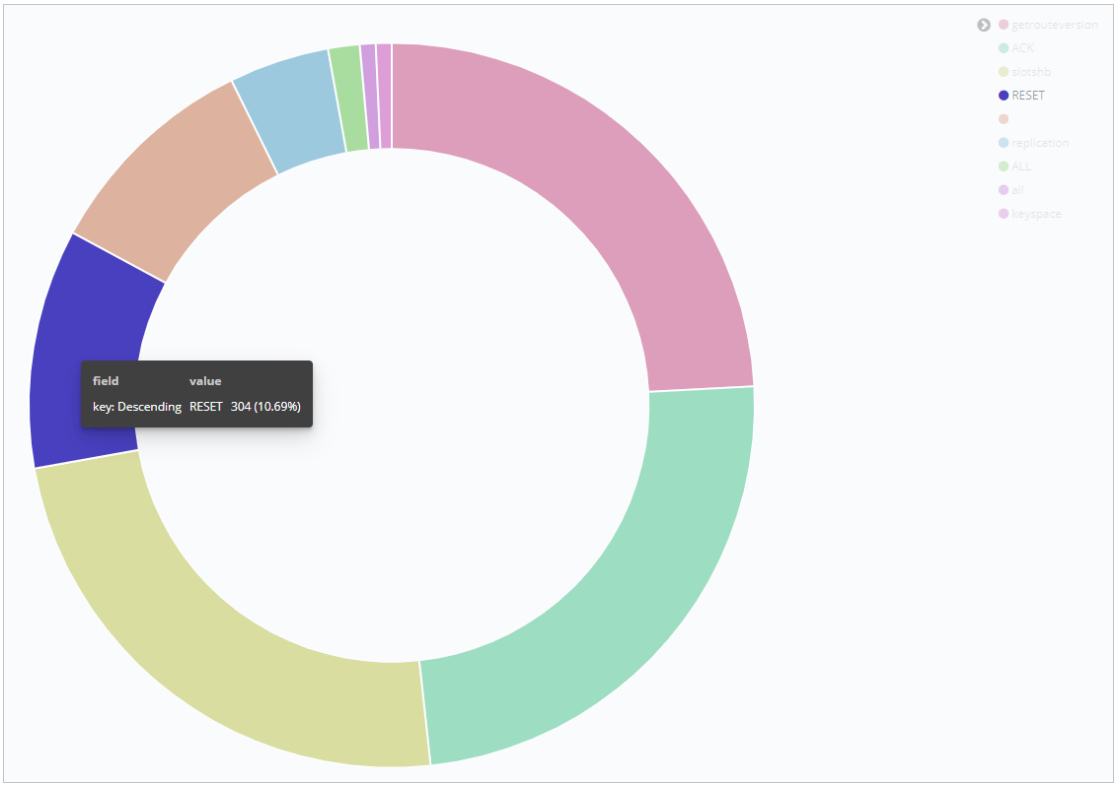
iv. In the From a New Search, Select Index Section, click rsbeat-*.



v. Configure Metrics and Buckets based on the following figure.



vi. Click the icon to view the result.



Note For more information about the usage notes of the Kibana console, see Kibana User Guide.

8.Collect data

8.1. Overview of best practices for server data collection

When applications are running, various types of data are generated, such as log data, system metric data, audit framework data, detection status data, and application performance monitoring (APM) data. You can select a solution based on your business requirements and business environment to collect the desired data and transfer the collected data to Alibaba Cloud Elasticsearch. This topic provides an overview of best practices for server data collection.

Best practice	Description
Data collection for Alibaba Cloud Elasticsearch	You can use one of the following tools to collect data: Beats, Logstash, clients, and Kibana.
Use Filebeat to collect Apache log data	You can use Filebeat to collect log data and use Logstash to filter the collected data and transfer the processed data to Elasticsearch for analysis.
Use Metricbeat to collect system data and NGINX service data	You can use Metricbeat to collect system data and NGINX service data and generate visual charts.
Use Auditbeat to collect system audit data and monitor file changes	You can use Auditbeat to collect data from the Linux audit framework, monitor system file changes, and generate visual charts.
Use Heartbeat to check ICMP and HTTP services	You can use Heartbeat to detect the statuses of Internet Control Message Protocol (ICMP) and HTTP services and generate visual charts.
Use user-created Metricbeat to collect system metrics	You can use Metricbeat to collect the metrics of a machine and send the collected data to Elasticsearch. Then, you can use Kibana to analyze and display the data in charts.
Use SkyWalking to implement end-to-end monitoring on Alibaba Cloud Elasticsearch	You can use SkyWalking to implement end-to-end monitoring on an Elasticsearch cluster and use Kibana to analyze and view the collected monitoring data.
Use Uptime to monitor Alibaba Cloud Elasticsearch clusters in real time	You can use Heartbeat to detect the statuses of the HTTP or HTTPS, TCP, and ICMP services and send the collected data to the Uptime application in Kibana. Then, the Uptime application monitors the availability and response time of applications and services in real time and reports errors before your business is affected.

8.2. Data collection for Alibaba Cloud Elasticsearch

This topic describes the methods that are used to collect and send data from a variety of data sources to an Alibaba Cloud Elasticsearch cluster.

Background information

Elasticsearch is widely used for data search and analytics. Developers and communities use Elasticsearch in a wide range of scenarios. The scenarios include application search, website search, logging, infrastructure monitoring, application performance monitoring (APM), and security analytics. Solutions for these scenarios are provided free of charge. However, before developers use these solutions, they must import the required data into Elasticsearch.

This topic provides the following common methods to collect data:

- Elastic Beats
- Logstash
- Clients
- Kibana

Elasticsearch provides a flexible RESTful API to communicate with client applications. You can call this RESTful API to collect, search for, and analyze data. You can also use the API to manage Elasticsearch clusters and indexes on the clusters.

Elastic Beats

Elastic Beats consists of a set of lightweight data shippers that can transfer data to Elasticsearch. These shippers do not incur a number of runtime overheads. Beats can run and collect data on devices that do not have sufficient hardware resources. The devices include IoT devices, edge devices, or embedded devices. If you want to collect data but do not have sufficient resources to run a resource-intensive data shipper, we recommend that you use Beats. Based on data collected by Beats from all Internet-connected devices, you can quickly identify exceptions, such as system errors and security issues. Then, you can take measures to deal with these exceptions.

Beats can also be used in systems that have sufficient hardware resources.

You can use Beats to collect various types of data.

- Filebeat

  Filebeat can be used to read, preprocess, and transfer data from files. In most cases, you can use Filebeat to read data from log files. Filebeat can also be used to read data from non-binary files. You can use Filebeat to read data from other data sources, such as TCP, UDP, container, Redis, and syslogs. Based on various modules, Filebeat provides an easy way to collect logs of common applications, such as Apache, MySQL, and Kafka. Then, Filebeat parses the logs to obtain the required data.

- Metricbeat

  Metricbeat can be used to collect and preprocess system and service metrics. System metrics include information about running processes, CPU utilization, memory usage, disk usage, and network usage. Based on various modules, Metricbeat can be used to collect data from various services, such as Kafka, Palo Alto Networks, and Redis.

- Packetbeat

  Packetbeat can be used to collect and preprocess real-time network data. You can use Packetbeat for security analytics, application monitoring, and performance analytics. Packetbeat supports the following protocols: DHCP, DNS, HTTP, MongoDB, NFS, and TLS.

- Winlogbeat

Winlogbeat can be used to capture event logs from Windows operating systems. The event logs include application, hardware, security, and system events.

- Auditbeat

  Auditbeat can be used to detect changes to critical files and collect audit events from the Linux audit framework. In most cases, Auditbeat is used for security analytics.

- Heartbeat

  Heartbeat can be used to check the availability of your system and services by probing. Heartbeat applies to many scenarios, such as infrastructure monitoring and security analytics. Heartbeat supports ICMP, TCP, and HTTP.

- Functionbeat

  Functionbeat can be used to collect logs and metrics from serverless environments such as AWS Lambda.

For more information about how to use Metricbeat, see Use user-created Metricbeat to collect system metrics. Use other shippers in a similar way.

Logstash

Logstash is a powerful and flexible tool that is used to read, process, and transfer all types of data. Logstash provides a variety of features and has high requirements for device performance. Beats does not support some features provided by Logstash, or it is costly to use Beats for some features. For example, it is costly to use Beats to enrich documents by searching for data in external data sources. Logstash has higher requirements for hardware resources than Beats. Therefore, Logstash cannot be deployed on devices whose hardware resources cannot meet the minimum requirements. If Beats is not qualified for specific scenarios, use Logstash instead.

In most cases, Beats and Logstash work collaboratively. Specifically, use Beats to collect data and Logstash to process data.

Alibaba Cloud Elasticsearch integrates the Logstash service. Alibaba Cloud Logstash is a server-side data processing pipeline. It is compatible with all the capabilities of open source Logstash. Alibaba Cloud Logstash can be used to dynamically collect data from multiple data sources at the same time and transform and store collected data to a specified location. Alibaba Cloud Logstash can be used to process and transform all types of events by using input, filter, and output plug-ins.

Logstash data processing pipelines are used to run tasks. Each pipeline consists of at least one input plug-in, one filter plug-in, and one output plug-in.

- Input plug-ins

  Input plug-ins can be used to read data from different data sources. The supported data sources include files, HTTP, IMAP, JDBC, Kafka, syslogs, TCP, and UDP.

- Filter plug-ins

  Filter plug-ins can process and enrich data by using multiple methods. In most cases, filter plug-ins first parse unstructured log data and transform the data into structured data. Logstash provides the Grok filter plug-in to parse regular expressions, CSV data, JSON data, key-value pairs, delimited unstructured data, and complex unstructured data. Logstash also provides other filter plug-ins to enrich data. The plug-ins are used to query DNS records, add the locations of IP addresses, or search for custom directories or Elasticsearch indexes. Additional filter plug-ins, such as mutate, allow you to perform diverse data transformations. For example, you can rename, delete, and copy data fields and values.

- Output plug-ins

  Output plug-ins can be used to write the parsed and enriched data into data sinks. These plug-ins are used in the final stage of Logstash pipelines. Multiple types of output plug-ins are available. However, this topic focuses on the Elasticsearch output plug-in. This plug-in can be used to send data collected from a variety of data sources to an Elasticsearch cluster.

The following section describes a sample Logstash pipeline. It can be used to complete the following operations:

- Read the Elastic Blogs RSS feed.
- Preprocess the data by copying or renaming fields and removing special characters and HTML tags.
- Send the data to Elasticsearch.

1. Configure a Logstash pipeline.

```
input {
  rss {
    url => "/blog/feed"
    interval => 120
  }
}
filter {
  mutate {
    rename => [ "message", "blog_html" ]
    copy => { "blog_html" => "blog_text" }
    copy => { "published" => "@timestamp" }
  }
  mutate {
    gsub => [
      "blog_text", "<.*?>", "",
      "blog_text", "[\n\t]", " "
    ]
    remove_field => [ "published", "author" ]
  }
}
output {
  stdout {
    codec => dots
  }
  elasticsearch {
    hosts => [ "https://<your-elsaticsearch-url>" ]
    index => "elastic_blog"
    user => "elastic"
    password => "<your-elasticsearch-password>"
  }
}
```

   Set `hosts` to a value in the format of `<Internal endpoint of your Elasticsearch cluster>:9200`. Set `password` to the password that is used to access the Elasticsearch cluster.

2. In the Kibana console, view the migrated index data.

```
POST elastic_blog/_search
```

   For more information, see Step 3: View synchronization results.

Clients

You can use Elasticsearch clients to integrate data collection code with tailored application code. These clients are libraries that abstract low-level details of the data collection. They allow you to focus on specific operations that are related to your application. Elasticsearch supports multiple programming languages for clients, such as Java, JavaScript, Go, .NET, PHP, Perl, Python, and Ruby. For more information about the programming languages and the details and sample code of your selected language, see Elasticsearch Clients.

If the programming language of your application is not included in the preceding supported languages, obtain the required information from Community Contributed Clients.

Kibana

We recommend that you use the Kibana console to develop and debug Elasticsearch requests. Kibana provides all features of the RESTful API in Elasticsearch and abstracts the technical details of underlying HTTP requests. You can use Kibana to add original JSON documents to an Elasticsearch cluster.

```
PUT my_first_index/_doc/1
{
    "title" :"How to Ingest Into Elasticsearch Service",
    "date" :"2019-08-15T14:12:12",
    "description" :"This is an overview article about the various ways to ingest into Elast
icsearch Service"
}
```

Note In addition to Kibana, you can use other tools to communicate with Elasticsearch and collect documents by calling the RESTful API. For example, you can use cURL to develop and debug Elasticsearch requests or integrate tailored scripts.

Summary

Multiple methods are provided to collect and send data from a variety of data sources to Elasticsearch. You must select the most suitable method based on your business scenarios, requirements, and operating systems.

- Beats data shippers are convenient, lightweight, and out-of-the-box. They can be used to collect data from various data sources. Modules that are packaged with Beats provide the configurations of data acquisition, parsing, indexing, and visualization for many common databases, operating systems, containers, web servers, and caches. These modules allow you to create a dashboard for your data within five minutes. Beats data shippers are most suited to embedded devices that do not have sufficient resources, such as IoT devices or firewalls.
- Logstash is a flexible tool to read, transform, and transfer data. It provides various input, filter, and output plug-ins. If Beats cannot meet the requirements for specific scenarios, you can use Beats to collect data, use Logstash to process data, and then transfer the processed data to Elasticsearch.
- To collect data from applications, we recommend that you use clients that are supported by open source Elasticsearch.
- To develop or debug Elasticsearch requests, we recommend that you use Kibana.

References

- How to ingest data into Elasticsearch Service

- Should I use Logstash or Elasticsearch ingest nodes?
- Get System Logs and Metrics into Elasticsearch with Beats System Modules

8.3. Use user-created Metricbeat to collect system metrics

Use Metricbeat to collect the metrics of a machine

If you want to view and analyze the metrics of a machine, you can use Metricbeat to collect the metrics. Then, Metricbeat sends the collected data to an Alibaba Cloud Elasticsearch cluster. You can view the data on the related dashboard in the Kibana console of the cluster. This topic uses a computer that runs the macOS operating system to describe the detailed procedure.

Prerequisites

You have completed the following operations:

- An Alibaba Cloud Elasticsearch cluster is created. For more information, see Create an Alibaba Cloud Elasticsearch cluster.

  > **Note** If you want to access the Elasticsearch cluster by using its internal endpoint, you must purchase an Elastic Compute Service (ECS) instance. This instance must reside in the same virtual private cloud (VPC), region, and zone as the Elasticsearch cluster.

- Metricbeat is downloaded.
  - Metricbeat installation package for macOS
  - Metricbeat installation package for Linux (x86)
  - Metricbeat installation package for Linux (x64)
  - Metricbeat installation package for Windows (x86)
  - Metric installation package for Windows (x64)

Context

Beats is a platform for single-purpose data shippers. After the data shippers are installed, they send data from thousands of machines and systems to Logstash or Elasticsearch.

Metricbeat is a lightweight data shipper that collects metrics from your systems and services. System metrics include CPU and memory metrics. Service metrics include Redis and NGINX metrics.

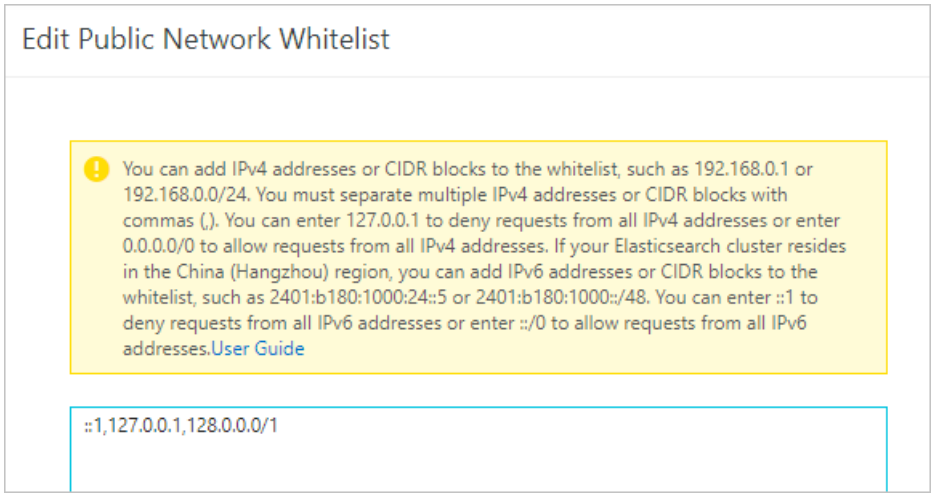
Procedure

1. Configure an Alibaba Cloud Elasticsearch cluster
2. Configure Metricbeat
3. View the related dashboard in the Kibana console

   > **Note** You can also use Metricbeat to collect metrics from a computer that runs the Linux or Windows operating system. Then, Metricbeat sends the metrics to Alibaba Cloud Elasticsearch.

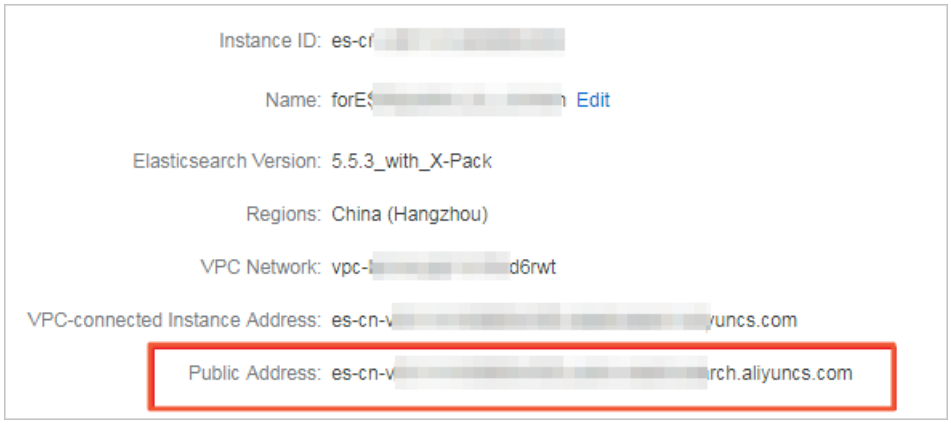
Configure an Alibaba Cloud Elasticsearch cluster

1.
2.
3. In the left-side navigation pane, click Elasticsearch Clusters. On the **Clusters** page, find your Elasticsearch cluster. Then, click its ID in the **Cluster ID/Name** column or **Manage** in the **Actions** column.
4. In the left-side navigation pane, click **Security**.
5. On the page that appears, turn on **Public Network Access** and click **Update** next to **Public Network Whitelist**. Then, enter the public IP address of your computer.

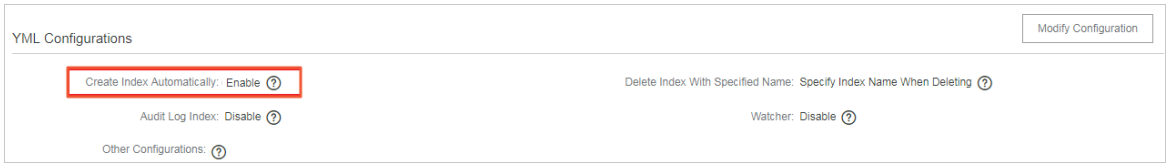


Notice If you are using a public network such as Wi-Fi, add the IP address of the jump server that controls outbound traffic of the public network to the whitelist. If you cannot obtain this IP address, we recommend that you add `0.0.0.0/1,128.0.0.0/1` to the whitelist. These two CIDR blocks are used in this topic. This configuration allows almost all public IP addresses to access the Elasticsearch cluster. We recommend that you evaluate the risks before you use this configuration.

6. In the left-side navigation pane, click **Basic Information**. On the page that appears, you can obtain the public endpoint of your Elasticsearch cluster for future use.



7. In the left-side navigation pane, click **Cluster Configuration**. On the page that appears, click **Modify Configuration** on the right side of **YML Configuration**. In the YML File Configuration pane, set **Auto Indexing** to **Enable**.



Warning This configuration takes effect only after your Elasticsearch cluster is restarted. To prevent impacts on your business, exercise caution when you change the settings of Auto Indexing.

8. Select **This operation will restart the cluster. Continue?** and click **OK**.

   You can view the restart progress in the Tasks dialog box. The configuration of the cluster takes effect after your Elasticsearch cluster is restarted.

Configure Metricbeat

1. Decompress the Metricbeat installation package you have downloaded and go to the Metricbeat folder.

```
MacBook-Pro:Desktop $ cd metricbeat-6.3.1-darwin-x86_64
MacBook-Pro:metricbeat-6.3.1-darwin-x86_64 $ ls
LICENSE.txt                 data                        logs                        metricbeat.yml
NOTICE.txt                  fields.yml                  metricbeat                  modules.d
README.md                   kibana                      metricbeat.reference.yml
deMacBook-Pro:metricbeat-6.3.1-darwin-x86_64 $
```

2. Open the *metricbeat.yml* file and edit the `Elasticsearch output` section in it. You must uncomment the involved content.

```
#================================ Outputs =====================================

# Configure what output to use when sending the data collected by the beat.

#-------------------------- Elasticsearch output ------------------------------
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["es-cn-                    .public.elasticsearch.aliyuncs.com:9200"]

  # Optional protocol and basic auth credentials.
  protocol: "http"
  username: "        :"
  password: "            "
```

Parameter	Description
`hosts`	The internal or public endpoint that is used to access the Elasticsearch cluster. The public endpoint is used to access the Elasticsearch cluster in this topic.
`protocol`	Set this parameter to `http`.
`username`	The default value of this parameter is `elastic`.
`password`	The password that is used to access the Elasticsearch cluster. This password is specified when you create the cluster.

3. Run the following command to start Metricbeat:

```
./metricbeat -e -c metricbeat.yml
```

```
deMacBook-Pro:metricbeat-6.3.1-darwin-x86_64 g$ ./metricbeat -e -c metricbeat.yml
```

After Metricbeat is started, it begins to send data to your Elasticsearch cluster.

View the related dashboard in the Kibana console

1. Log on to the Kibana console of your Elasticsearch cluster.

   For more information, see Log on to the Kibana console.

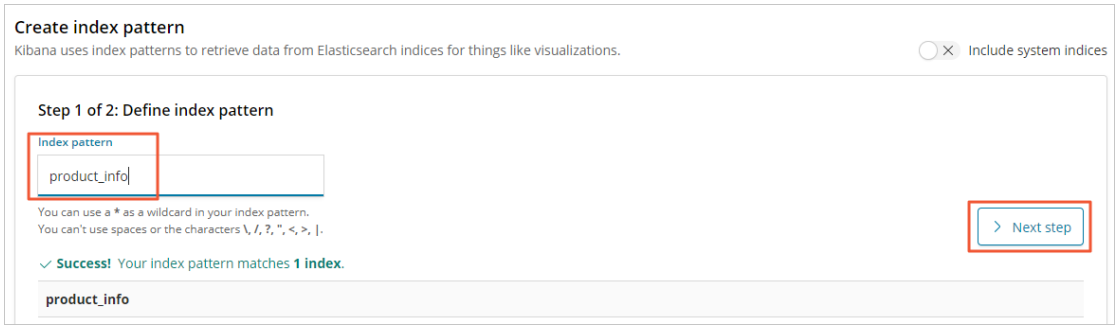
2. In the left-side navigation pane, click **Management** and follow these steps to create an index pattern:

   > **Notice** If you have created an index pattern, skip this step.

   i. In the **Kibana** section of the **Management** page, click **Index Patterns**.

   ii. In the **Create index pattern** section, enter an index pattern name (the name of the index that you want to query).

   iii. Click **Next step**.

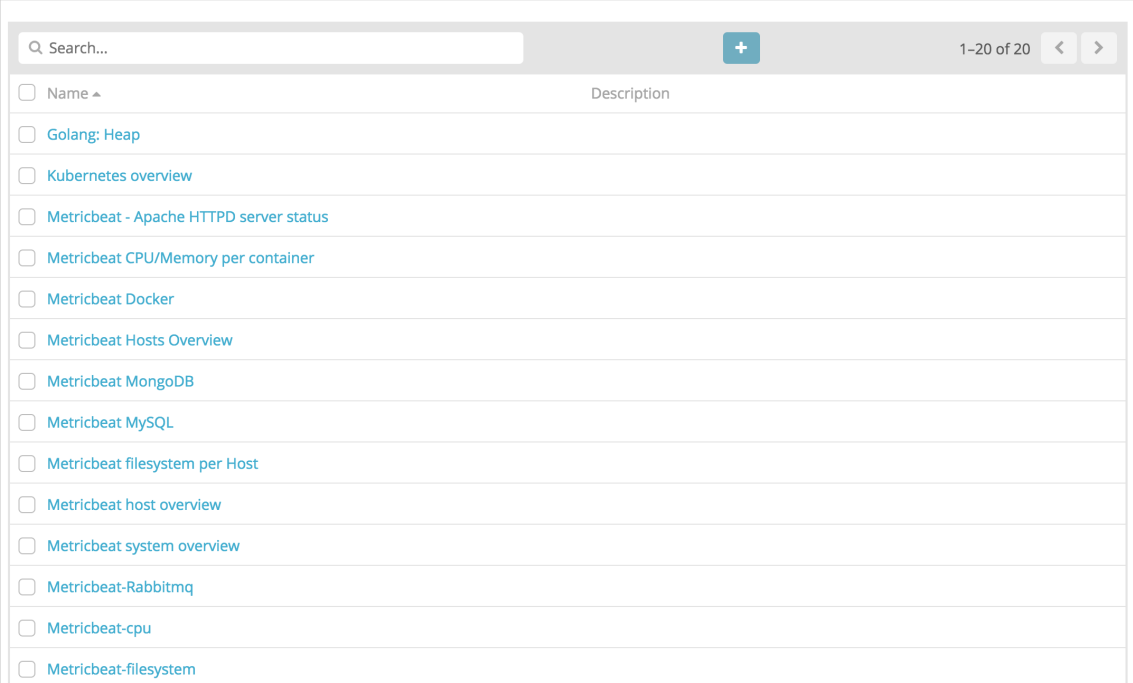
   

   iv. Click **Create index pattern**.

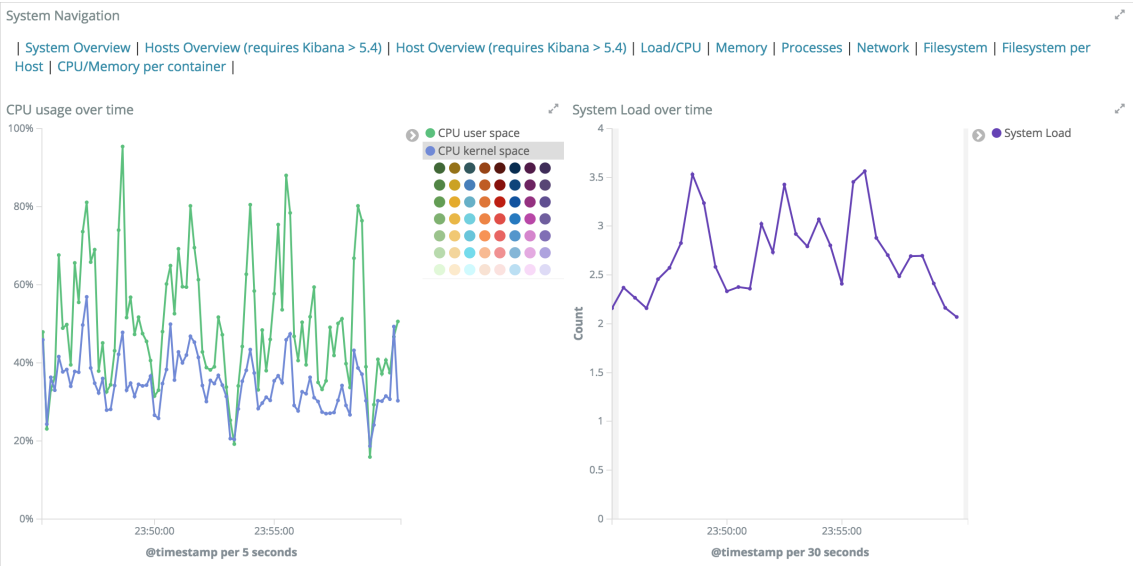
3. In the left-side navigation pane, click **Dashboard**.

4. On the **Dashboards** page, you can view the collected data.

   - The following figure shows relevant metrics.



- Click **Metricbeat-cpu** to view CPU metrics.



Note You can configure metrics to be refreshed at 5-second intervals. The system generates reports for the metrics collected at these intervals. You can also connect to WebHook to configure alerts.

References

Build a visualized operations and maintenance (O&M) system with Beats

8.4. Use SkyWalking to implement end-to-end monitoring on Alibaba Cloud Elasticsearch

Use SkyWalking to implement end-to-end monitoring on Alibaba Cloud Elasticsearch

SkyWalking is a distributed application performance monitoring (APM) tool and a distributed tracing system. This topic describes how to use SkyWalking to monitor an Alibaba Cloud Elasticsearch V7.4 cluster.

Context

SkyWalking has the following features:

- SkyWalking provides auto instrument agents so that you do not need to modify the application code.

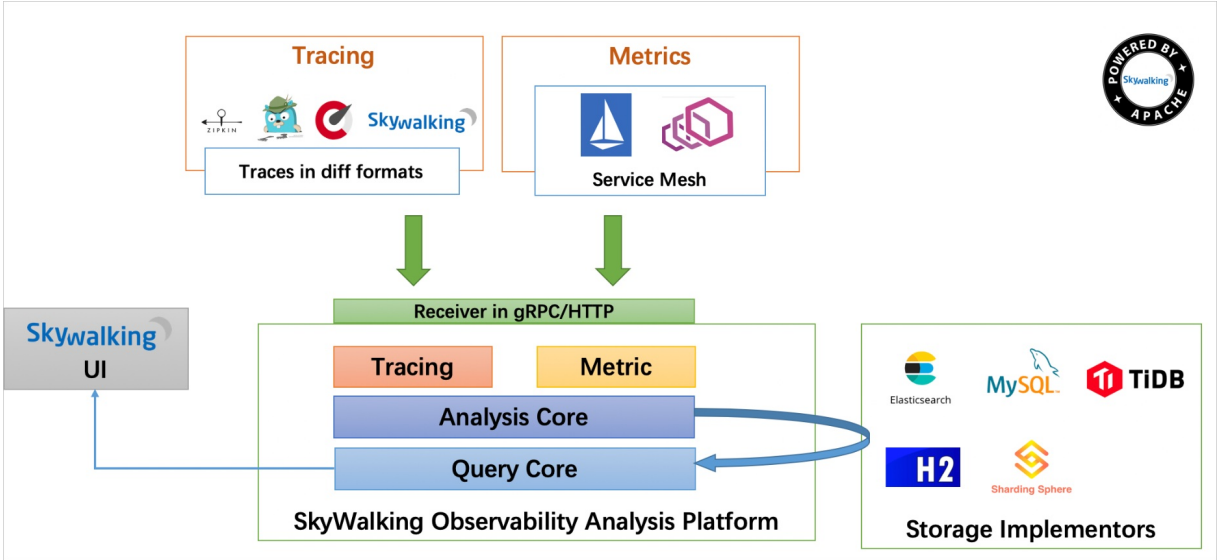
  > **Note** For the middleware and components supported by SkyWalking, see Apache SkyWalking documentation.

- SkyWalking provides manual instrument agents that support OpenTracing SDKs. Manual instrument agents can monitor components supported by OpenTracing API for Java.

  > **Note** For the components supported by OpenTracing API for Java, see OpenTracing Registry.

- Auto instrument agents and manual instrument agents can be used at the same time. Manual instrument agents can monitor components that are not supported by auto instrument agents, and even private components.
- SkyWalking is a Java-based backend program for analytics. It provides RESTful APIs and analytics capabilities for agents of other languages.
- SkyWalking provides high-performance streaming analytics.

The following figure shows the SkyWalking architecture.



SkyWalking is a platform for storing data analysis and measurement results. Data analysis and measurement results are submitted to SkyWalking Collector over HTTP or gRPC. SkyWalking Collecter analyzes and aggregates data and stores the data in Elasticsearch, H2, MySQL, or TiDB. You can view the analysis results on the SkyWalking UI. SkyWalking collects data in different formats from multiple sources, such as SkyWalking agents in multiple programming languages, Zipkin v1, Zipkin v2, Istio telemetry, and Envoy.

> **Note** In this topic, SkyWalking is integrated into Alibaba Cloud Elasticsearch V7.4. You can also use a Skywalking client to report data to a Java application.

Prerequisites

You have completed the following operations:

- An Alibaba Cloud Elasticsearch cluster is created. Alibaba Cloud Elasticsearch V7.4 is used in this example,.

  For more information, see Create an Alibaba Cloud Elasticsearch cluster.

- A Linux server is prepared. The server is installed with JDK 1.8 or later.

  We recommend that you use an Alibaba Cloud Elastic Compute Service (ECS) instance. For more information, see Step 1: Create an ECS instance.

  > **Note** For information about how to install JDK, see Install the JDK. If JDK is not correctly installed and you start SkyWalking to view logs, the error message "Java not found" or "java-xxx: No such file or directory" is reported.

- Ports 8080, 10800, 11800, and 12800 on the Linux server are not occupied.
- The firewall and Security-Enhanced Linux (SELinux) of the Linux server are disabled.

Procedure

1. Step 1: Download and install SkyWalking
2. Step 2: Configure SkyWalking to connect to Alibaba Cloud Elasticsearch

3. Step 3: Verify the results

Step 1: Download and install SkyWalking

1. Download the SkyWalking package to the Linux server.

   We recommend that you select the latest version 7.0.0. Alibaba Cloud Elasticsearch V7.4 is used in this example. Therefore, Binary Distribution for ElasticSearch 7 is selected. Run the following command to download the package:

   ```
   wget https://mirror.bit.edu.cn/apache/skywalking/7.0.0/apache-skywalking-apm-es7-7.0.0.tar.gz
   ```

2. Run the following command to decompress the package:

   ```
   tar -zxvf apache-skywalking-apm-es7-7.0.0.tar.gz
   ```

3. Run the following command to view the decompressed files:

   ```
   ll apache-skywalking-apm-bin-es7/
   ```

   The following result is returned:

   ```
   total 92
   drwxrwxr-x 8 1001 1002 143 Mar 18 23:50 agent
   drwxr-xr-x 2 root root   241 Apr 10 16:03 bin
   drwxr-xr-x 2 root root   221 Apr 10 16:03 config
   -rwxrwxr-x 1 1001 1002 29791 Mar 18 23:37 LICENSE
   drwxrwxr-x 3 1001 1002  4096 Apr 10 16:03 licenses
   -rwxrwxr-x 1 1001 1002 32838 Mar 18 23:37 NOTICE
   drwxrwxr-x 2 1001 1002 12288 Mar 19 00:00 oap-libs
   -rw-rw-r-- 1 1001 1002  1978 Mar 18 23:37 README.txt
   drwxr-xr-x 3 root root    30 Apr 10 16:03 tools
   drwxr-xr-x 2 root root    53 Apr 10 16:03 webapp
   ```

Step 2: Configure SkyWalking to connect to Alibaba Cloud Elasticsearch

1. Run the following commands to open the *application.yml* file in the config directory:

   ```
   cd apache-skywalking-apm-bin-es7/config/
   vi application.yml
   ```

2. Locate `storage`, change H2 to elasticsearch7, and configure the file as follows:

```
storage:
  selector: ${SW_STORAGE:elasticsearch7}
  elasticsearch7:
    nameSpace: ${SW_NAMESPACE:"skywalking-index"}
    clusterNodes: ${SW_STORAGE_ES_CLUSTER_NODES:es-cn-4591kzdzk000i****.public.elasticsearch.aliyuncs.com:9200}
    protocol: ${SW_STORAGE_ES_HTTP_PROTOCOL:"http"}
   # trustStorePath: ${SW_SW_STORAGE_ES_SSL_JKS_PATH:"../es_keystore.jks"}
   # trustStorePass: ${SW_SW_STORAGE_ES_SSL_JKS_PASS:""}
    enablePackedDownsampling: ${SW_STORAGE_ENABLE_PACKED_DOWNSAMPLING:true} # Hour and Day metrics will be merged into minute index.
    dayStep: ${SW_STORAGE_DAY_STEP:1} # Represent the number of days in the one minute/hour/day index.
    user: ${SW_ES_USER:"elastic"}
    password: ${SW_ES_PASSWORD:"es_password"}
```

Note SkyWalking stores data in H2 by default. However, H2 does not support persistent data storage, and you must change H2 to Elasticsearch.

Parameter	Description
`selector`	The storage selector. For this example, set the value to elasticsearch7.
`nameSpace`	The namespace. The value of this parameter is used as the prefix for all indexes of the Elasticsearch cluster.
`clusterNodes`	The endpoint of the Elasticsearch cluster. Alibaba Cloud Elasticsearch is not in the same Virtual Private Cloud (VPC) as SkyWalking. You must use the public endpoint to access Elasticsearch. For more information, see View the basic information of a cluster.
`user`	The username of your Elasticsearch cluster. The default username is elastic.
`password`	The password of your Elasticsearch cluster. The password is specified when you create the Elasticsearch cluster.

Notice Specify only the username and password. Comment out trustStorePath and trustStorePass. Otherwise, the error message "NoSuchFileException: /es_keystore.jks" is reported.

3. (Optional)Modify the IP address or port for listening.

   By default, SkyWalking communicates with Elasticsearch over port 12800 for RESTful API operations and port 11800 for gRPC API operations. The IP address or port can be modified in the core part of the *application.yml* file. In this example, the default values are used.

```
core:
  selector: ${SW_CORE:default}
  default:
    # Mixed: Receive agent data, Level 1 aggregate, Level 2 aggregate
    # Receiver: Receive agent data, Level 1 aggregate
    # Aggregator: Level 2 aggregate
    role: ${SW_CORE_ROLE:Mixed} # Mixed/Receiver/Aggregator
    restHost: ${SW_CORE_REST_HOST:0.0.0.0}
    restHost: ${SW_CORE_REST_HOST:0.0.0.0}
    restContextPath: ${SW_CORE_REST_CONTEXT_PATH:/}
    gRPCHost: ${SW_CORE_GRPC_HOST:0.0.0.0}
    gRPCPort: ${SW_CORE_GRPC_PORT:11800}
```

4. (Optional)In the webapp directory, modify the configurations in the webapp.yml file.

   Default configurations are used in this example. You can modify the configurations as required.

```
server:
  port: 8080
collector:
  path: /graphql
  ribbon:
    ReadTimeout: 10000
    # Point to all backend's restHost:restPort, split by ,
    listOfServers: 127.0.0.1:12800
```

Step 3: Verify the results

1. Run the following commands to start SkyWalking on the Linux server:

```
cd ../bin
./startup.sh
```

> **Notice**
>
> - Make sure that your Elasticsearch cluster is started before you start SkyWalking.
> - SkyWalking Collector and SkyWalking UI are both started by running the `./startup.sh` command.

   If SkyWalking is started, the following result is returned:

```
SkyWalking OAP started successfully!
SkyWalking Web Application started successfully!
```

2. Enter http://<IP address of the Linux server>:8080/ in the address bar of your browser.

Note When you use SkyWalking to connect to Alibaba Cloud Elasticsearch for the first time, the startup is slow. This is because SkyWalking needs to create a large number of indexes in Elasticsearch. Before the creation is complete, the accessed page is blank. You can view logs that are stored in `<SkyWalking installation path>logs/skywalking-oap-server.log` to check whether the creation is complete.

3. Log on to the Kibana console of your Elasticsearch cluster. For more information, see Log on to the Kibana console. Run the `GET _cat/indices? v` command to view the index data.
   In the returned results, a large number of indexes starting with `skywalking-index` exist.

```
green  open   skywalking-index_all_percentile-20200408
green  open   skywalking-index_endpoint_inventory
green  open   skywalking-index_service_apdex-20200408
green  open   skywalking-index_database_access_resp_time_month-202004
green  open   skywalking-index_service_relation_client_cpm-20200408
green  open   skywalking-index_database_access_sla_month-202004
green  open   skywalking-index_service_relation_server_call_sla-20200408
green  open   skywalking-index_endpoint_sla-20200408
green  open   skywalking-index_instance_jvm_memory_noheap-20200408
green  open   skywalking-index_service_relation_server_percentile-20200408
green  open   skywalking-index_service_instance_relation_server_resp_time-20200408
green  open   skywalking-index_profile_task_segment_snapshot-20200408
green  open   skywalking-index_endpoint_relation_cpm-20200408
green  open   skywalking-index_instance_jvm_old_gc_time-20200408
green  open   skywalking-index_service_sla-20200408
green  open   skywalking-index_top_n_database_statement-20200408
green  open   skywalking-index_service_relation_client_call_sla-20200408
green  open   skywalking-index_endpoint_percentile_month-202004
green  open   apm-agent-configuration
```

8.5. Use Uptime to monitor Alibaba Cloud Elasticsearch clusters in real time

Heartbeat shippers can detect the statuses of network endpoints based on the HTTP or HTTPS, TCP, and ICMP services on a regular basis. They can also send the collected detection data to the Uptime application in Kibana. Then, the Uptime application monitors the availability and response time of applications and services in real time and reports errors before your business is affected. This topic describes how to use Uptime to monitor an Alibaba Cloud Elasticsearch cluster in real time.

Context

Uptime must be used with the following services:

- Heartbeat

- Heartbeat
- Elasticsearch
- Kibana

> **Note** You can also use the Alerting and Actions feature of Kibana V7.7 to configure monitoring and alerting. For more information, see Alerting and Actions.

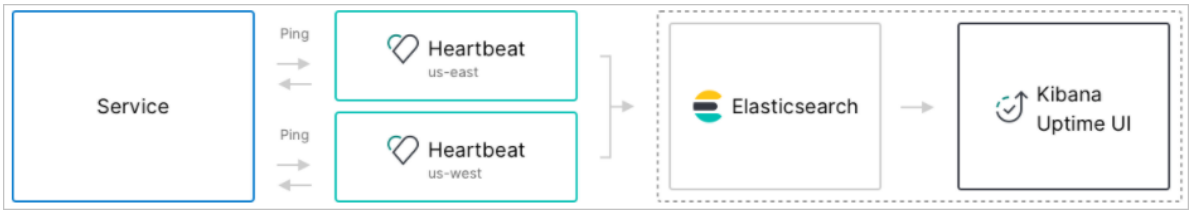
Deployment architecture

- Deployment of a single Heartbeat shipper

  A single Heartbeat shipper is deployed at a single position to monitor a single service. The Heartbeat shipper sends monitoring data to Elasticsearch. You can view the heartbeat data of the service on the Uptime page of Kibana and determine the service status based on the heartbeat data.

- Deployment of multiple Heartbeat shippers

  Two Heartbeat shippers are deployed at different positions to monitor the same service. The two Heartbeat shippers send monitoring data to Elasticsearch. You can view the heartbeat data of the service on the Uptime page of Kibana and determine the service status based on the heartbeat data. If the Heartbeat shipper at one position becomes faulty, the Heartbeat shipper at the other position can help locate the fault.



For more information about the deployment architecture, see Deployment Architecture.

Preparations

1. Create an Alibaba Cloud Elasticsearch cluster and enable the Auto Indexing feature for the cluster.

   For more information, see Create an Alibaba Cloud Elasticsearch cluster and Configure the YML file.

2. Create an Elastic Compute Service (ECS) instance, which is used to deploy a Heartbeat shipper. The ECS instance must reside in the same virtual private cloud (VPC) as the Elasticsearch cluster.

   For more information, see Create an instance by using the wizard.

   > **Notice** Beats supports only the following operating systems: Alibaba Cloud Linux, Red Hat Enterprise Linux (RHEL), and Community Enterprise Operating System (CentOS). Therefore, you must select one of the preceding operating systems when you create the ECS instance.

3. Install Cloud Assistant and Docker on the ECS instance.

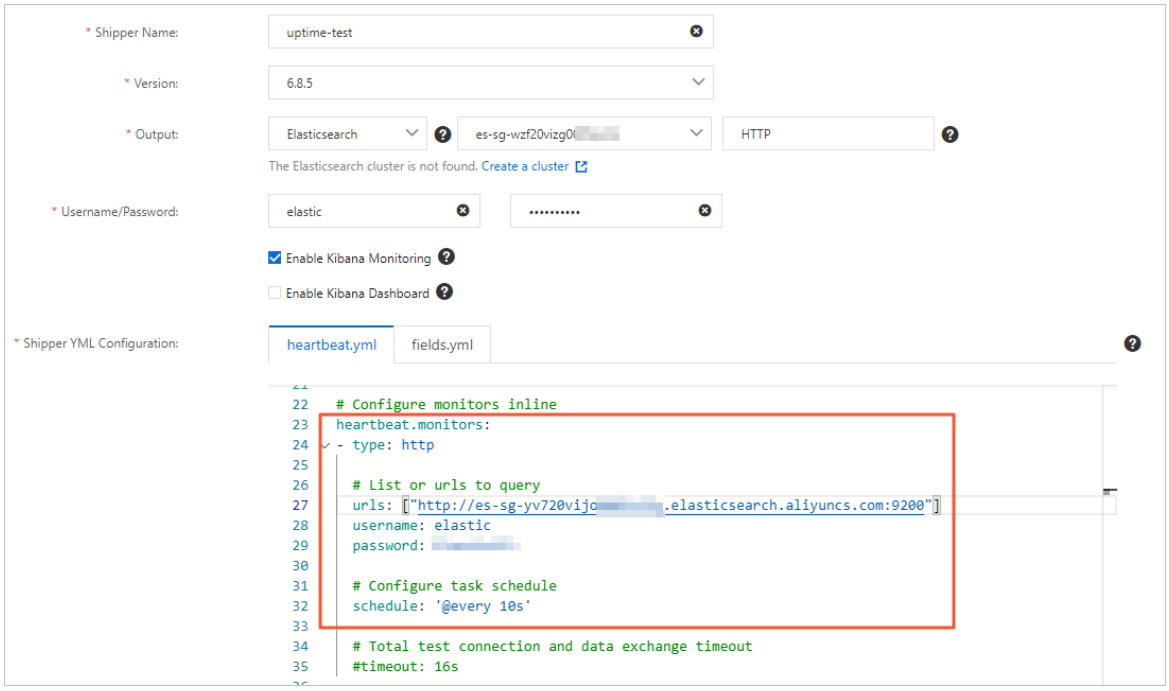
   For more information, see Install the Cloud Assistant client and Deploy and use Docker on Alibaba

Cloud Linux 2 instances.

Create a Heartbeat shipper

1.
2. On the Beats Data Shippers page, click **Heartbeat** in the **Create Shipper** section.
3. Install and configure a Heartbeat shipper.

   For more information, see Collect the logs of an ECS instance and Prepare the YML configuration files for a shipper.

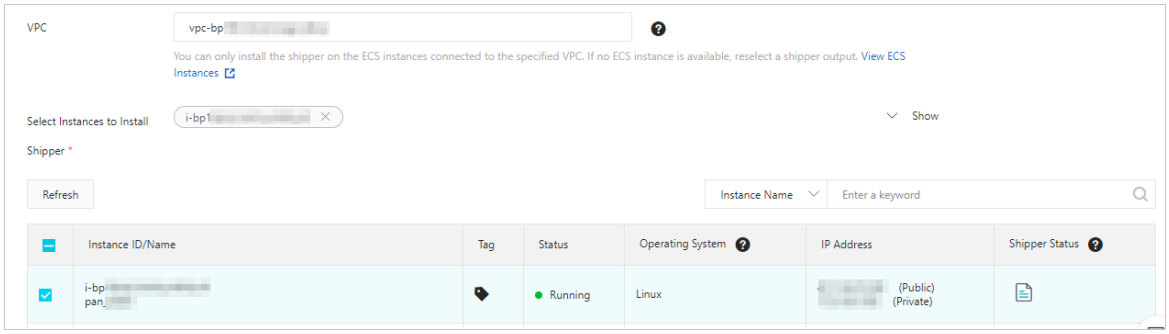


Configurations for heartbeat.monitors

Parameter	Description
type	In this topic, http is used. **Note** The Heartbeat shipper can monitor HTTP or HTTPS, TCP, and ICMP services. If you use an HTTP or HTTPS monitor, response code, request bodies, and request headers can be monitored. If you use a TCP monitor, port numbers and strings can be monitored.
urls	The URLs that you want to check. You can specify multiple HTTP services. In this topic, an Alibaba Cloud Elasticsearch cluster is checked. This parameter is set to the internal endpoint of the Elasticsearch cluster that you want to check.
schedule	The interval at which checks are performed. The value @every 10s indicates that the check is performed every 10 seconds.

4. Click **Next**.
5. In the **Install Shipper** step, select the ECS instance on which you want to install the Heartbeat

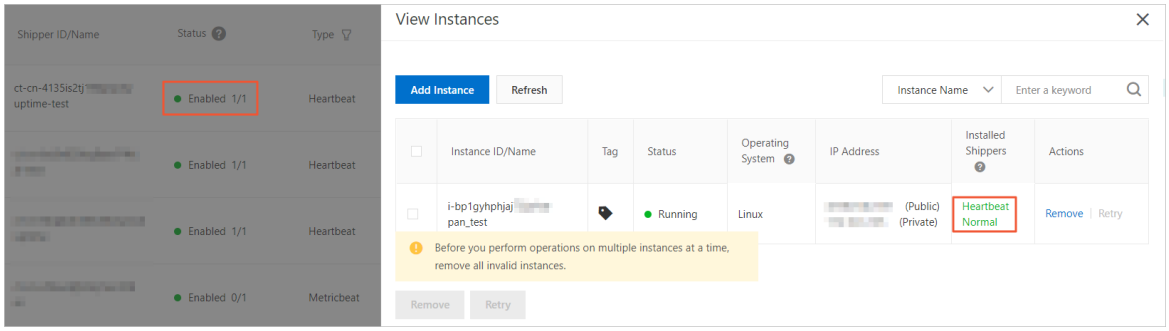
shipper.



6. Enable the Heartbeat shipper and check whether the Heartbeat shipper is installed.

   For more information, see Collect the logs of an ECS instance.

   If the state of the Heartbeat shipper is **Enabled**, and the installation state of the Heartbeat shipper is **Heartbeat Normal**, the Heartbeat shipper is installed.

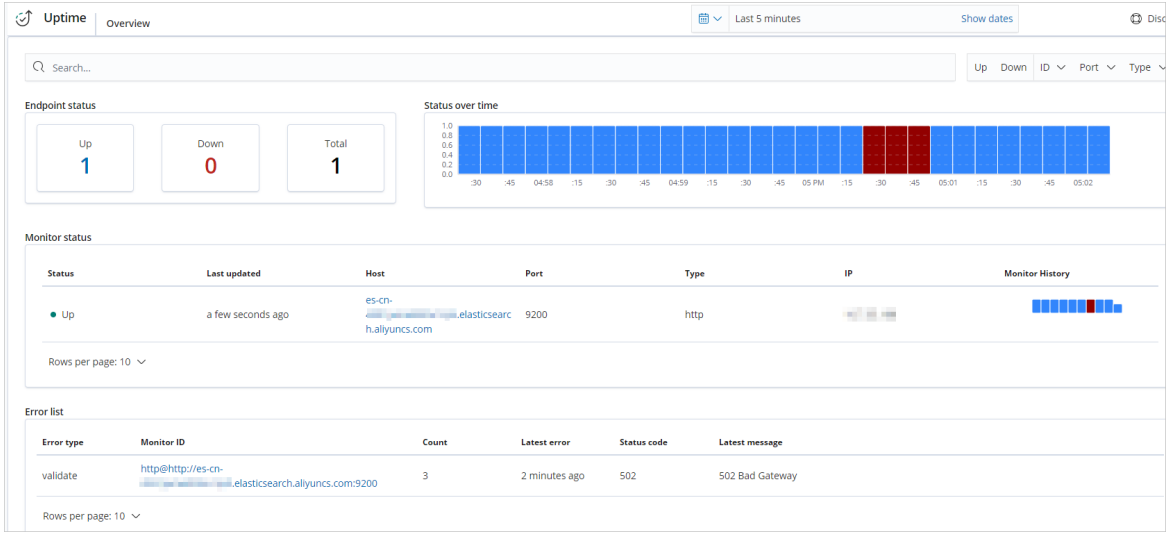


View the monitoring information on the Uptime page

1. Log on to the Kibana console of your Elasticsearch cluster.

   The Kibana console is the one that corresponds to the Elasticsearch cluster that you specified for **Output** when you create the Heartbeat shipper. For more information, see Log on to the Kibana console.

2. In the left-side navigation pane, click **Uptime**. On the Uptime page, you can view the monitoring information.



   - Color red: indicates that the Elasticsearch cluster is in an abnormal state. Check the

communication status of the Heartbeat shipper or the status of the Elasticsearch cluster.

- Color blue: indicates that the Elasticsearch cluster is in a normal state.

9.Cluster management

9.1. Overview of cluster management

You can use various methods to manage your Alibaba Cloud Elasticsearch clusters. This topic provides an overview of best practices for cluster management to meet your business requirements in various scenarios.

Best practice	References	Description
Hot and cold data separation and lifecycle management	Use ILM to manage Heartbeat indexes	Time series data increase over time. You can use the index lifecycle management (ILM) feature to periodically roll over the data to new indexes. This ensures high query efficiency and reduces query costs. As indexes age and fewer queries are required, you can migrate the indexes to a less expensive disk and reduce the numbers of primary and replica shards.
	Use ILM to separate hot data from cold data	The cluster that uses the hot-warm architecture contains hot nodes and warm nodes. This architecture improves the performance and stability of your Elasticsearch cluster. When you use an Alibaba Cloud Elasticsearch cluster, you can use the ILM feature to separate hot data from cold data in the cluster. This improves the read and write performance of the cluster, automates the maintenance of hot and cold data, and reduces your production costs.
Application of X-Pack advanced features	Use the CCR feature to migrate data	You can use the cross-cluster replication (CCR) feature to migrate index data between a local Alibaba Cloud Elasticsearch cluster and a remote Alibaba Cloud Elasticsearch cluster. This feature helps implement high availability and disaster recovery for your Alibaba Cloud Elasticsearch cluster. You can also use the feature for cross-region data access from a nearby cluster.
	Use X-Pack to configure LDAP authentication	When you use an Alibaba Cloud Elasticsearch cluster, you can configure Lightweight Directory Access Protocol (LDAP) authentication for the cluster to allow LDAP users with the required roles to access the cluster.

Best practice	References	Description
	Use the RBAC mechanism provided by Elasticsearch X-Pack to implement access control	If you want to grant access permissions on items such as clusters, indexes, and fields, you can use the role-based access control (RBAC) mechanism that is provided by the X-Pack plug-in of Elasticsearch. This mechanism allows you to grant permissions to custom roles and assign the roles to users to implement access control. Elasticsearch provides a variety of built-in roles. You can create custom roles based on the built-in roles to meet your business requirements.
	Configure AD user authentication	Elasticsearch allows you to configure Active Directory (AD) user authentication for your Elasticsearch cluster. This way, users in an AD domain that are assigned Elasticsearch roles can be used to access the cluster.
Integrated monitoring	Use Elastic Stack to implement integrated monitoring for containers in Kubernetes	Elastic Stack provides the integrated monitoring feature. This feature allows you to use Kibana to analyze and display the logs, metrics, and application performance monitoring (APM) data of a Container Service for Kubernetes (ACK) cluster in a centralized manner. If you deploy your applications in the pods of an ACK cluster, you can view the logs generated by the pods, event metrics of the hosts and network, and APM data in the Kibana console. This facilitates troubleshooting.
Data management and visualization	Use Terraform to manage Alibaba Cloud Elasticsearch clusters	Terraform allows you to use code to allocate resources such as physical machines. You can use Terraform to write a configuration file to purchase a cloud server or apply for resources, such as the resources of the Alibaba Cloud Elasticsearch and Object Storage Service (OSS) services. You can use Terraform to manage your Alibaba Cloud Elasticsearch clusters. For example, you can use Terraform to create, update, view, or delete a cluster.
	Use Curator	Curator is an index management tool provided by open source Elasticsearch. This tool allows you to create, delete, and disable indexes. It also allows you to merge index segments.
	Use the rollup mechanism to summarize traffic data	Time series data increases over time. If you want to store large volumes of data in your Alibaba Cloud Elasticsearch cluster, the storage costs will linearly increase. You can use the rollup mechanism of Elasticsearch to store data at a fraction of the cost.

Best practice	References	Description
	Use Cerebro to access an Elasticsearch cluster	In addition to Kibana, curl commands, and clients, you can use third-party plug-ins or tools such as Elasticsearch-Head and Cerebro to access an Alibaba Cloud Elasticsearch cluster.
Notification of alerts for clusters	Configure a DingTalk chatbot to receive alert notifications from X-Pack Watcher	X-Pack Watcher is a monitoring and alerting service developed for Elasticsearch. If you configure X-Pack Watcher for your cluster, X-Pack Watcher can trigger actions when specific conditions are met. For example, if the logs index contains errors, X-Pack Watcher triggers the system to send alert notifications by using emails, DingTalk messages, or DingTalk chatbots. X-Pack Watcher is an Elasticsearch-based monitoring and alerting service.

9.2. Hot and cold data separation and lifecycle management

9.2.1. Use ILM to manage Heartbeat indexes

Use Elasticsearch ILM

Time series data increase over time. You can use the index lifecycle management (ILM) feature to periodically roll over the data to new indexes. This ensures high query efficiency and reduces query costs. As indexes age and fewer queries are required, you can migrate the indexes to a less expensive disk and reduce the numbers of primary and replica shards. This topic describes how to use ILM to manage Heartbeat indexes.

Context

In this topic, the following test scenario is used:

A large number of time series indexes whose names start with heartbeat- exist in your Elasticsearch cluster, and the size of a single index is about 4 MB each day. The number of shards increases with the data volume. This may cause cluster overload. In this case, you must configure different rollover policies for indexes in the following four phases: hot, warm, cold, and delete. In the hot phase, data in historical monitoring indexes whose names start with heartbeat- is rolled over to new indexes. In the warm phase, indexes are shrunk, and segments in each index are merged. In the cold phase, data is migrated from hot nodes to warm nodes. In the delete phase, data is deleted on a regular basis.

Precautions

- An ILM policy can be attached to an index only after an index template and an alias are configured for the index.
- If you modify an ILM policy during a rollover, the new policy takes effect from the next rollover.

Procedure

1. Step 1: Create an Elasticsearch cluster that uses the hot-warm architecture

   Create an Elasticsearch cluster that uses the hot-warm architecture, enable the Auto Indexing

feature for the cluster, and configure a public IP address whitelist for the cluster.

2. Step 2: Enable and configure the ILM feature in the heartbeat.yml file

   In the *heartbeat.yml* file, enable and configure the ILM feature for the cluster. After the configuration is complete, the system generates a Heartbeat index template for the cluster.

3. Step 3: Create an ILM policy

   Call the ILM policy operation to create an ILM policy. This policy defines the conditions to roll over data and archive indexes.

4. Step 4: Attach the ILM policy to an index template

   Attach the ILM policy to the Heartbeat index template.

5. Step 5: Attach the ILM policy to an index

   Attach the ILM policy to the first index that is created by using the Heartbeat index template. This way, the policy can apply to all indexes that are created by using this template.

6. Step 6: View indexes in different phases

   View the indexes that are archived in the hot, warm, cold, and delete phases.

Step 1: Create an Elasticsearch cluster that uses the hot-warm architecture

1. Create an Elasticsearch cluster that uses the hot-warm architecture and view the hot or warm attribute of nodes in the cluster.

   The cluster that uses the hot-warm architecture contains hot nodes and warm nodes. This architecture improves the performance and stability of your Elasticsearch cluster. The following table lists the differences between hot nodes and warm nodes.

Node type	Type of data stored	Read and write performance	Specifications	Disk
Hot node	Recent data, such as log data over the last two days.	High	High, such as 32 vCPUs and 64 GiB of memory	We recommend that you use a standard SSD. You can specify the storage space based on the volume of data.
Warm node	Historical data, such as log data before the last two days.	Low	Low, such as 8 vCPUs and 32 GiB of memory	We recommend that you use an ultra disk. You can specify the storage space based on the volume of data.

i. When you purchase an Elasticsearch cluster, you can purchase warm nodes to create an Elasticsearch cluster that uses the hot-warm architecture.

After you create a cluster that contains warm nodes, the system adds the -Enode.attr.box_type parameter to the startup parameters of nodes.

- Hot node: -Enode.attr.box_type=hot
- Warm node: -Enode.attr.box_type=warm

> **Note**
>
> - Data nodes become hot nodes only after you purchase warm nodes.
> - In this topic, an Alibaba Cloud Elasticsearch V6.7.0 cluster is used. All operations described and figures provided in this topic are suitable only for clusters of this version. If you use a cluster of another version, operations required in the Elasticsearch console prevail.

ii. Log on to the **Kibana console** of the Elasticsearch cluster.

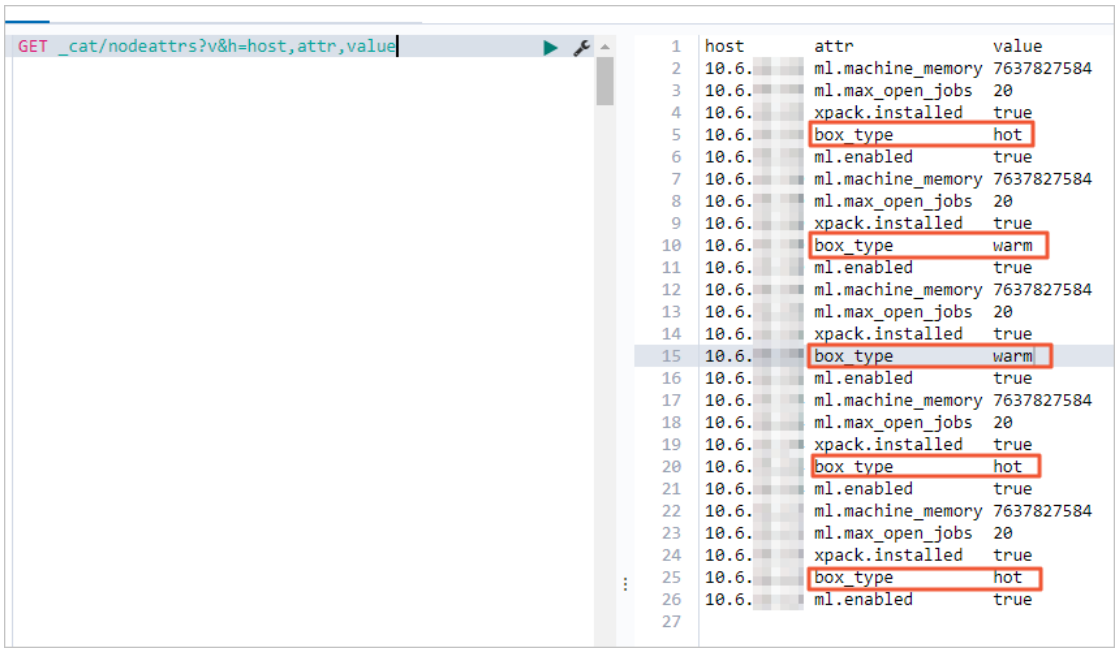
For more information about how to log on to the Kibana console, see Log on to the Kibana console.

iii. In the left-side navigation pane, click **Dev Tools**.

iv. On the **Console** tab of the page that appears, run the following command to view the attributes of nodes:

```
GET _cat/nodeattrs?v&h=host,attr,value
```

If the command is successfully run, the result shown in the following figure is returned. This figure shows that the Elasticsearch cluster contains three hot nodes and two warm nodes to support the hot-warm architecture.



2. Enable the Auto Indexing feature for the Elasticsearch cluster.

For more information, see Configure the YML file.

3. Configure a public IP address whitelist for the Elasticsearch cluster and add the IP address of the server on which Heartbeat is installed to the whitelist.

   For more information, see Configure a public or private IP address whitelist for an Elasticsearch cluster.

Step 2: Enable and configure the ILM feature in the heartbeat.yml file

To manage Heartbeat indexes by using the ILM feature of Elasticsearch, you can configure the feature in the heartbeat.yml file. For more information, see Set up index lifecycle management.

1. Download the Heartbeat installation package and decompress it.
2. Specify the heartbeat.monitors, setup.template.settings, setup.kibana, and output.elasticsearch configurations in the heartbeat.yml file.

   The following configurations are used in this example:

```
heartbeat.monitors:
- type: icmp
  schedule: '*/5 * * * * * *'
  hosts: ["47.111.xx.xx"]
setup.template.settings:
  index.number_of_shards: 3
  index.codec: best_compression
  index.routing.allocation.require.box_type: "hot"
setup.kibana:
  # Kibana Host
  # Scheme and port can be left out and will be set to the default (http and 5601)
  # In case you specify and additional path, the scheme is required: http://localhost:5
601/path
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
  host: "https://es-cn-4591jumei00xxxxxx.kibana.elasticsearch.aliyuncs.com:5601"
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["es-cn-4591jumei00xxxxxx.elasticsearch.aliyuncs.com:9200"]
  ilm.enabled: true
  setup.template.overwrite: true
  ilm.rollover_alias: "heartbeat"
  ilm.pattern: "{now/d}-000001"
  # Enabled ilm (beta) to use index lifecycle management instead daily indices.
  #ilm.enabled: false
  # Optional protocol and basic auth credentials.
  #protocol: "https"
  username: "elastic"
  password: "<your_password>"
```

   The following table describes some parameters in the preceding configurations. For more information about other parameters, see open source Heartbeat configuration documentation.

Parameter	Description
index.number_of_shards	The number of primary shards. Default value: 1.

Parameter	Description
index.routing.allocation.require.box_type	Specifies whether to write data to hot nodes.
host	The public IP address that is used to access the Kibana service. You can obtain the IP address on the Kibana Configuration page.
hosts	The internal or public endpoint that is used to access the Elasticsearch cluster. You can obtain the endpoint on the Basic Information page of the cluster. For more information, see View the basic information of a cluster. **Note** If you set the hosts parameter to the public endpoint of the cluster, you must configure a public IP address whitelist for the cluster. For more information, see Configure a public or private IP address whitelist for an Elasticsearch cluster. If you set the hosts parameter to the internal endpoint of the cluster, you must make sure that the cluster resides in the same virtual private cloud (VPC) as the server on which Heartbeat is installed.
ilm.enabled	Specifies whether to enable the ILM feature. If this parameter is set to true, the feature is enabled.
setup.template.overwrite	Specifies whether to overwrite the original index template. If you have loaded an index template of a specific version to Elasticsearch, you must set this parameter to true to overwrite the original index template with the loaded template.
ilm.rollover_alias	Specifies the alias of the index that is generated during a rollover. Default value: heartbeat-\{beat.version\}.
ilm.pattern	The index pattern that is generated during a rollover. date math is supported. Default value: {now/d}-000001. If a rollover condition is met, the system increments the last digit in the index name by one to generate a new index name. For example, an index generated after the first rollover is named heartbeat-2020.04.29-000001. If another rollover condition is met, Elasticsearch creates an index named heartbeat-2020.04.29-000002.
username	
password	

Notice If you change the setting of ilm.rollover_alias or ilm.pattern after an index template is loaded, you must set setup.template.overwrite to true to overwrite the original index template with the loaded index template.

3. Start the Heartbeat service.

```
sudo ./heartbeat -e
```

Step 3: Create an ILM policy

Elasticsearch allows you to use API calls or the Kibana console to create an ILM policy. This step describes how to call the ILM policy operation to create an ILM policy.

> **Note** Heartbeat allows you to run the `./heartbeat setup --ilm-policy` command to load the default policy and write it to Elasticsearch. You can run the `./heartbeat export ilm-policy` command to export the default policy to stdout. Then, you can modify the default policy to manually create an ILM policy.

Run the following command in the Kibana console of the Elasticsearch cluster to create an ILM policy:

```
PUT /_ilm/policy/hearbeat-policy
{
  "policy": {
    "phases": {
      "hot": {
        "actions": {
          "rollover": {
            "max_size": "5mb",
            "max_age": "1d",
            "max_docs": 100
          }
        }
      },
      "warm": {
        "min_age": "60s",
        "actions": {
          "forcemerge": {
                "max_num_segments":1
              },
          "shrink": {
                "number_of_shards":1
              }
        }
      },
      "cold": {
        "min_age": "3m",
        "actions": {
          "allocate": {
            "require": {
              "box_type": "warm"
            }
          }
        }
      },
      "delete": {
        "min_age": "1h",
        "actions": {
          "delete": {}
        }
      }
    }
  }
}
```

The following table describes the configurations in the preceding ILM policy.

Parameter	Description

Parameter	Description
hot	A rollover is triggered if an index to which the ILM policy is attached meets one of the following conditions: The volume of data in the index reaches 5 MB, the index has been used for more than one day, and the number of documents in the index exceeds 100. During the rollover, the system creates an index and enables the ILM policy for the new index. The original index enters the warm phase 60 seconds after the rollover. **Notice** If the value of max_docs, max_size, or max_age is reached during a rollover, Elasticsearch archives the index.
warm	After the index enters the warm phase, the system shrinks it down to a new index that has only one primary shard and merges segments in the index into one segment. The index enters the cold phase 3 minutes after the rollover starts.
cold	After the index enters the cold phase, the system migrates the index from hot nodes to warm nodes. The index enters the delete phase 1 hour later after the rollover starts.
delete	After the index enters the delete phase, it is deleted.

Note

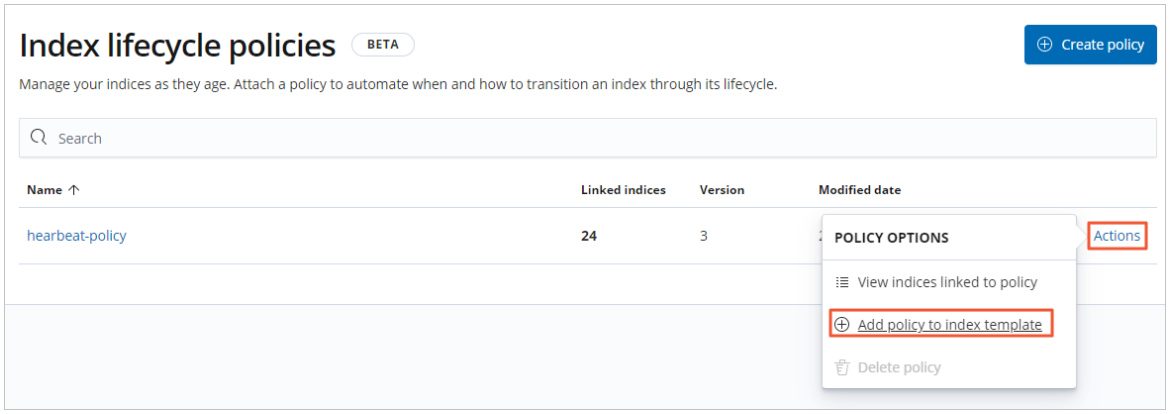
- After an ILM policy is created, you cannot change the policy name.
- In this step, you can specify the max_age parameter in the minimum unit of seconds. If you use the Kibana console to create an ILM policy, you can specify this parameter only in the minimum unit of hours.

Step 4: Attach the ILM policy to an index template

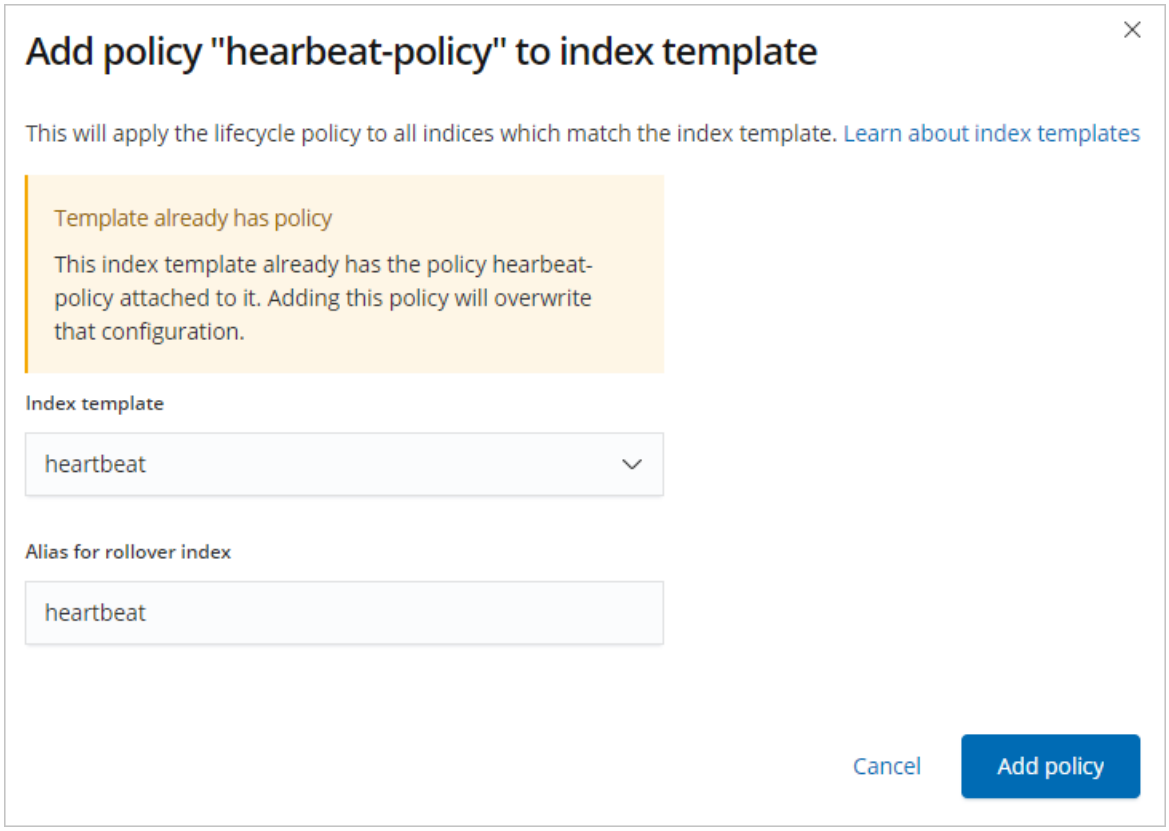
After you start Heartbeat, the system creates a Heartbeat index template in your Elasticsearch cluster. You must attach the ILM policy created in Step 3: Create an ILM policy to this index template.

1. Log on to the Kibana console of the Elasticsearch cluster.

   For more information, see Log on to the Kibana console.
2. In the left-side navigation pane, click **Management**.
3. In the **Elasticsearch** section, click **Index Lifecycle Policies**.
4. In the **Index lifecycle policies** section, find the ILM policy you created, and choose **Actions > Add policy to index template**.



5. In the dialog box that appears, select an index template from the **Index template** drop-down list and enter an alias for indexes in the **Alias for rollover index** field.

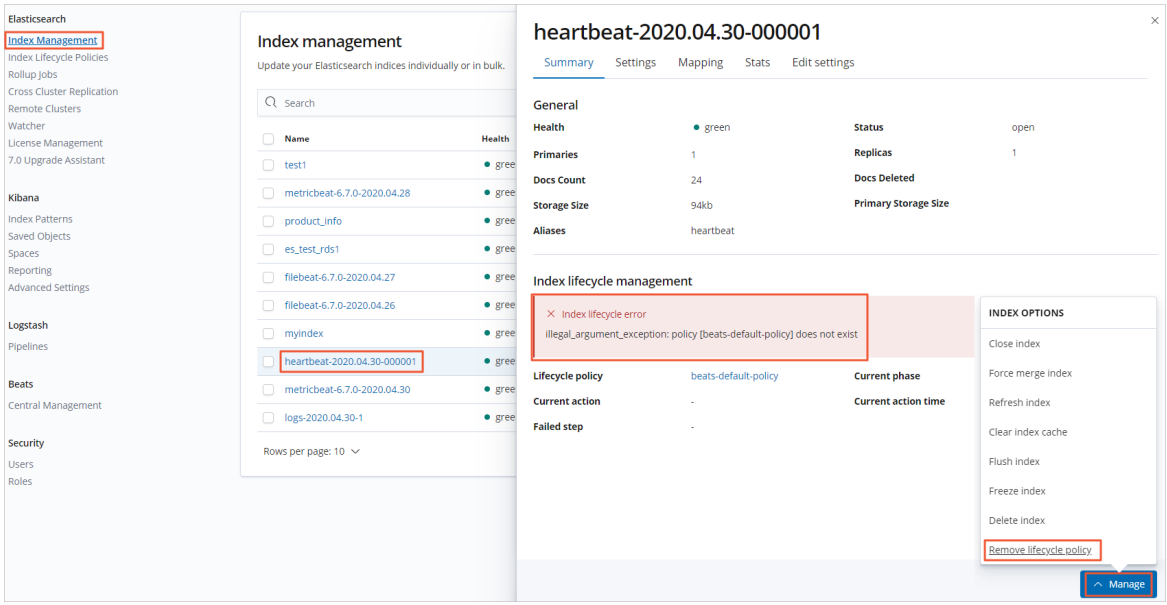


6. Click **Add policy**.

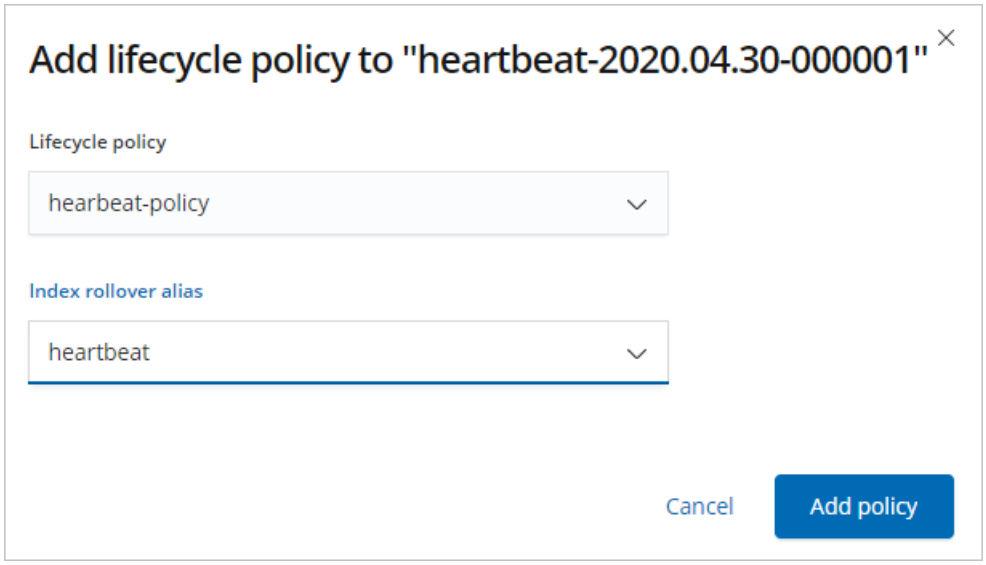
Step 5: Attach the ILM policy to an index

After you start Heartbeat, the system creates Heartbeat indexes in your Elasticsearch cluster. You must attach the ILM policy that is attached to the index template you created to the first index created by using the template. For more information, see Step 4: Attach the ILM policy to an index template.

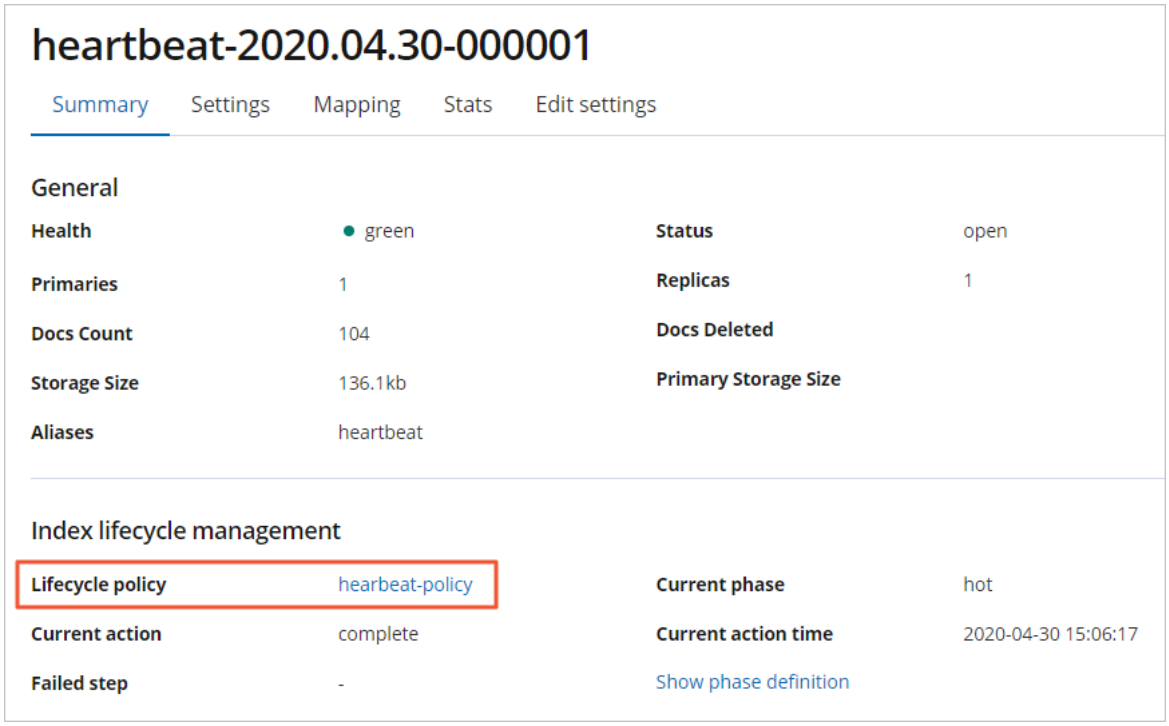
1. In the **Elasticsearch** section of the **Management** page, click **Index Management**.
2. In the **Index management** section, find the desired index and click its name.
3. On the **Summary** tab of the pane that appears, choose **Manage > Remove lifecycle policy** to remove the default policy of Heartbeat.



4. In the dialog box that appears, click **Remove policy**.
5. Choose **Manage > Add lifecycle policy** again.
6. In the dialog box that appears, select the ILM policy you created in Step 3: Create an ILM policy from the **Lifecycle policy** drop-down list and set **Index rollover alias** to the alias that you specify in Step 4: Attach the ILM policy to an index template. Then, click **Add policy**.

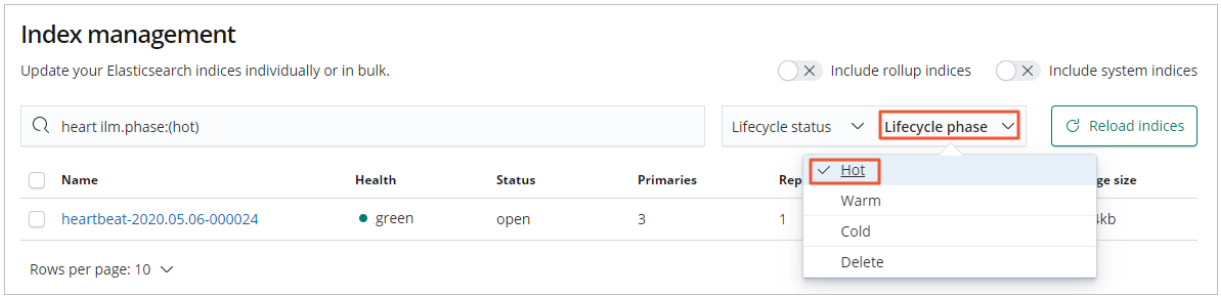


If the ILM policy is attached to the index, the information shown in the following figure appears.



Step 6: View indexes in different phases

To view indexes in the hot phase, select **Hot** from the **Lifecycle phase** drop-down list in the **Index management** section.



You can use this method to view indexes in other phases.

FAQ

Q: How do I configure a check interval for an ILM policy?

A: The system periodically checks for indexes that match an ILM policy. The default interval is 10 minutes. If the system detects matched indexes, it rolls over data for the indexes. For example, you set max_docs to 100 when you create an ILM policy. In this case, if the system detects that the number of documents in an index reaches 100 during a check, it triggers a rollover for the index. You can use the indices.lifecycle.poll_interval parameter to control the check interval. This ensures that data is rolled over for indexes in a timely manner.

> **Notice** Set this parameter to an appropriate value. A small value may cause node overload. In this example, this parameter is set to 1m.

```
PUT _cluster/settings
{
  "transient": {
    "indices.lifecycle.poll_interval":"1m"
  }
}
```

9.2.2. Use ILM to separate hot data from cold data

This topic describes how to use the index lifecycle management (ILM) feature to separate hot data from cold data in an Alibaba Cloud Elasticsearch cluster. The separation enables you to implement the hot-warm architecture. This architecture improves the read/write performance of the cluster, automates the maintenance of hot and cold data, and reduces your production costs.

Context

In the era of big data, data constantly changes. Data stored in Elasticsearch increases over time. When the data volume reaches a specific level, the memory usage, CPU utilization, and I/O throughput also increase. This affects the full-text search capability of Elasticsearch. To address this issue, Elasticsearch V6.6.0 and later provide the ILM feature. You can use this feature to create, set, enable, disable, or delete Elasticsearch indexes throughout their lifecycle. You can use the feature for time series data, cold data, and hot data to reduce data storage costs. This topic uses cold and hot data to demonstrate how to use the ILM feature. Business scenario:

1. Write data to the indexes of an Elasticsearch cluster in real time. When the data volume in the cluster reaches a specific level, the system automatically rolls over data to new indexes.
2. The new indexes stay in the hot phase for 30 minutes and enter the warm phase.
3. In the warm phase, the system shrinks the new indexes and merges the segments in the indexes. The indexes stay in the warm phase for 30 minutes and enter the cold phase.
4. In the cold phase, data is migrated from hot nodes to warm nodes to separate hot data from cold data. The indexes are deleted one hour later.

Recommended configurations

- You must configure ILM policies based on your business model. For example, we recommend that you configure different aliases and ILM policies for indexes with different structures. This facilitates index management.
- The name of an initial index must end with an auto-increment six-digit number, such as -000001. Otherwise, ILM policies cannot take effect. For example, an initial index is named myindex-000001. After a rollover, a new index named myindex-000002 is generated. If the names of your indexes do not meet the preceding requirements, we recommend that you reindex your data.
- In the hot phase, the system writes data. To ensure that data is written in chronological order, we recommend that you do not write data to indexes in the warm or cold phase. For example, for the warm phase, set `actions` to `shrink` or `read only`. This way, indexes are read only after they enter the warm phase.

> **Note** For more information about each lifecycle phase, see Use ILM to manage Heartbeat indexes.

- You configure more vCPUs and use disks with higher I/O performance for hot nodes to process hot data. You configure more disk space for warm nodes to store cold data. Warm nodes can still provide services even if you configure fewer vCPUs and use disks with lower I/O performance for them.

Configure an ILM policy for indexes

1. Log on to the Kibana console of your Elasticsearch cluster.

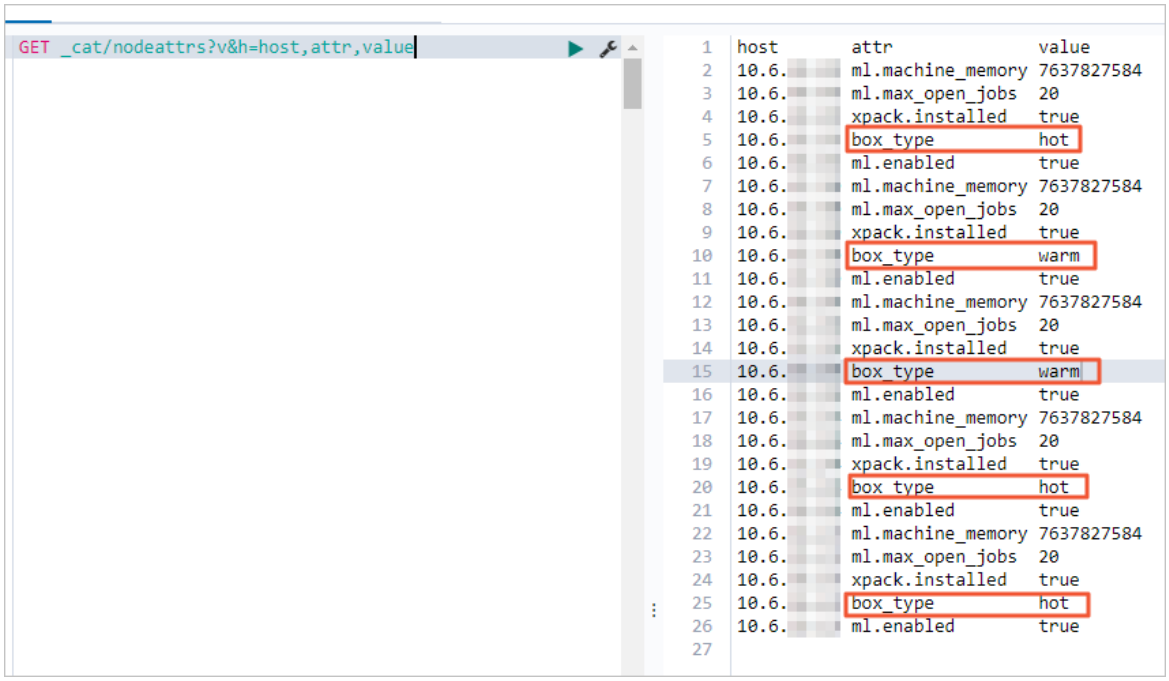
   For more information, see Log on to the Kibana console.

2. In the left-side navigation pane, click **Dev Tools**.

3. On the **Console** tab of the page that appears, run the following command to view the attributes of nodes:

   ```
   GET _cat/nodeattrs?v&h=host,attr,value
   ```

   If the command is successfully executed, the result shown in the following figure is returned. This figure shows that the Elasticsearch cluster contains three hot nodes and three warm nodes to support the hot-warm architecture.

   > **Note** When you purchase an Elasticsearch cluster, you must purchase warm nodes. The system then automatically deploys the hot-warm architecture. For more information, see Use ILM to manage Heartbeat indexes.

4. Call an API operation to define an ILM policy.

```
PUT /_ilm/policy/game-policy
{
  "policy": {
    "phases": {
      "hot": {
        "actions": {
          "rollover": {
            "max_size": "1GB",
            "max_age": "1d",
            "max_docs": 1000
          }
        }
      },
      "warm": {
        "min_age": "30m",
        "actions": {
          "forcemerge": {
                "max_num_segments":1
              },
          "shrink": {
                "number_of_shards":1
              }
        }
      },
      "cold": {
        "min_age": "1h",
        "actions": {
          "allocate": {
            "require": {
              "box_type": "warm"
            }
          }
        }
      },
      "delete": {
        "min_age": "2h",
        "actions": {
          "delete": {}
        }
      }
    }
  }
}
```

Note

- After an ILM policy is created, you cannot change the policy name.
- In this step, you can specify the `max_age` parameter in the minimum unit of seconds. If you use the Kibana console to create an ILM policy, you can specify this parameter only in the minimum unit of hours.

5. Create an index template.

In the `settings` configuration, specify the hot attribute. This way, data can be stored in hot nodes after it is written.

```
PUT _template/gamestabes_template
{
  "index_patterns" : ["gamestabes-*"],
  "settings": {
    "index.number_of_shards": 5,
    "index.number_of_replicas": 1,
    "index.routing.allocation.require.box_type":"hot",
    "index.lifecycle.name": "game-policy",
    "index.lifecycle.rollover_alias": "gamestabes"
  }
}
```

Parameter	Description
`index.routing.allocation.require.box_type`	The type of nodes to which newly created indexes are allocated.
`index.lifecycle.name`	The name of the ILM policy.
`index.lifecycle.rollover_alias`	The alias of the index that is generated during a rollover.

6. Create an index based on an auto-increment number.

```
PUT gamestabes-000001
{
"aliases": {
    "gamestabes":{
       "is_write_index": true
        }
      }
}
```

You can also create an index based on time. For more information, see Using date math.

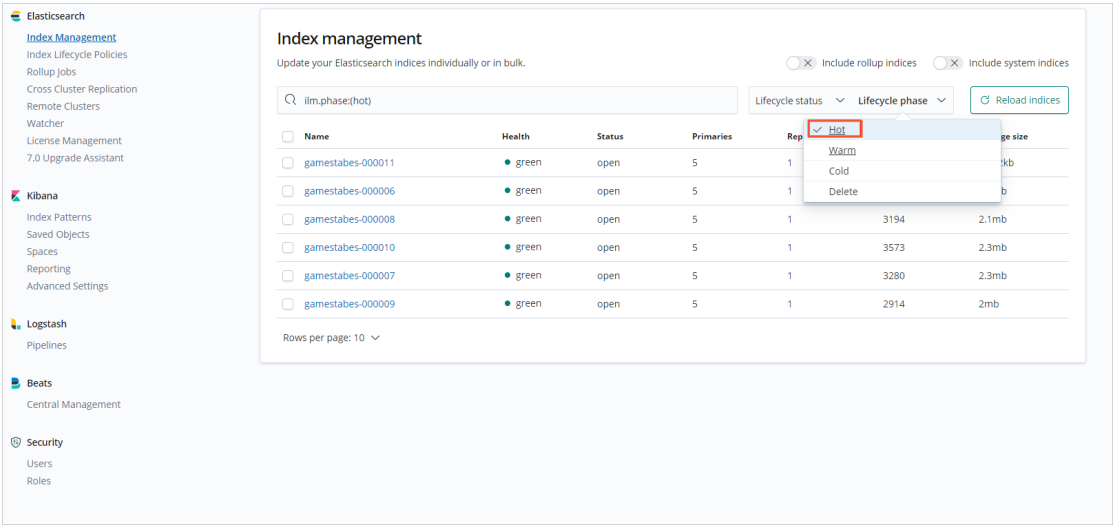
7. Use the index alias to write data.

The system periodically checks for indexes that match an ILM policy. If the system finds matched indexes, it rolls over the data in the indexes.

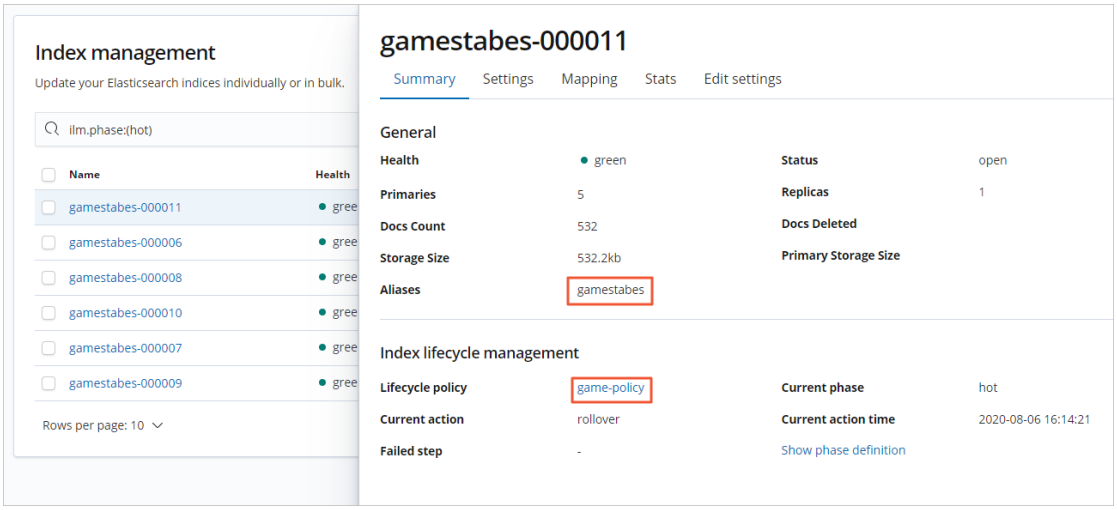
```
PUT gamestabes/_doc/1
{
    "EU_Sales" : 3.58,
    "Genre" : "Platform",
    "Global_Sales" : 40.24,
    "JP_Sales" : 6.81,
    "Name" : "Super Mario Bros.",
    "Other_Sales" : 0.77,
    "Platform" : "NES",
    "Publisher" : "Nintendo",
    "Year_of_Release" : "1985",
    "na_Sales" : 29.08
}
```

Note By default, the system checks for indexes that match an ILM policy at 10-minute intervals. You can specify the `indices.lifecycle.poll_interval` parameter to change the check interval. After the data in an index is rolled over, it enters the next phase.

8. Filter indexes based on lifecycle phases and view detailed index configurations.
   i. In the left-side navigation pane, click **Management**.
   ii. In the **Elasticsearch** section, click **Index Management**.
   iii. In the **Index management** section, click **Lifecycle phase** next to **Lifecycle status** and select a phase.

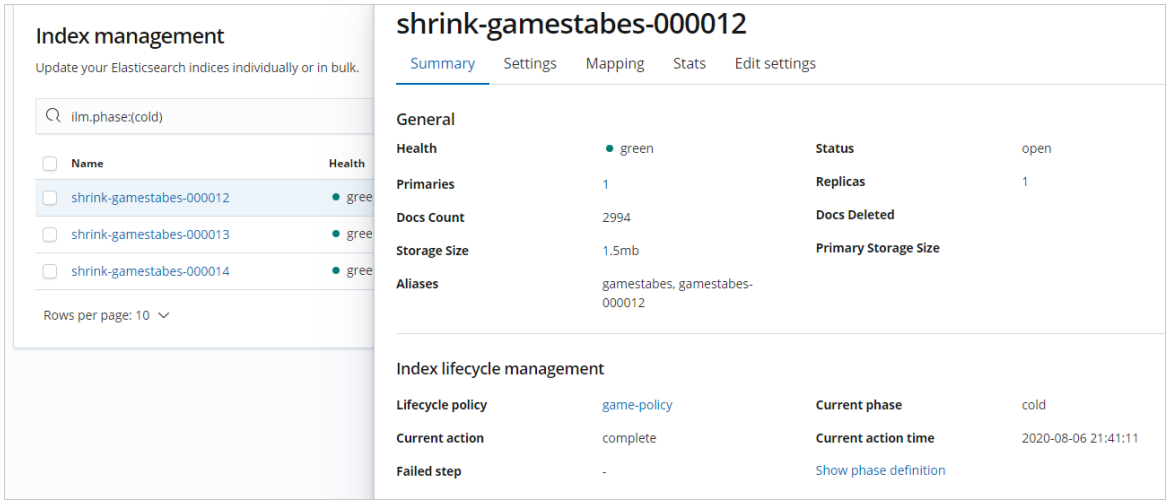


iv. Click an index name to view its details.



Verify data distribution

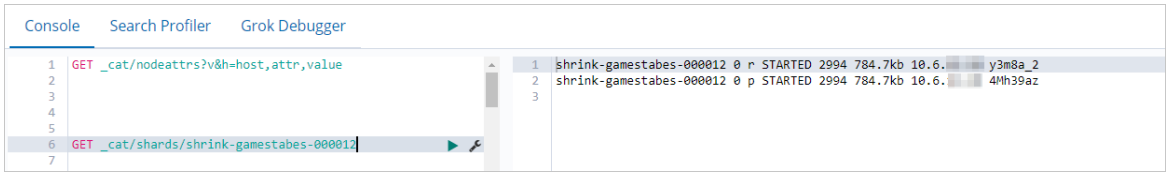
1. Query indexes in the cold phase and view their configurations.



2. Query the distribution of shards for indexes in the cold phase.

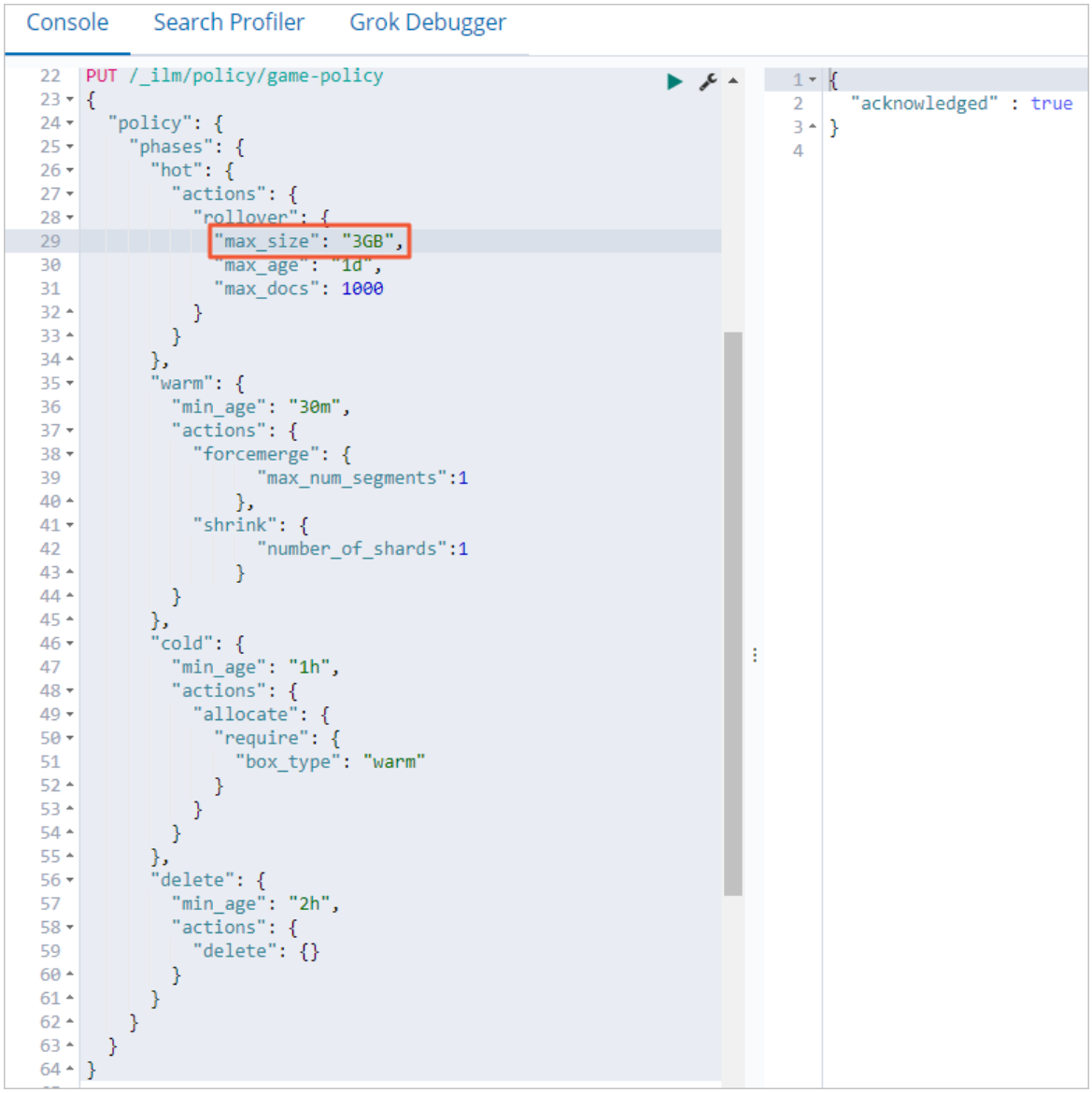
```
GET _cat/shards?shrink-gamestables-000012
```

If the command is successfully executed, the result shown in the following figure is returned. This figure shows that data in the indexes is mainly distributed on warm nodes.



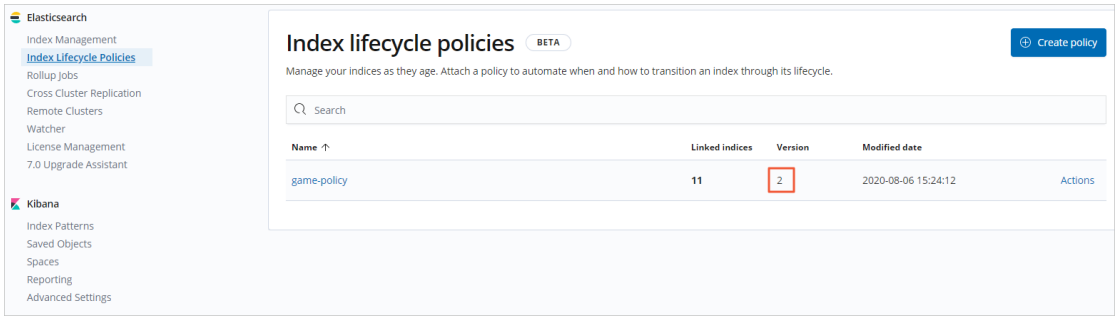
Update the ILM policy

1. Update the running ILM policy.



2. View the version of the updated policy.
   i. In the left-side navigation pane, click **Management**.
   ii. In the **Elasticsearch** section, click **Index Lifecycle Policies**.
   iii. In the **Index lifecycle policies** section, view the version of the updated policy.

   The version number of the updated policy is one more than that of the original policy. The updated policy takes effect from the next rollover.



Switch the ILM policy

1. Create another ILM policy.

```
PUT /_ilm/policy/game-new
{
  "policy": {
    "phases": {
      "hot": {
        "actions": {
          "rollover": {
            "max_size": "3GB",
            "max_age": "1d",
            "max_docs": 1000
          }
        }
      },
      "warm": {
        "min_age": "30m",
        "actions": {
          "forcemerge": {
                "max_num_segments":1
              },
          "shrink": {
                "number_of_shards":1
              }
        }
      },
      "cold": {
        "min_age": "1h",
        "actions": {
          "allocate": {
            "require": {
              "box_type": "warm"
            }
          }
        }
      },
      "delete": {
        "min_age": "2h",
        "actions": {
          "delete": {}
        }
      }
    }
  }
}
```

2. Associate the new policy with the index template.

```
PUT _template/gamestabes_template
{
  "index_patterns" : ["gamestabes-*"],
  "settings": {
    "index.number_of_shards": 5,
    "index.number_of_replicas": 1,
    "index.routing.allocation.require.box_type":"hot",
    "index.lifecycle.name": "game-new",
    "index.lifecycle.rollover_alias": "gamestabes"
  }
}
```

Notice

- The new policy takes effect from the next rollover.
- If you want to associate the new policy with the indexes that are created based on the original policy, you can run the `PUT gamestabes-*/_settings` command. For more information, see Switching policies for an index.

Summary

This topic provides instructions on how to separate hot data from cold data by using ILM.

- Configure an ILM policy for indexes.

  Procedure:

  i. Configure hot and warm attributes.
  ii. Configure an index template based on your needs.
  iii. Configure an ILM policy based on your needs and associate the policy with the index template.
  iv. Create an initial index whose name ends with -000001. The name of the index generated after a rollover is automatically incremented by one.

- Verify data distribution.

  Check whether the shards of indexes in the cold phase are distributed on warm nodes.

- Update the ILM policy.

  Update the ILM policy.

- Switch the ILM policy.

  Switch the ILM policy.

9.3. Application of X-Pack advanced features

9.3.1. Use the CCR feature to migrate data

This topic describes how to use the cross-cluster replication (CCR) feature to migrate data between a local Alibaba Cloud Elasticsearch cluster and a remote Alibaba Cloud Elasticsearch cluster.

Background information

CCR is a commercial feature released in open source Elasticsearch Platinum. After you purchase an Alibaba Cloud Elasticsearch cluster, you can use this feature free of charge based on a few simple configurations. Only single-zone Elasticsearch clusters of V6.7.0 or later support this feature. CCR is used in the following scenarios:

- Disaster recovery and high availability

  You can use CCR to back up data among Elasticsearch clusters that reside in different regions. If a cluster fails, you can retrieve its index data from other clusters. This prevents data loss.

- Data access from a nearby cluster

  For example, Company A has multiple subsidiaries that are located in different regions. To speed up business processing, you can plan the business of the subsidiaries based on their geographical locations. Then, use CCR to distribute business data to Elasticsearch clusters in different regions. Each subsidiary can directly use the cluster in the region where the subsidiary is located to process business.

- Centralized reporting

  You can use CCR to replicate data from multiple small clusters to one cluster. Then, you can perform visualized analytics and reporting for the data in a centralized manner.

To use CCR, you must prepare two types of clusters: local clusters and remote clusters. Remote clusters provide source data, which is stored in leader indexes. Local clusters replicate the data and store it in follower indexes. You can also use CCR to migrate large volumes of data at a time in real time. For more information, see Cross-cluster replication.

Procedure

1. Preparations

   Prepare a local cluster, a remote cluster, and a leader index.

2. Step 1: Connect clusters

   Connect the remote cluster to the local cluster.

3. Step 2: Add the remote cluster

   In the Kibana console of the local cluster, add the remote cluster.

4. Step 3: Configure CCR

   In the Kibana console of the local cluster, configure the leader index and a follower index.

5. Step 4: View migration results

   Insert data into the remote cluster. Then, verify the data migration on the local cluster.

Preparations

1. Create a local cluster and a remote cluster.

   For more information, see Create an Alibaba Cloud Elasticsearch cluster. The two clusters must be single-zone clusters, reside in the same virtual private cloud (VPC) and vSwitch, and be of the same version (V6.7.0 or later).

2. Log on to the Kibana console of the remote cluster and create a leader index.

> **Notice**
>
> - If you create an index in an Elasticsearch cluster of V7.0 or earlier, you must enable the soft_deletes attribute. Otherwise, an error is reported.
> - If you want to migrate data in an existing index, you can call the reindex API to enable the soft_deletes attribute.

```
PUT myindex
{
  "settings": {
    "index.soft_deletes.retention.operations": 1024,
    "index.soft_deletes.enabled": true
  }
}
```

3. Disable the physical replication feature for the leader index.

   The physical replication feature is automatically enabled for indexes in Elasticsearch V6.7.0 clusters. Before you use CCR, you must disable the physical replication feature.

   i. Disable the index.

   ```
   POST myindex/_close
   ```

   ii. Update the settings configuration of the index to disable the physical replication feature.

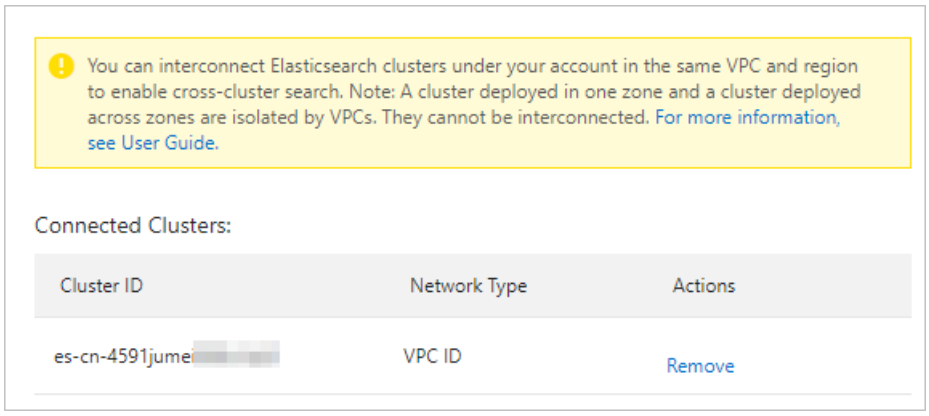
   ```
   PUT myindex/_settings
   {
   "index.replication.type" : null
   }
   ```

   iii. Enable the index.

   ```
   POST myindex/_open
   ```

Step 1: Connect clusters

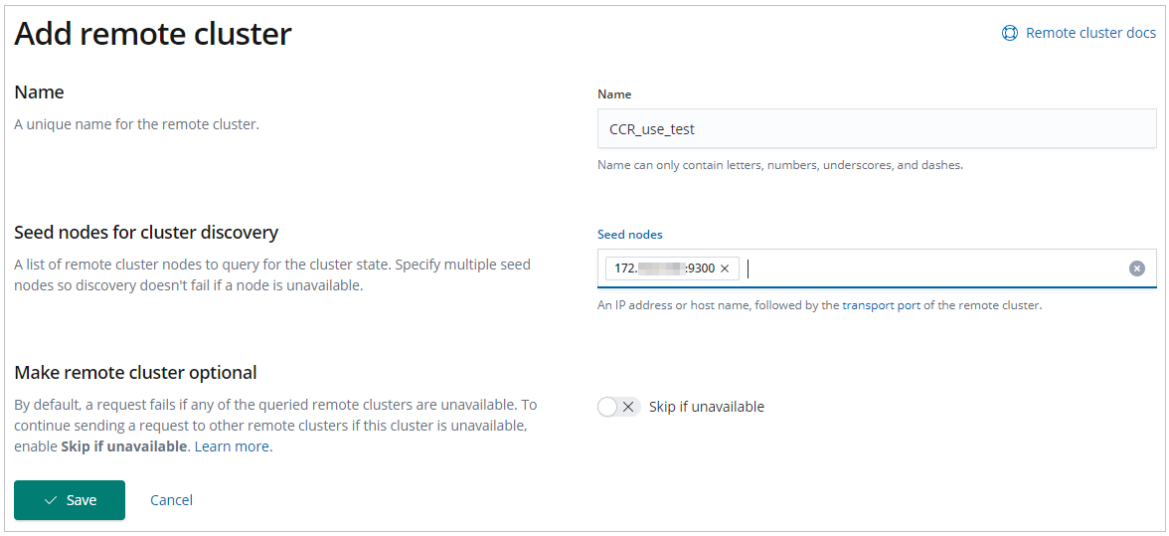
Configure the remote cluster to connect it to the local cluster. For more information, see Connect Elasticsearch clusters. If the two clusters are connected, the information shown in the following figure appears.



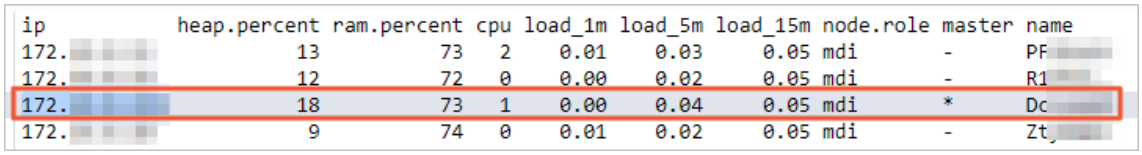
Step 2: Add the remote cluster

1. Log on to the Kibana console of the local cluster.

   For more information, see Log on to the Kibana console.
2. In the left-side navigation pane, click **Management**.
3. In the **Elasticsearch** section, click **Remote Clusters**.
4. Click **Add a remote cluster**.
5. In the **Add remote cluster** section, configure the following parameters.



   - **Name**: the name of the remote cluster. The name must be unique.
   - **Seed nodes**: the nodes in the remote cluster. Specify each node in the format of Node IP address:9300. To obtain the IP addresses of nodes, log on to the Kibana console of the remote cluster and run the `GET /_cat/nodes?v` command on the Console tab of the Dev Tools page. The nodes you specify must include a dedicated master node of the remote cluster. We recommend that you specify multiple nodes. This ensures that you can still use CCR when the specified dedicated master node fails.

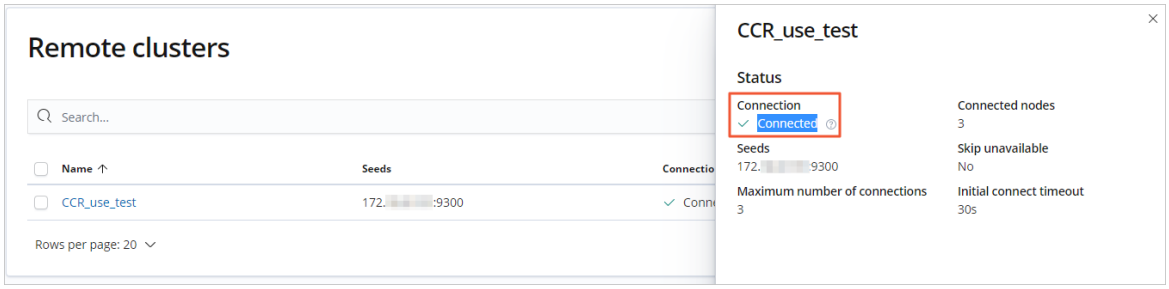


ip	heap.percent	ram.percent	cpu	load_1m	load_5m	load_15m	node.role	master	name
172.	13	73	2	0.01	0.03	0.05	mdi	-	PF
172.	12	72	0	0.00	0.02	0.05	mdi	-	R1
172.	18	73	1	0.00	0.04	0.05	mdi	*	Dc
172.	9	74	0	0.01	0.02	0.05	mdi	-	Zt

> **Notice** During CCR, Kibana uses the IP addresses of data nodes to access clusters over TCP port 9300. HTTP port 9200 is not supported.

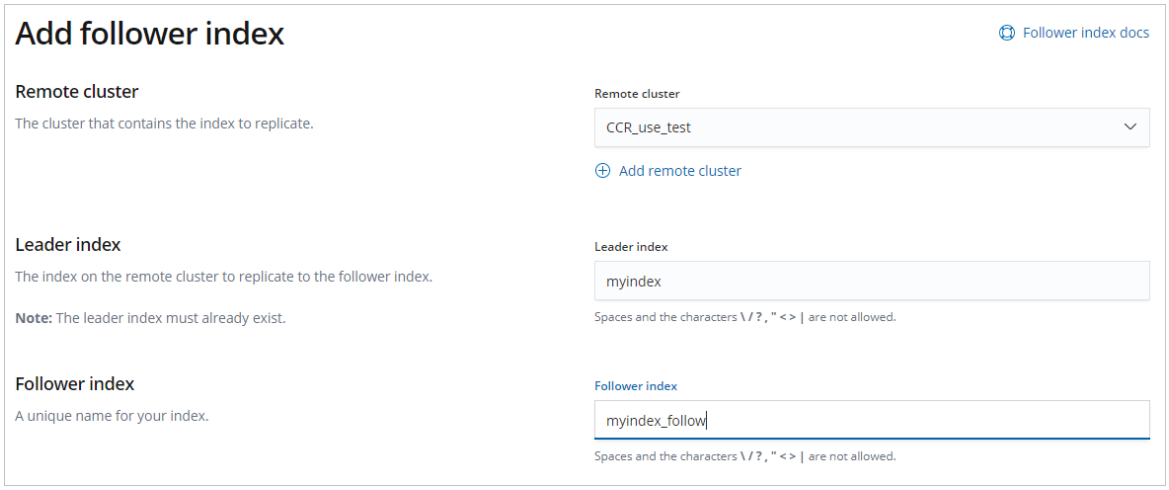
6. Click **Save**.

   The system then automatically connects to the remote cluster. If the connection is established, **Connected** appears.



Step 3: Configure CCR

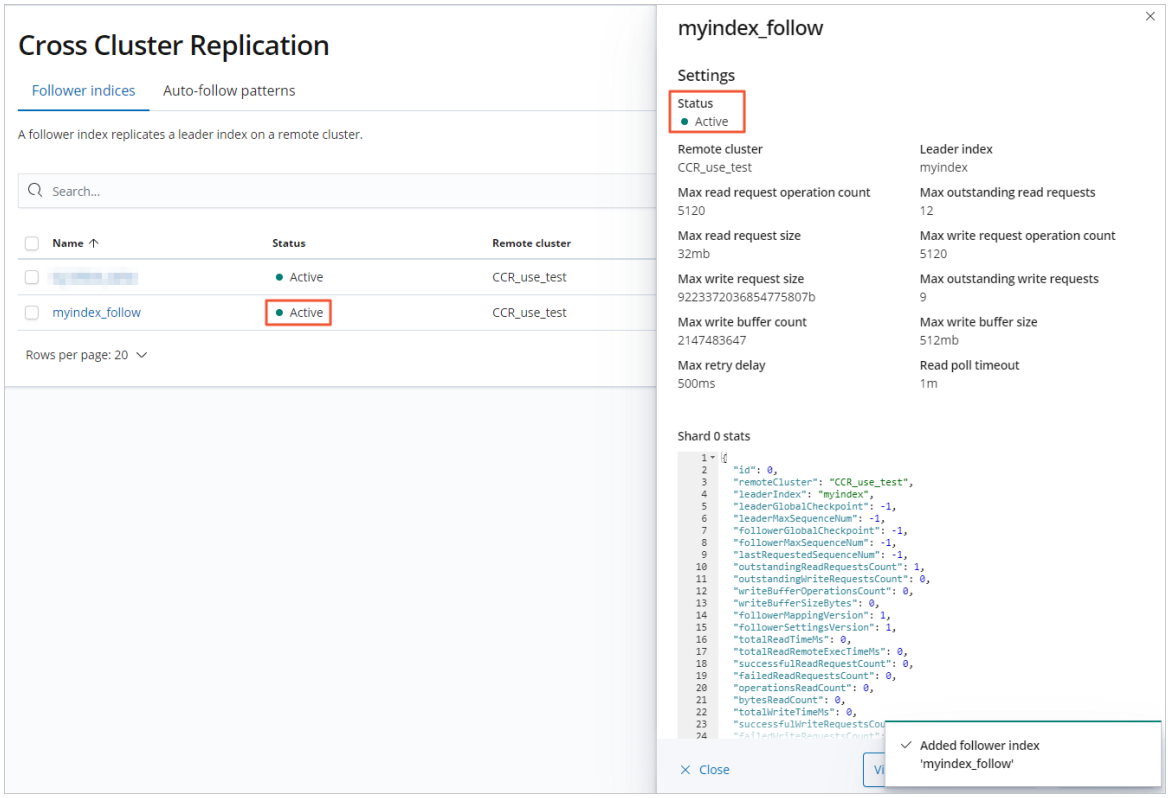
1. Log on to the Kibana console of the local cluster. In the left-side navigation pane, click **Management**. In the **Elasticsearch** section of the page that appears, click **Cross Cluster Replication**.
2. On the Follower indices tab, click **Create a follower index**.
3. In the **Add follower index** section, configure the following parameters.



Parameter	Description
Remote cluster	Select the cluster you added in Step 2: Add the remote cluster.
Leader index	The index whose data you want to migrate. In this example, the **myindex** index that is created in Preparations is used.
Follower index	The index to which data is migrated. You must specify a unique index name.

4. Click **Create**.

   After the follower index is created, the index is in the **Active** state.



Step 4: View migration results

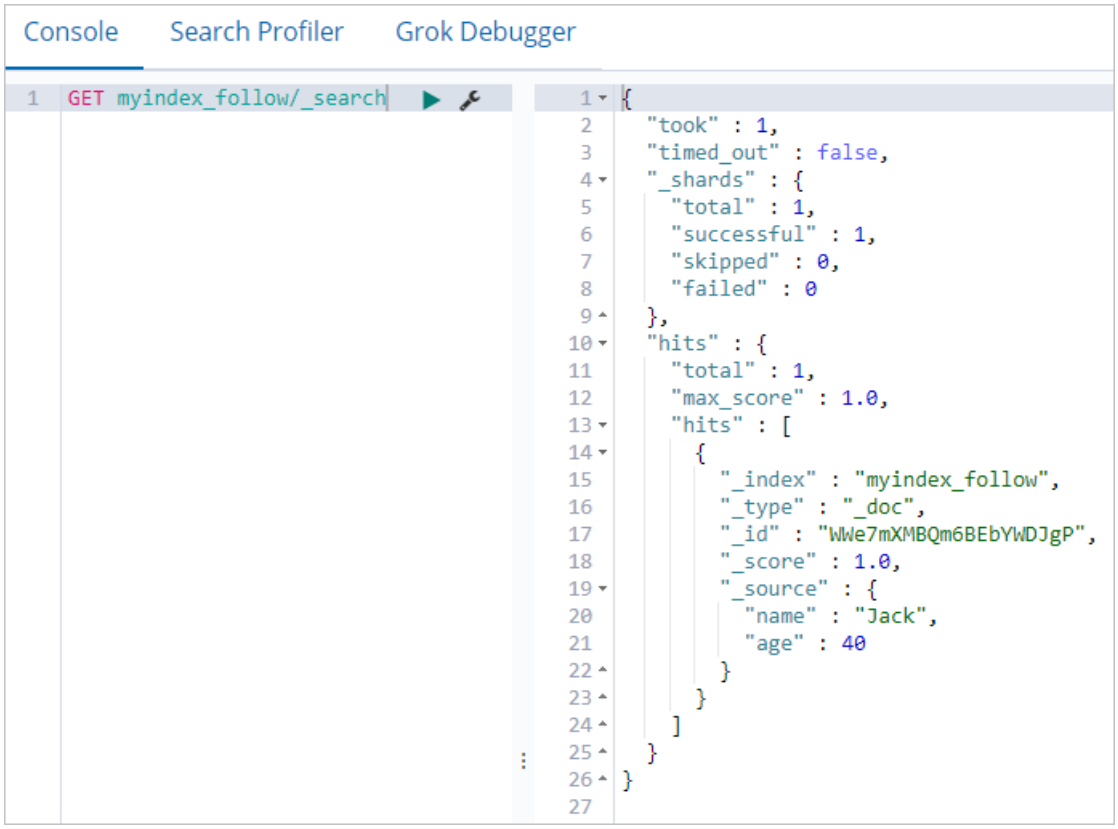
1. Log on to the Kibana console of the remote cluster and insert data into the remote cluster.

```
POST myindex/_doc/
{
  "name":"Jack",
  "age":40
}
```

2. Run the following command in the Kibana console of the local cluster to check whether the inserted data is migrated to the local cluster:

```
GET myindex_follow/_search
```

If the command is successfully run, the result shown in the following figure is returned.



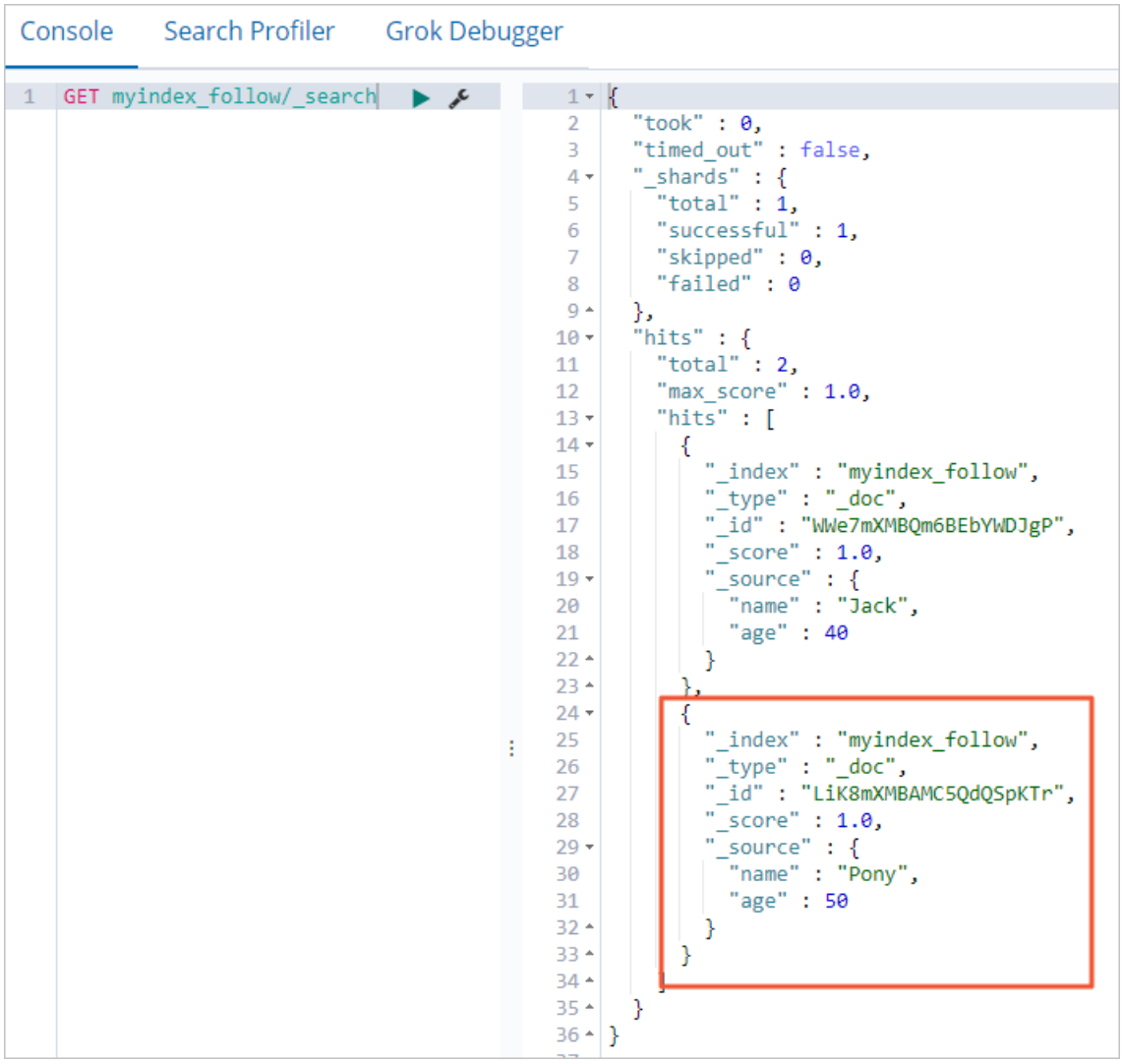
The preceding figure shows that data in the leader index myindex of the remote cluster is migrated to the follower index myindex_follow of the local cluster.

> **Notice** The follower index myindex_follow is read-only. If you want to write data to the follower index, convert the follower index into a common index first. For more information, see Use Elasticsearch CCR to migrate data across data centers.

3. Insert a data record into the remote cluster and check whether the data record is migrated to the local cluster in real time.

```
POST myindex/_doc/
{
  "name":"Pony",
  "age":50
}
```

Query the inserted data record in the local cluster. The following figure shows the data record.



The preceding figure shows that the CCR feature can implement real-time migration of incremental data.

> **Note** You can also call the APIs for the CCR feature to perform cross-cluster replication operations. For more information, see Cross-cluster replication APIs.

FAQ

Q: I can use port 9300 to add a remote cluster. Why is only port 9200 accessible when I use a domain name to access an Elasticsearch cluster?

A: Port 9300 is an open port. However, when you access a cluster over the Internet, Server Load Balancer (SLB) enables only port 9200 during port verification for security purposes. This will be adjusted in the future.

9.3.2. Use X-Pack to configure LDAP authentication

This topic describes how to configure Lightweight Directory Access Protocol (LDAP) authentication for an Alibaba Cloud Elasticsearch cluster to allow LDAP users with the required roles to access the cluster.

Prerequisites

- An Alibaba Cloud Elasticsearch cluster is created. In this example, an Elasticsearch V6.7.0 cluster is used.

  For more information, see Create an Alibaba Cloud Elasticsearch cluster.

  > **Note** You cannot configure LDAP authentication for an Elasticsearch cluster of V7.0.0 or later in the Elasticsearch console.

- A private connection is configured for the Elasticsearch cluster if the cluster is deployed in the new network architecture. To configure a private connection for an Elasticsearch cluster, perform the following steps:

  > **Note**

  i. Create a Classic Load Balancer (CLB) instance that supports the PrivateLink service and resides in the same VPC as the Elasticsearch cluster.

     For more information, see Step 1: Create a CLB instance that supports PrivateLink.

  ii. Configure the CLB instance.

     For more information, see Step 2: Configure the CLB instance.

     > **Note** You must add the Elastic Compute Service (ECS) instance for which LDAP is configured to the CLB instance as a backend server. In this topic, port 389 is used as the listening port.

  iii. Create an endpoint service.

     For more information, see Step 3: Create an endpoint service.

  iv. Configure a private connection to the Elasticsearch cluster.

     For more information, see Step 4: Configure a private connection for the Elasticsearch cluster.

  v. Obtain the domain name of the endpoint that is used to access the endpoint service.

     For more information, see View the domain name of an endpoint.

- The LDAP service is activated in the VPC where the Elasticsearch cluster resides. In this topic, OpenLDAP 2.4.44 is used.
- The LDAP environment and user data are prepared.

  For more information, see Official LDAP documentation.

Limits

You can configure LDAP authentication in the Elasticsearch console only for Elasticsearch clusters whose versions are earlier than V7.0.0.

Precautions

-

- Elasticsearch clusters created in October 2020 or later are deployed in the new network architecture. In this network architecture, the LDAP authentication feature is limited. To resolve this issue, you can use the PrivateLink service to establish private connections between VPCs. For more information, see Configure a private connection for an Elasticsearch cluster. If you want to connect such a cluster to the Internet, configure an NGINX proxy to forward requests.
- In the original network architecture, only single-zone Elasticsearch clusters support LDAP authentication. In the new network architecture, both single-zone and multi-zone Elasticsearch clusters support LDAP authentication if you use the PrivateLink service.

Procedure

1. Step 1: Configure LDAP authentication
2. Step 2: Map the user to a role
3. Step 3: Verify the result

Step 1: Configure LDAP authentication

You can use X-Pack to configure LDAP authentication in the following modes:

- User search mode
- Distinguished name (DN) template-based mode

The user search mode is commonly used. In user search mode, a user who has permissions to query the LDAP directory is used to search for the DN of a user who you want to authenticate. The search is performed based on the username and LDAP attribute that are provided by X-Pack. After the DN of the user is found, X-Pack attempts to bind the user to the LDAP directory by using the DN and the related password to authenticate the user. For more information, see Configure an LDAP realm.

The following sample code provides the mapping configurations that are required by LDAP to manage a DN. You must add the configurations to the YML file of the Elasticsearch cluster.

```
xpack.security.authc.realms.ldap1.type: ldap
xpack.security.authc.realms.ldap1.order: 0
xpack.security.authc.realms.ldap1.url: "ldap://ep-bp1dhpobznlgjhj9****-cn-hangzhou-i.epsrv-
bp1q8tcj2jjt5dwr****.cn-hangzhou.privatelink.aliyuncs.com:389"
xpack.security.authc.realms.ldap1.bind_dn: "cn=zhang lei,ou=support,dc=yaobili,dc=com"
xpack.security.authc.realms.ldap1.bind_password: 123456
xpack.security.authc.realms.ldap1.user_search.base_dn: "ou=support,dc=yaobili,dc=com"
xpack.security.authc.realms.ldap1.user_search.filter: "(cn={0})"
xpack.security.authc.realms.ldap1.group_search.base_dn: "ou=support,dc=yaobili,dc=com"
xpack.security.authc.realms.ldap1.unmapped_groups_as_roles: false
```

Parameter	Description
type	The type of the realm. You must set this parameter to ldap.

Parameter	Description
url	The URL and port number that are used to connect to the LDAP server. ldap indicates that a common connection and port 389 are used. ldaps indicates that an SSL-encrypted connection and port 636 are used. **Note** If your Elasticsearch cluster is deployed in the new network architecture, you must specify this parameter in the format of `Domain name of the endpoint:Port number`. In this example, `ep-bp1dhpobznlgjhj9****-cn-hangzhou-i.epsrv-bp1q8tcj2jjt5dwr****.cn-hangzhou.privatelink.aliyuncs.com:389` is used.
bind_dn	The DN of the user who you want to search for and bind to the LDAP directory. This parameter is valid only in user search mode.
bind_password	The password of the user.
user_search.base_dn	The container DN that is used to search for the user.
group_search.base_dn	The container DN that is used to search for the group to which the user belongs. If you do not specify this parameter, Elasticsearch searches for the attribute that is specified by the user_group_attribute parameter to determine the group to which the user belongs.
unmapped_groups_as_roles	The default value of this parameter is false. If you set this parameter to true, the names of unmapped LDAP groups are used as role names.

After you add the preceding configurations, click **OK** to restart the cluster. For more information about the parameters, see Security settings in Elasticsearch.

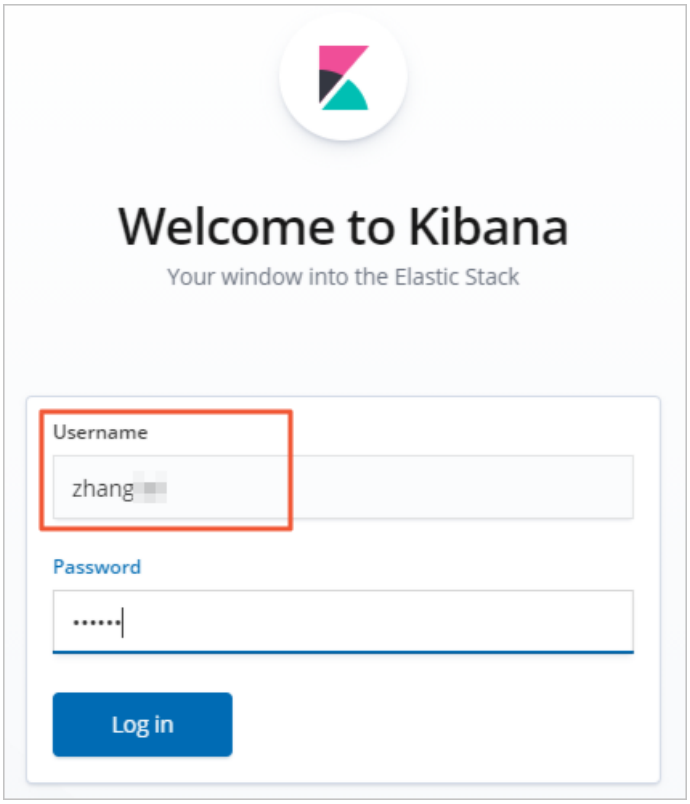
Step 2: Map the user to a role

Run the following command to map the zhang* account to the administrator role:

```
POST _xpack/security/role_mapping/ldap_super_user1?pretty
{
  "roles": [ "superuser" ],
  "enabled": true,
  "rules": {
    "any": [
      {
        "field": {
          "username": "zhang*"
        }
      }
    ]
  }
}
```

Step 3: Verify the result

Log on to the Kibana console of the Elasticsearch cluster by using the zhang* account.



Run the following command:

```
PUT _cluster/settings
{
  "persistent": {
    "action.auto_create_index": true
  }
}
```

If the result shown in the following figure is returned, the account has the required permissions.

9.3.3. Use the RBAC mechanism provided by Elasticsearch X-Pack to implement access control

If you want to grant access permissions on items such as clusters, indexes, and fields, you can use the role-based access control (RBAC) mechanism that is provided by the X-Pack plug-in of Elasticsearch. This mechanism allows you to grant permissions to custom roles and assign the roles to users to implement access control. Elasticsearch provides a variety of built-in roles. You can create custom roles based on the built-in roles to meet your business requirements. This topic describes how to create and configure a custom role to implement access control.

Context

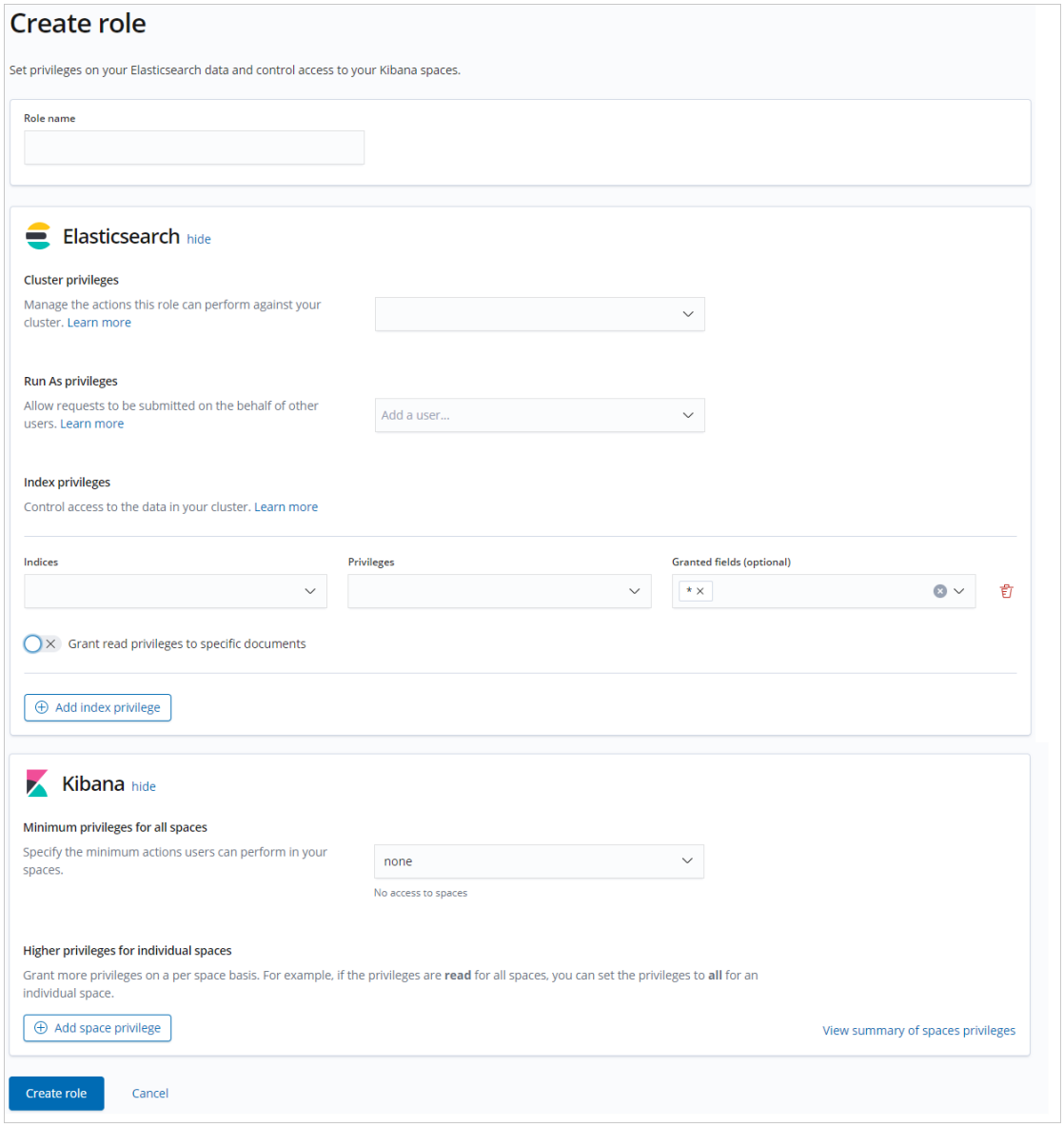
- Elasticsearch supports the RBAC mechanism that is provided by the X-Pack plug-in. For more information, see User authorization.
- Elasticsearch supports various security authentication features. For more information, see Identity authentication and authorization in Elasticsearch.

Procedure

> **Note** An Elasticsearch V6.7.0 cluster is used in this topic. Operations on clusters of other versions may differ. The actual operations in the console prevail.

1. Create a role.
   i. Log on to the Kibana console of your Elasticsearch cluster.

      For more information, see Log on to the Kibana console.
   ii. In the left-side navigation pane, click **Management**.
   iii. In the **Security** section, click **Roles**.
   iv. In the Roles section, click **Create role**.



Parameter	Description
Role name	The name of the role.
Cluster privileges	The operation permissions on the cluster, such as the permissions to view the health status and settings of the cluster and the permission to create snapshots. For more information, see Cluster privileges.
Run As privileges	The user who assumes the role. This parameter is optional. If you do not configure this parameter, you can assign the role to a user when you create the user. For more information, see Create a user.

Parameter	Description
Index privileges	The operation permissions on indexes. For example, if you want to grant the role the read-only permissions on all fields in all indexes, set the Indices parameter to an asterisk (*) and the Privileges parameter to read. You can set the Indices parameter to an asterisk or regular expression. For more information, see Indices privileges. When you configure the Index privileges parameter, you need to configure the following parameters: ■ **Indices**: the index pattern, such as heartbeat-*. **Note** If no index patterns are available, click **Index Pattern** in the **Kibana** section of the **Management** page and create an index pattern as prompted. ■ **Privileges**: the permissions that you want to grant to the role. ■ **Granted fields (optional)**: The fields on which you want to grant permissions. This parameter is optional.
Kibana privileges	The operation permissions on Kibana. **Notice** Versions earlier than Kibana V7.0 support only base privileges. Kibana V7.0 and later support base privileges and feature privileges. After you assign a base privilege to a role, the role has access permissions on all Kibana spaces. After you assign a feature privilege to a role, the role has access permissions only on a specific feature. To assign a feature privilege, you must specify a Kibana space.

When you create a role, you must grant permissions to the role. In this example, the following permissions are granted:

- Read-only permissions on a specific index

  For more information, see Configure read-only permissions on indexes.
- Permissions to view all or some dashboards

  For more information, see Configure operation permissions on dashboards.
- Read and write permissions on some indexes and read-only permissions on all clusters, such as the permissions to view the health statuses, snapshots, and settings of clusters, write data to indexes, or update index mappings

  For more information, see Configure read and write permissions on indexes and read-only permissions on clusters.

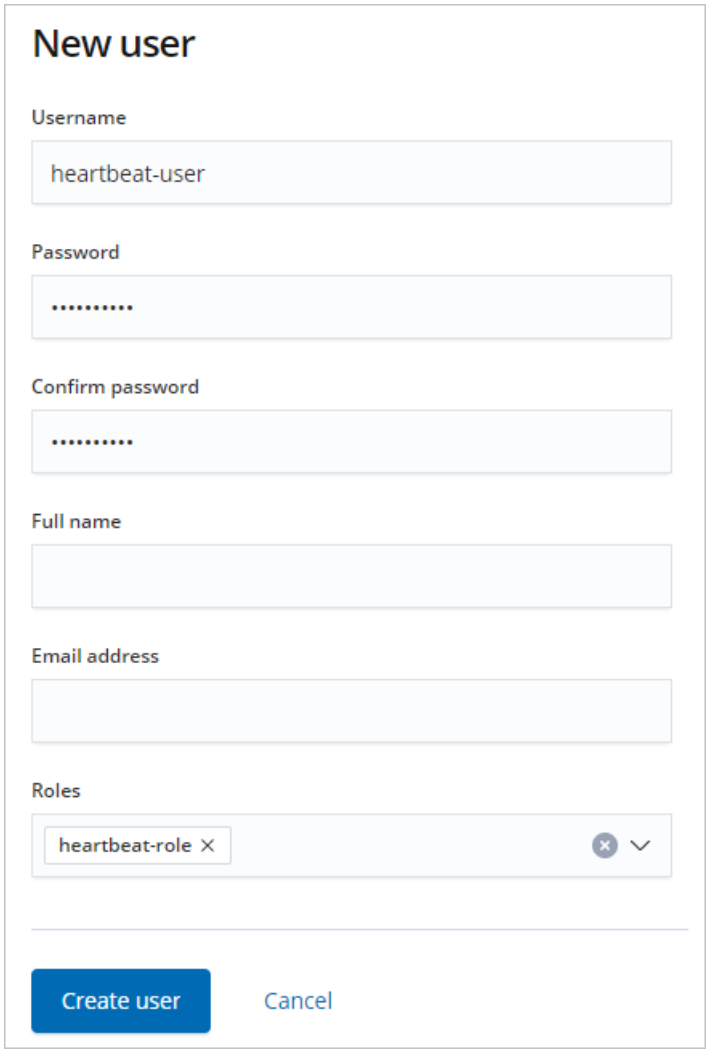
v. Click **Create role**.

2. Create a user and assign the role to the user.

   i. In the left-side navigation pane of the Kibana console, click **Management**.

ii. In the **Security** section, click **Users**.

iii. In the upper-right corner of the Users section, click **Create new user**.



Parameter	Description
Username	The username, which is used to log on to the Kibana console. You can customize a username.
Password	The password of the user, which is used to log on to the Kibana console. You can customize a password.
Confirm password	The value must be the same as that of the **Password** parameter.
Full name	The full name of the user, which can be customized.
Email address	The email address of the user.

Parameter	Description
Roles	The role that is assigned to the user. You can specify one or more roles. The roles can be built-in or custom roles. **Notice** If you specify a user when you create a role, you still need to configure this parameter. Otherwise, an error is reported when you use the user to log on to the Kibana console.

iv. Click **Create user**.

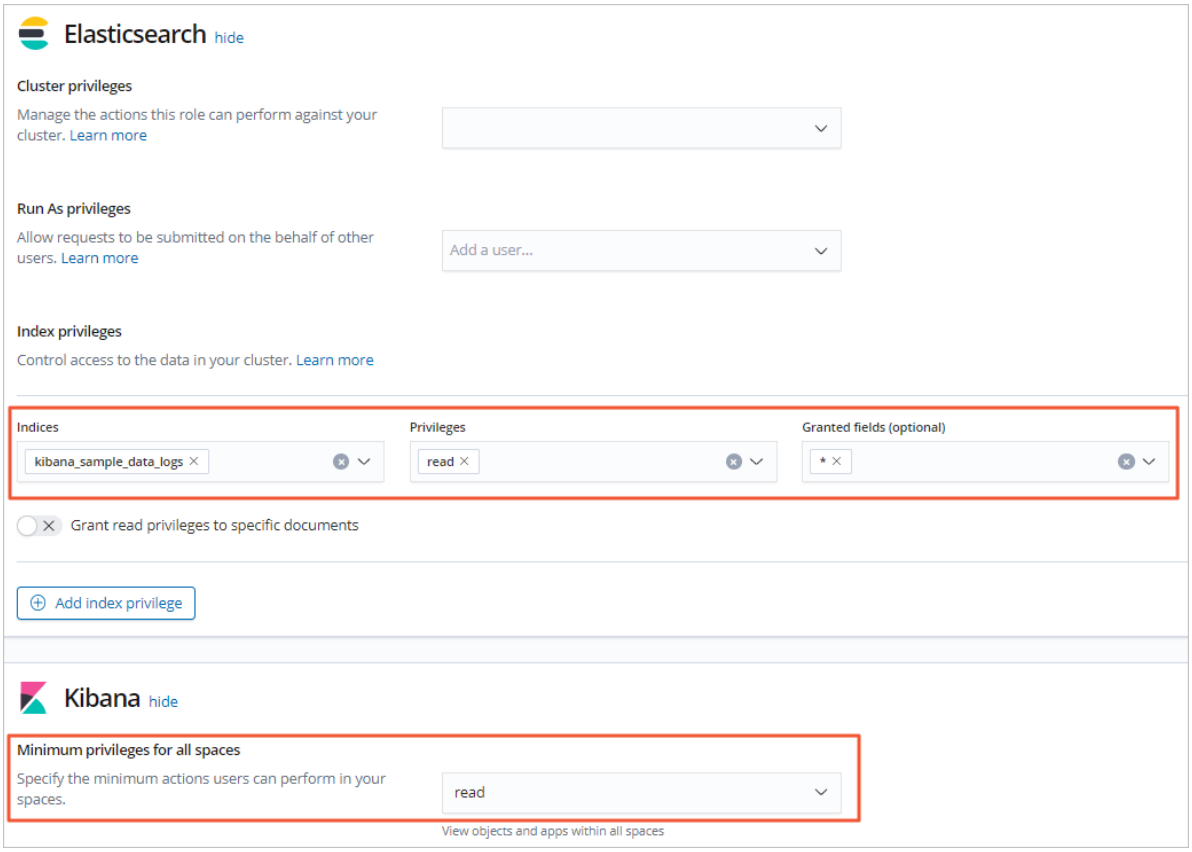
3. Use the user to log on to the Kibana console and perform operations to check whether the user has the related permissions.

Configure read-only permissions on indexes

- Scenario

  Grant the read-only permissions on a specific index to a common user. In this case, the user can query data from the index in the Kibana console but cannot access clusters.

- Role configuration



Permissions

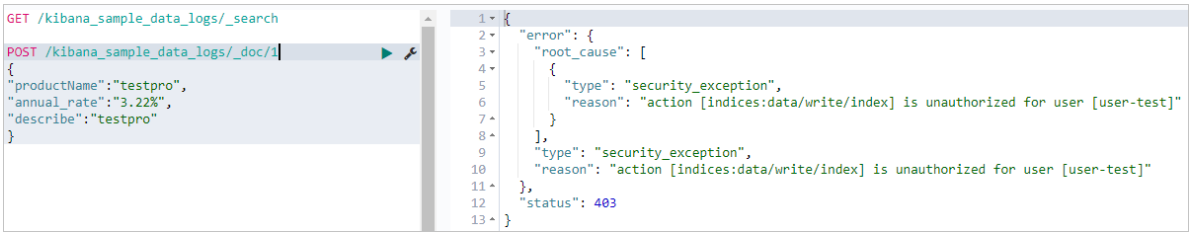
Permission type	Permission key	Permission value	Description
Index privileges	**indices**	kibana_sample_data_logs	The name of the index. You can specify a full index name, alias, wildcard, or regular expression. For more information, see Indices Privileges.
	privileges	read	The read-only permissions on the index. The read-only permissions include the permissions to call the count, explain, get, mget, scripts, search, and scroll APIs. For more information, see privileges-list-indices.
	Granted fields (optional)	*	The fields in the index. The value * indicates all fields.
Kibana privileges	**privileges**	read	The read-only permissions on Kibana. The permissions are granted to all spaces. Default value: none. This value indicates that no spaces are authorized to access Kibana. **Notice** Versions earlier than Kibana V7.0 support only base privileges. Kibana V7.0 and later support base privileges and feature privileges. After you assign a base privilege to a role, the role has access permissions on all Kibana spaces. After you assign a feature privilege to a role, the role has access permissions only on a specific feature. To assign a feature privilege, you must specify a Kibana space.

- Verification

  Use the common user to log on to the Kibana console and run an index read command. The system returns results as expected. Then, run an index write command. The system returns an error message. The message indicates that the user is not authorized to perform write operations.

```
GET /kibana_sample_data_logs/_search
```

```
POST /kibana_sample_data_logs/_doc/1
{
    "productName": "testpro",
    "annual_rate": "3.22%",
    "describe": "testpro"
}
```



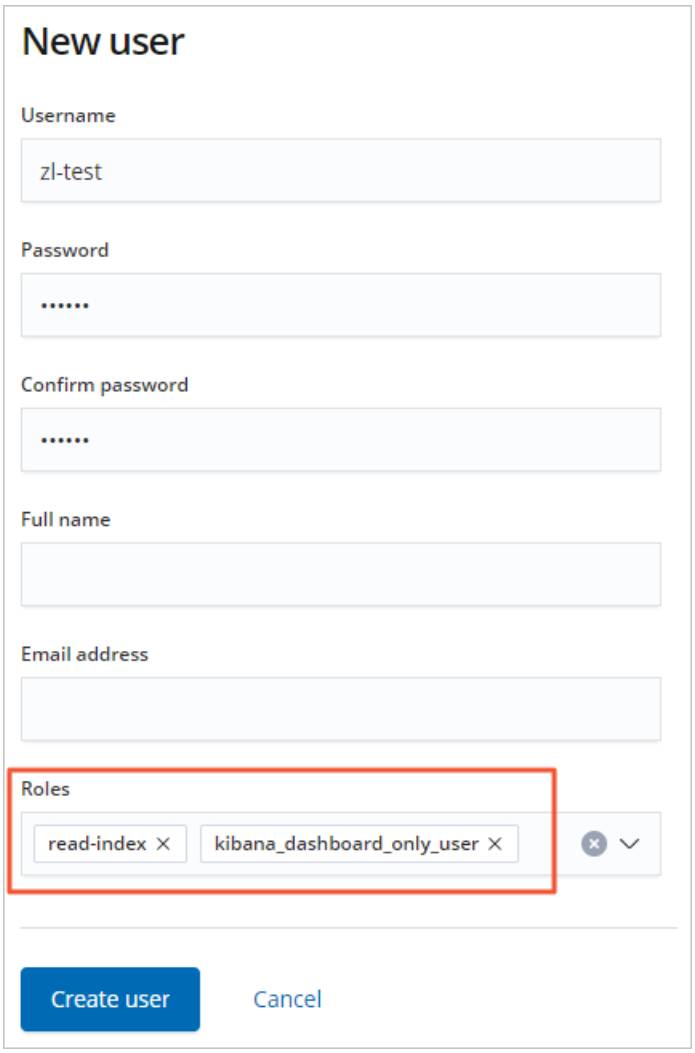
Configure operation permissions on dashboards

- Scenario

  Grant the read-only permissions on a specific index and the permissions to view the dashboards for the index to a common user.

- Role configuration

  When you create a user, assign the read-index and kibana_dashboard_only_user roles to the user.



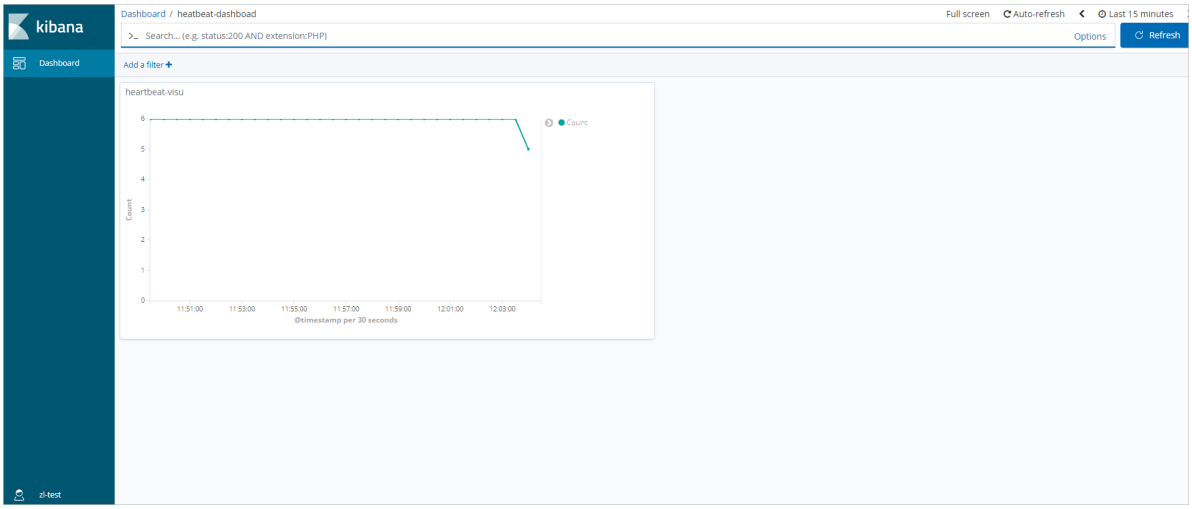
  - read-index: a custom role. You must manually create a custom role. This role has read-only permissions on the specific index.

- kibana_dashboard_only_user: a Kibana built-in role. This role has the permissions to view the dashboards for the index.

  > **Notice**
  >
  > - In Kibana V7.0 and later, the kibana_dashboard_only_user role is deprecated. If you want to view the dashboards for a specific index, you need only to configure the read-only permissions on the index. For more information, see Configure read-only permissions on indexes.
  > - The kibana_dashboard_only_user role can be used with custom roles in various scenarios. If you want to configure the **Dashboards only roles** feature only for a custom role, perform the following steps: In the **Kibana** section of the **Management** page, click **Advanced Settings**. Then, in the **Dashboard** section on the page that appears, set the Dashboards only roles parameter to the custom role. The default value of this parameter is kibana_dashboard_only_user.

- Verification

  Use the common user to log on to the Kibana console and view the dashboards for the specific index.

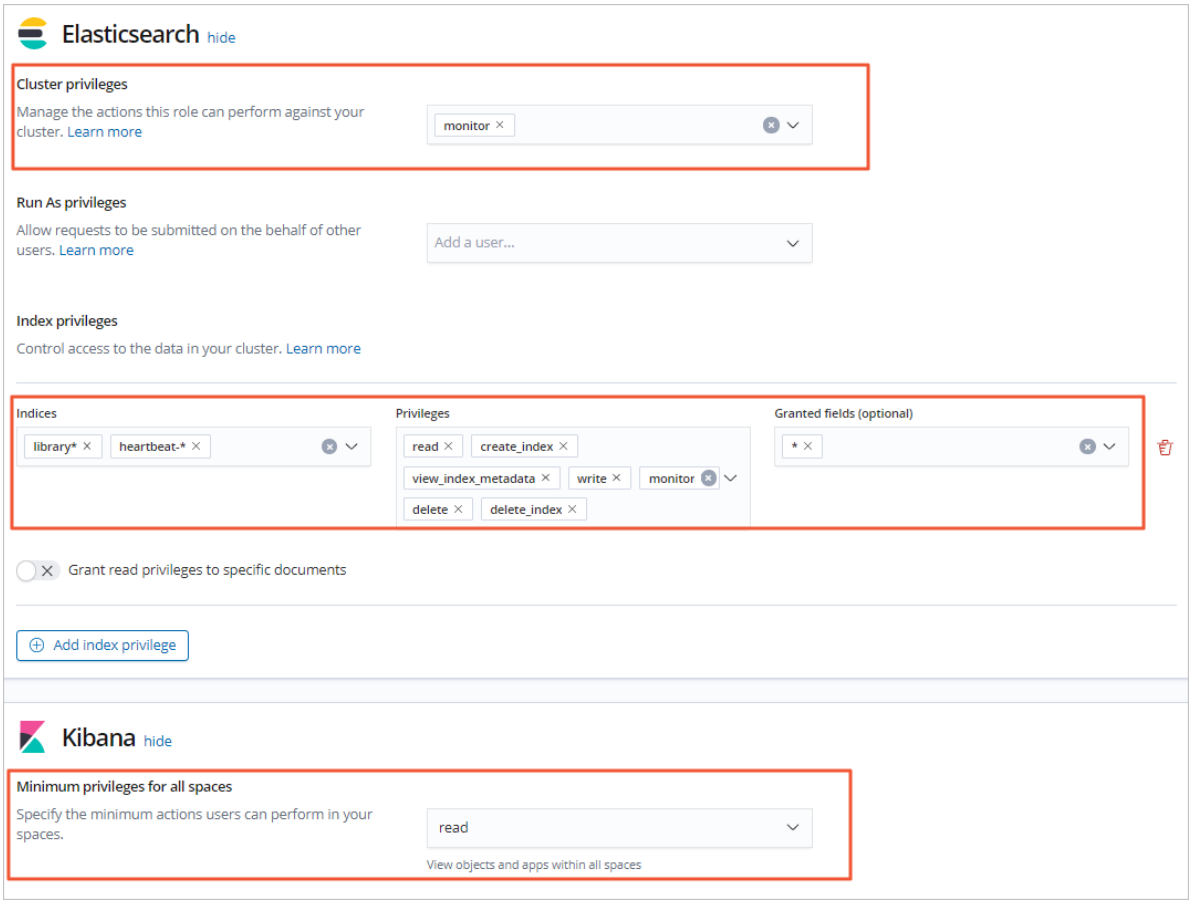


Configure read and write permissions on indexes and read-only permissions on clusters

- Scenario

  Grant the read, write, and delete permissions on specific indexes and the read-only permissions on clusters and Kibana to a common user.

- Role configuration



Permissions

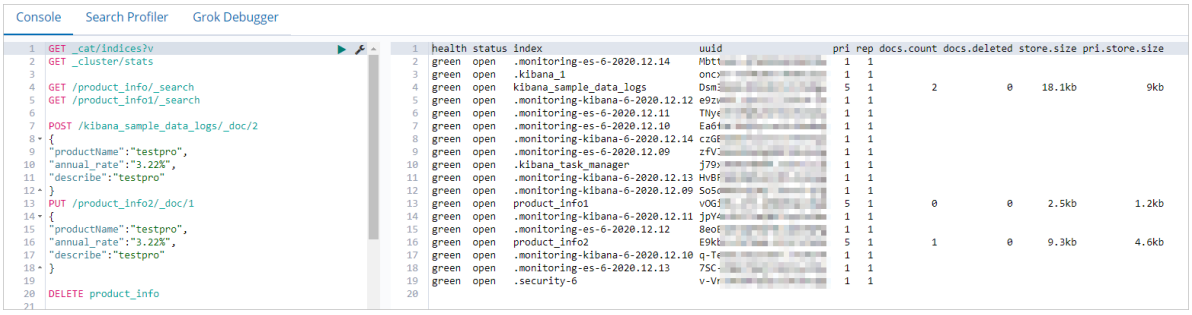
Permission type	Permission key	Permission value	Description
Cluster privileges	**cluster**	monitor	The read-only permissions on clusters, such as the permissions to view the running statuses, health statuses, hot threads, node information, and blocked tasks of clusters.
	indices	heartbeat-*,library*	The names of the indexes. You can specify a full index name, alias, wildcard, or regular expression. For more information, see roles-indices-privileges.
		read	The read-only permissions on the indexes. The read-only permissions include the permissions to call the count, explain, get, mget, scripts, search, and scroll APIs. For more information, see privileges-list-indices.

Permission type	Permission key	Permission value	Description
Index privileges	privileges	create_index	The permission to create indexes. If you specify an alias when you create an index, you must grant the manage permission to the user. **Notice** The alias must meet the matching rules that are defined by the Indices parameter.
		view_index_metadata	The read-only permissions on index metadata. The permissions include the permissions to call the following APIs: aliases, aliases exists, get index, exists, field mappings, mappings, search shards, type exists, validate, warmers, settings, and ilm.
		write	The permission to perform all write operations on documents. The operations include the operations that are performed by calling the index, update, delete, or bulk API and mapping updates. The write permission involves more operation permissions than the create and index permissions.
		monitor	The permission to monitor all operations. The operations include the operations that are performed by calling the index recovery, segments info, index stats, or status API.
		delete	The permission to delete documents.
		delete_index	The permission to delete indexes.
	granted fields	*	The fields on which you want to grant permissions. The value * indicates all fields.

Permission type	Permission key	Permission value	Description
Kibana privileges	privileges	read	The read-only permissions on Kibana. The permissions are granted to all spaces. Default value: none. This value indicates that no spaces are authorized to access Kibana. **Notice** Versions earlier than Kibana V7.0 support only base privileges. Kibana V7.0 and later support base privileges and feature privileges. After you assign a base privilege to a role, the role has access permissions on all Kibana spaces. After you assign a feature privilege to a role, the role has access permissions only on a specific feature. To assign a feature privilege, you must specify a Kibana space.

- Verification

  Use the common user to log on to the Kibana console and run the following commands. The system returns results as expected.



  - View the details about indexes in a cluster

    ```
    GET /_cat/indices?v
    ```

  - View the status of a cluster

    ```
    GET /_cluster/stats
    ```

  - Query data in the product_info index

    ```
    GET /product_info/_search
    ```

  - Query data in the product_info1 index

    ```
    GET /product_info1/_search
    ```

- Use a POST request to write data to the kibana_sample_data_logs index

```
POST /kibana_sample_data_logs/_doc/2
{
    "productName": "testpro",
    "annual_rate": "3.22%",
    "describe": "testpro"
}
```

- Use a PUT request to write data to the product_info2 index

```
PUT /product_info2/_doc/1
{
    "productName": "testpro",
    "annual_rate": "3.22%",
    "describe": "testpro"
}
```

- Delete the product_info index

```
DELETE product_info
```

9.3.4. Configure AD user authentication

Alibaba Cloud Elasticsearch allows you to configure Active Directory (AD) user authentication for your Elasticsearch cluster. This way, users in an AD domain that are assigned Elasticsearch roles can be used to access the cluster. This topic describes how to configure AD user authentication for an Alibaba Cloud Elasticsearch cluster.

Prerequisites

- An Alibaba Cloud Elasticsearch cluster is created.

  For more information, see Create an Alibaba Cloud Elasticsearch cluster. In this example, an Elasticsearch V7.10 cluster is created.

- If your Elasticsearch cluster is created in October 2020 or later, the cluster is deployed in the new network architecture, and the following operations must be performed:

  i.

  ii. Configure a Classic Load Balancer (CLB) instance. For more information, see Step 2: Configure the CLB instance.

  iii.

  iv.

  v.

- An AD domain is created and configured on an Elastic Compute Service (ECS) instance that runs the Windows operating system and resides in the same virtual private cloud (VPC) as the Elasticsearch cluster. In this example, the Windows Server 2012 operating system is used. In addition, data is prepared.

  In this example, the ccy1 user and the ccy.com root domain are used.

Limits

-
- Elasticsearch clusters that are created in October 2020 or later are deployed in the new network architecture. If you want to use AD user authentication for such a cluster, you must first use the PrivateLink service to establish private connections between VPCs. For more information, see Configure a private connection for an Elasticsearch cluster. If you want to connect such a cluster to the Internet, configure an NGINX proxy to forward requests.
- For Elasticsearch clusters that are deployed in the original network architecture, only single-zone clusters support AD user authentication. For Elasticsearch clusters that are deployed in the new network architecture, both single-zone clusters and multi-zone clusters support AD user authentication.

Procedure

1. Step 1: Configure AD user authentication
2. Step 2: Map the user to a role
3. Step 3: Verify the result

Step 1: Configure AD user authentication

An Elasticsearch cluster uses its security features to communicate with the AD domain and authenticate users. The security features communicate with the AD domain based on Lightweight Directory Access Protocol (LDAP). An AD domain is similar to an LDAP domain. Like an LDAP directory, an AD domain stores users and groups in a hierarchical manner. An AD domain authenticates a user by sending an LDAP bind request. After the user passes the authentication, the AD domain searches for the entry of the user in the AD domain. After the AD domain finds the entry, the AD domain retrieves the group membership of the user from the tokenGroups attribute of the entry. For more information, see Configuring an Active Directory realm.

If the version of your Elasticsearch cluster is V6.X, add the following configurations to the YML configuration file of your cluster to configure AD user authentication. For more information, see Configure the YML file. If the version of your Elasticsearch cluster is V7.X, submit a ticket to contact Alibaba Cloud technical support to configure related settings for your cluster.

```
xpack.security.authc.realms.active_directory.my_ad.order: 0
xpack.security.authc.realms.active_directory.my_ad.domain_name: ccy.com
xpack.security.authc.realms.active_directory.my_ad.url: ldap://ep-bp1i321219*********-cn-ha
ngzhou-h.epsrv-bp15571d5ps*********.cn-hangzhou.privatelink.aliyuncs.com:389
xpack.security.authc.realms.active_directory.my_ad.bind_dn: ccy1@ccy.com
xpack.security.authc.realms.active_directory.my_ad.secure_bind_password: your_password
```

Parameter	Description
order	The priority of the AD domain. The priority determines the sequence in which the AD domain is checked during user authentication.
domain_name	The name of the root domain.

Parameter	Description
url	The URL and port number that are used to establish a private network connection between the AD domain and the ECS instance. For more information, see Configuring an Active Directory realm. **Notice** If your Elasticsearch cluster is deployed in the new network architecture, you must set this parameter to a value that is in the format of `ldap://<Domain name of the related endpoint>:<Port number>`. In this example, ldap://ep-bp1i321219*********-cn-hangzhou-h.epsrv-bp15571d5ps*********.cn-hangzhou.privatelink.aliyuncs.com:389 is used.
bind_dn	The distinguished number (DN) of the user that is used to perform searches.
secure_bind_password	The password that is used to authenticate the user.

Step 2: Map the user to a role

1. Log on to the Kibana console of the Elasticsearch cluster.

   For more information, see Log on to the Kibana console.

   Note In this example, an Elasticsearch V7.10.0 cluster is used. Operations on clusters of other versions may differ. The actual operations in the console prevail.

2. Go to the homepage of the Kibana console and click **Dev tools** in the upper-right corner.
3. On the **Console** tab, run the following command to map the ccy1 user in the AD domain to the administrator role:

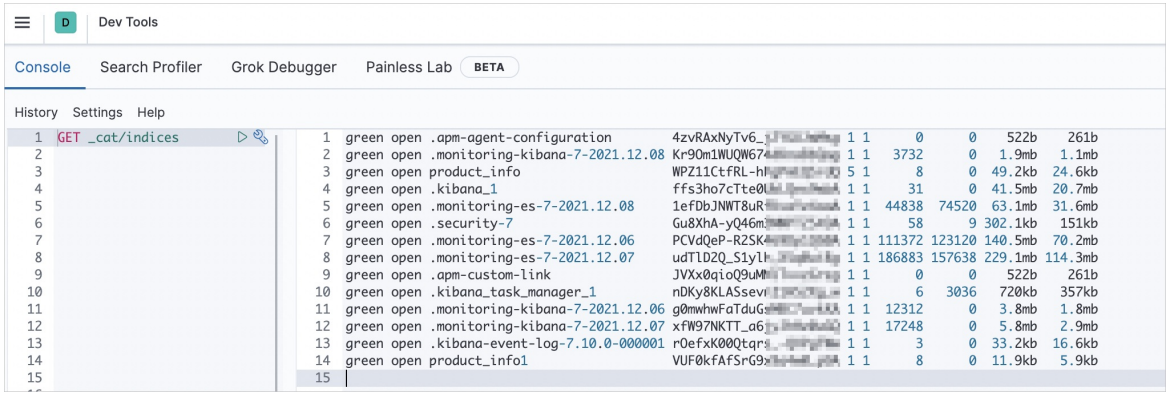
```
PUT /_security/role_mapping/basic_users
{
  "roles": [ "superuser" ],
  "enabled": true,
  "rules": {
    "any": [
      {
        "field": {
          "groups": "cn=ali,dc=ccy,dc=com"
        }
      },
      {
        "field": {
          "dn": "cn=ccy1,cn=ali,dc=ccy,dc=com"
        }
      }
    ]
  }
}
```

Step 3: Verify the result

1. Use the ccy1 user to log on to the Kibana console of the Elasticsearch cluster.
2. Go to the homepage of the Kibana console and click **Dev tools** in the upper-right corner.
3. On the **Console** tab, run the following command to check whether the ccy1 user has permissions to perform the related operation:

```
GET _cat/indices
```

If permissions are granted to the ccy1 user, the result shown in the following figure is returned.



9.4. Cluster security configuration

9.4.1. Use IDaaS to implement SAML SSO to the Kibana console of an Alibaba Cloud Elasticsearch cluster

This topic describes how to use Alibaba Cloud Identity as a Service (IDaaS) to implement Security Assertion Markup Language (SAML) single sign-on (SSO) to the Kibana console of an Alibaba Cloud Elasticsearch cluster. IDaaS serves as the identity provider (IdP), and Kibana serves as the service provider (SP).

Context

Elasticsearch allows you to implement SAML SSO to the Kibana console of your Elasticsearch cluster. Kibana serves as the SAML SP and allows you to configure SAML 2.0 browser-based SSO and SAML 2.0 single logout (SLO). This way, you can use an IdP that complies with SAML 2.0, such as IDaaS or Active Directory Federation Service (AD FS), to access Elasticsearch and Kibana. In this example, IDaaS is used as the IdP.

In this topic, the following terms are involved:

- IDaaS: a centralized platform that provides management over identities, permissions, and applications for enterprises. IDaaS supports various services, such as Employee Identity and Access Management (EIAM) and Customer Identity and Access Management (CIAM).
- SAML: an XML-based open standard that implements SSO across domains. SAML transfers identity information between an IdP and an SP by using security tokens that contain assertions. SAML is a sound identity authentication protocol. It is widely used in public and private clouds worldwide.

- SSO: indicates that you can access multiple mutually trusted application systems with only one logon.

Prerequisites

- An Alibaba Cloud Elasticsearch V7.10 cluster is created, and HTTPS is enabled for the cluster.

  For more information about how to create an Elasticsearch cluster, see Create an Alibaba Cloud Elasticsearch cluster. In this example, an Elasticsearch V7.10 cluster is used. The operations and configurations required for the clusters of other versions may vary. The operations and configurations required in the Elasticsearch console prevail.

  For more information about how to enable HTTPS for an Elasticsearch cluster, see Enable HTTPS.

  > **Notice** You can enable HTTPS only for an Elasticsearch cluster that contains client nodes. Make sure that your Elasticsearch cluster contains client nodes.

- An IDaaS EIAM instance is created.

  > **Note** Elasticsearch supports only HTTP-Redirect binding for SAML authentication requests and does not support other methods such as HTTP-POST binding. You need only to make sure that your computer can access the IdP and SP.

- SAML SSO can be configured only at the backend. You must refer to the operations in this topic to configure the related settings in a test environment and make sure that the test logon to the Kibana console is successful. Then, you can submit a ticket to provide Alibaba Cloud Elasticsearch technical personnel with the configuration information.

  > **Note** This topic consists of the following sections: Configure the IDaaS SAML application (client side) and Create a custom role and configure the SAML information in Elasticsearch (backend). You must manually perform the operations described in Configure the IDaaS SAML application (client side). The operations described in Create a custom role and configure the SAML information in Elasticsearch (backend) must be performed by Alibaba Cloud Elasticsearch technical personnel at the backend. The operations at the backend are described in this topic to help you understand configuration principles and perform a configuration test.

Configure the IDaaS SAML application (client side)

1. Log on to the IDaaS console and click the name of the EIAM instance. On the page that appears, add the SAML application.

   For more information, see Add an application.

2. In the **Add Application (SAML)** panel, find the desired signing key and click **Select** in the Actions column to configure the parameters related to the IdP and SP.

   > **Note** If no signing key is available, you must import or create one.

   You must configure the parameters that are described in the following table. Retain default values for other parameters.

Parameter	Description
Application Name	The name of the SAML application. You can customize the value of this parameter.
IDP IdentityId	The authentication parameter configured in IDaaS. You must configure this parameter for the SP. In this example, set this parameter to IDaaS.
SP Entity ID	The URL of the SP. In this example, the SP is Kibana. Therefore, you must set this parameter to the base URL of Kibana. The base URL must use HTTPS.
SP ACS URL(SSO Location)	The Assertion Consumer Service (ACS) endpoint that receives authentication messages from the IdP. In most cases, the value of this parameter is the URL of Kibana. This ACS endpoint supports only SAML HTTP-POST binding. In mots cases, set this parameter to `${kibana-url}/api/security/v1/saml`. `${kibana-url}` is the base URL of Kibana.
NameIdFormat	The format of the name identifier. Set this parameter to **urn:oasis:names:tc:SAML:2.0:nameid-format:persistent**.
Binding	Set this parameter to the default value **POST**.
Assertion Attribute	The attribute of the assertion. You can customize a name for the attribute, but you must select **Sub-account** as the value.
Account Linking Type	Set this parameter to **Account mapping**.

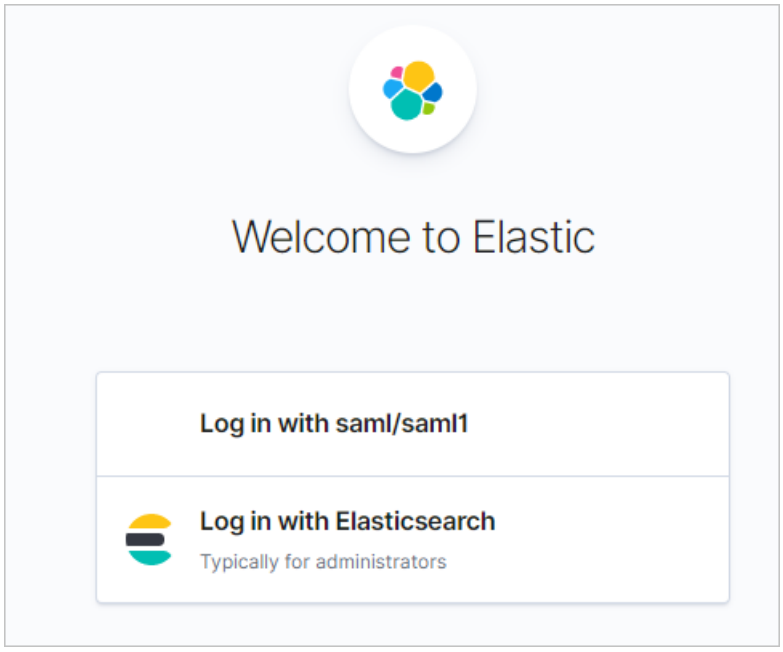
3. Click **Submit**.
4. In the **System Prompt** message, click **Authorize now** to grant permissions to the SAML application.

   > **Notice** Before you grant permissions to the SAML application, make sure that you have synchronized the account information of the application to IDaaS or created an account for the application. For more information, see Accounts.

5. Click Application Authorization in the left-side navigation pane. On the Application Authorization page, click the Authorize Accounts by Application tab. On the Authorize Accounts by Application tab, select the account. Then, click **Save**. In the System Prompt message, click OK to complete the authorization.
6. Export the IDaaS SAML metadata from the added SAML application as a configuration file.
7. Submit a ticket to provide Alibaba Cloud Elasticsearch technical personnel with the metadata configuration file.

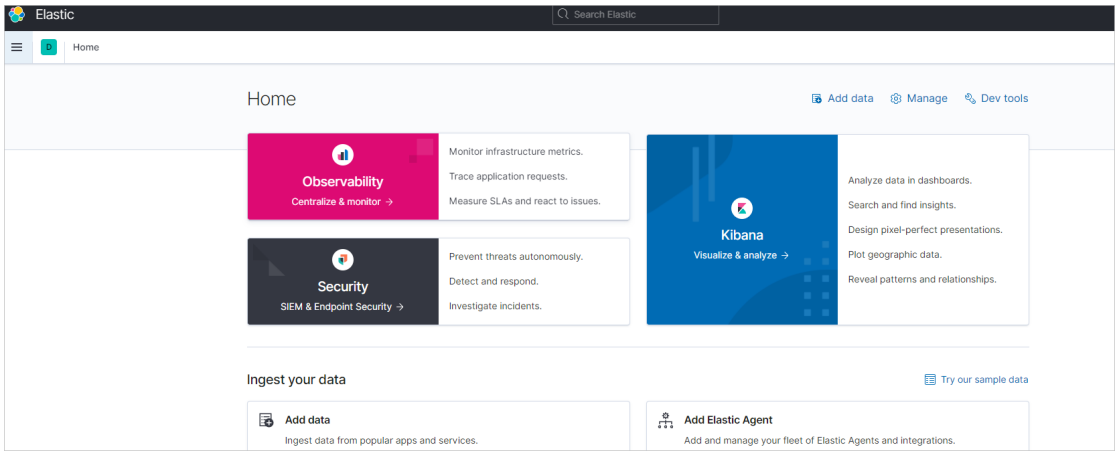
   Then, the technical personnel configure the SAML information in Elasticsearch by following the operations described in Create a custom role and configure the SAML information in Elasticsearch (backend). You can refer to the operations described in this section to perform a test in a self-managed Elasticsearch cluster.
8. After the technical personnel complete the configuration, log on to the Kibana console by using SSO.

i. Refer to the steps described in Log on to the Kibana console to go to the logon page of the Kibana console. Click **Log in with saml/saml1**.



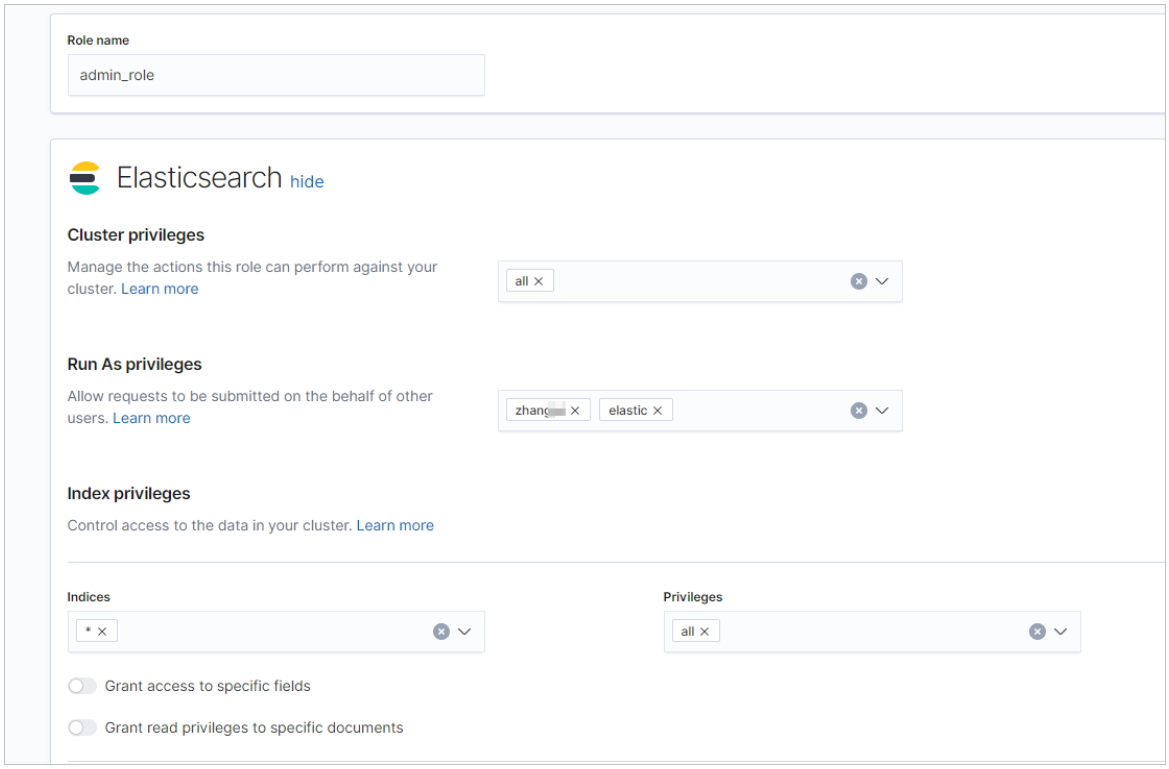
ii. Enter the account that is associated with IDaaS and click **Submit**.

The following figure shows the homepage that appears after your logon.



Create a custom role and configure the SAML information in Elasticsearch (backend)

1. Log on to the Kibana console of the Elasticsearch cluster.
2. Create a custom role.



3. Map the role to the SAML application.

```
PUT /_security/role_mapping/idaas-test
{
  "roles": [ "admin_role" ],
  "enabled": true,
  "rules": {
    "field": { "realm.name": "saml1" }
  }
}
```

> **Note** You must replace the value of the roles parameter with the name of the role created in the preceding step.

4. Upload the metadata configuration file exported in Configure the IDaaS SAML application (client side) to the *config/saml* path of the Elasticsearch cluster.
5. Add SAML information to the YML configuration files of Elasticsearch and Kibana.

> **Notice** The SAML information that you add to the YML configuration files must be consistent with the SAML information configured in Configure the IDaaS SAML application (client side).

   - YML configuration file of Elasticsearch

```
# YML configuration file of Elasticsearch
xpack.security.authc.token.enabled: 'true'
xpack.security.authc.realms.saml.saml1:
  order: 0
  idp.metadata.path: saml/metadata.xml
  idp.entity_id: "https://es-cn-n6xxxxxx1d.elasticsearch.aliyuncs.com/"
  sp.entity_id: "https://es-cn-n6xxxxxx1d.kibana.elasticsearch.aliyuncs.com:5601/"
  sp.acs: "https://es-cn-n6xxxxxx1d.kibana.elasticsearch.aliyuncs.com:5601/api/security/v1/saml"
  attributes.principal: "nameid:persistent"
  attributes.groups: "roles"
```

Parameter	Description
xpack.security.authc.token.enabled	Specifies whether to enable the Token service. You must set this parameter to true to configure SAML SSO. For more information about how to enable the Token service, see saml-enable-token.
xpack.security.authc.realms.saml.saml1	The identity authentication realm. In this example, set this parameter to saml1. For more information about realms, see Realms.
order	The priority of the realm. A small value indicates a high priority.
idp.metadata.path	The path to the metadata file that you saved for the IdP.
idp.entity_id	The identifier of the IdP. The identifier must match the EntityID attribute within the metadata file.
sp.entity_id	The unique identifier of Kibana. This parameter is required if you add Kibana as an SP of your IdP. We recommend that you set this parameter to the base URL of Kibana. **Notice** Make sure that the value of this parameter is consistent with the information of your business environment. If you use a reverse proxy to access Kibana, instead of using a URL, you must specify the endpoint and port number of the reverse proxy in this parameter.
sp.acs	The Assertion Consumer Service (ACS) endpoint that receives authentication messages from the IdP. In most cases, the value of this parameter is the URL of Kibana. This ACS endpoint supports only SAML HTTP-POST binding. In mots cases, set this parameter to `${kibana-url}/api/security/v1/saml`. `${kibana-url}` is the base URL of Kibana.
sp.logout	The URL that Kibana uses to receive the logout information from the IdP. The value format of this parameter is similar to that of the sp.acs parameter. You must set this parameter to `${kibana-url}/logout`. `${kibana-url}` is the base URL of Kibana.

Parameter	Description
attributes.principal	The assertion information. For more information, see Attribute mapping.
attributes.groups	The assertion information. For more information, see Attribute mapping.

- YML configuration file of Kibana

```
# YML configuration file of Kibana
xpack.security.authc.providers:
  saml.saml1:
    order: 0
    realm: "saml1"
  basic.basic1:
    order: 1
    icon: "logoElasticsearch"
    hint: "Typically for administrators"
```

Parameter	Description
xpack.security.authc.providers	The provider of the SAML application. This parameter specifies that SAML SSO is used as the identity authentication method of Kibana.
xpack.security.authc.providers.saml.<provider-name>.realm	The SAML authentication realm. Replace <provider-name> with the realm that you specify in the YML configuration file of Elasticsearch. In this example, saml1 is used.
xpack.security.authc.providers.basic.basic1	After you configure SAML information in the YML configuration file of Kibana, only users who have passed SAML authentication can access Kibana. To log on to the Kibana console as a basic user, you can specify values for the configuration items in basic.basic1. If you test the logon to the Kibana console as a basic user, you may need to use the elastic username and its password to log on to the Elasticsearch cluster, create a role, and then map the role to the SAML application. After you specify values for the configuration items in basic.basic1, the Kibana logon page displays the entry point for you to log on to the Kibana console as a basic user. For more information, see Authentication in Kibana. **Note** If you do not need to log on to the Kibana console as a basic user, you do not need to configure the items in basic.basic1.

9.5. Integrated monitoring

9.5.1. Use Elastic Stack to implement integrated monitoring for containers in Kubernetes

Elastic Stack provides the integrated monitoring feature. This feature allows you to use Kibana to analyze and display the logs, metrics, and application performance monitoring (APM) data of a Container Service for Kubernetes (ACK) cluster in a centralized manner. If you deploy your applications in the pods of an ACK cluster, you can view the logs generated by the pods, event metrics of the hosts and network, and APM data in the Kibana console. This facilitates troubleshooting. This topic describes how to implement integrated monitoring for an ACK cluster.

Prerequisites

- An Alibaba Cloud Elasticsearch V6.8 cluster is created, a whitelist is configured for the cluster, and the Auto Indexing feature is enabled for the cluster.

  For more information, see Create an Alibaba Cloud Elasticsearch cluster, Configure a public or private IP address whitelist for an Elasticsearch cluster, and Configure the YML file.

- An ACK cluster is created, and pods are created in the cluster. In this example, the ACK cluster version 1.18.8-aliyun.1 is used, and each Elastic Compute Service (ECS) instance used for the cluster has 2 vCPUs and 8 GiB of memory.

  For more information, see Create an ACK managed cluster.

- The kubectl client is configured and can be used to access the ACK cluster.

  For more information, see Connect to ACK clusters by using kubectl.

Context

This topic describes how to use Elastic Stack to implement integrated monitoring for an ACK cluster. For more information, see the following sections:

- Use Metricbeat to collect metrics
- Use Filebeat to collect logs
- Use Elastic APM to monitor the performance of applications

For more information about the features of Metricbeat, Filebeat, and Elastic APM, see Infrastructure monitoring, Log monitoring, and Elastic APM.

Use Metricbeat to collect metrics

The following controllers can be used to deploy Metricbeat to the ACK cluster:

- DaemonSet: The DaemonSet controller ensures that each node in the cluster runs one pod. This enables Metricbeat to collect host metrics, system metrics, Docker statistics, and the metrics of all the services that are run on the ACK cluster.
- Deployment: You can use the Deployment controller to deploy a single Metricbeat shipper. The shipper is used to retrieve the unique metrics of the ACK cluster, such as metrics for Kubernetes events and the kube-state-metrics service.

> **Notice**
>
> - In this example, both the DaemonSet and Deployment controllers are used to deploy Metricbeat to the ACK cluster. You can also use only the DaemonSet or Deployment controller to deploy Metricbeat.
> - Metricbeat depends on the monitoring feature provided by the kube-state-metrics service. Before you deploy Metricbeat, you must make sure that kube-state-metrics is deployed. By default, kube-state-metrics is deployed in the arms-prom namespace of a container in an ACK cluster.

1. Use the kubectl client to access the ACK cluster and download the Metricbeat configuration file.

   ```
   curl -L -O https://raw.githubusercontent.com/elastic/beats/6.8/deploy/kubernetes/metricbeat-kubernetes.yaml
   ```

2. Modify the Metricbeat configuration file.

   > **Notice**
   >
   > The official Metricbeat configuration file uses the extensions/v1beta1 API version for DaemonSets and Deployments. This API version is deprecated in ACK V1.18 and later. You must modify the Metricbeat configuration file to use the apps/v1 API version for Kubernetes, DaemonSets, Deployments, and ReplicaSets in ACK V1.18 and later.

   i. Modify the configurations in kind: Deployment and kind: DaemonSet.

- Modify environment variables. The following code provides an example:

```
env:
- name: ELASTICSEARCH_HOST
  value: es-cn-nif23p3mo0065****.elasticsearch.aliyuncs.com
- name: ELASTICSEARCH_PORT
  value: "9200"
- name: ELASTICSEARCH_USERNAME
  value: elastic
- name: ELASTICSEARCH_PASSWORD
  value: ****
- name: KIBANA_HOST
  value: es-cn-nif23p3mo0065****-kibana.internal.elasticsearch.aliyuncs.com
- name: KIBANA_PORT
  value: "5601"
```

Notice By default, Kibana variables are not defined in the downloaded Metricbeat configuration file. You can use the env parameter to pass the variables.

Parameter	Description
ELASTICSEARCH_HOST	The internal endpoint of your Elasticsearch cluster.
ELASTICSEARCH_PORT	The private port of your Elasticsearch cluster.
ELASTICSEARCH_USERNAME	The username of your Elasticsearch cluster. The default value of this parameter is elastic.
ELASTICSEARCH_PASSWORD	The password that corresponds to the elastic username.
KIBANA_HOST	The internal endpoint of Kibana.
KIBANA_PORT	The private port of Kibana.

- Add the spec.selector configurations. The following code provides an example:

```
## kind: DaemonSet
spec:
  selector:
    matchLabels:
       k8s-app: metricbeat
  template:
    metadata:
      labels:
        k8s-app: metricbeat
## kind: Deployment
spec:
  selector:
    matchLabels:
       k8s-app: metricbeat
  template:
    metadata:
      labels:
        k8s-app: metricbeat
```

ii. Configure the Kibana output information in name: metricbeat-daemonset-config and name: metricbeat-deployment-config to use the environment variables that are configured in the configuration file.

```
output.elasticsearch:
      hosts: ['${ELASTICSEARCH_HOST:elasticsearch}:${ELASTICSEARCH_PORT:9200}']
      username: ${ELASTICSEARCH_USERNAME}
      password: ${ELASTICSEARCH_PASSWORD}
setup.kibana:
      host: "https://${KIBANA_HOST}:${KIBANA_PORT}"
setup.dashboards.enabled: true
```

iii. Modify configurations related to metricbeat-daemonset-modules. Define the system metrics monitored by the System module and the metrics that can be obtained by the Kubernetes module. The System module monitors the following metrics: CPU, load, memory, and network.

> **Note** For more information about the configurations and metrics of modules in Metricbeat, see System module and Kubernetes module.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: metricbeat-daemonset-modules
  namespace: kube-system
  labels:
    k8s-app: metricbeat
data:
  system.yml: |-
    - module: system
      period: 10s
      metricsets:
        - cpu
        - load
        - memory
        - network
        - process
        - process_summary
        - core
        - diskio
        - socket
      processes: ['.*']
      process.include_top_n:
        by_cpu: 5      # include top 5 processes by CPU
        by_memory: 5   # include top 5 processes by memory
    - module: system
      period: 1m
      metricsets:
        - filesystem
        - fsstat
      processors:
      - drop_event.when.regexp:
          system.filesystem.mount_point: '^/(sys|cgroup|proc|dev|etc|host|lib)($|/)
'
  kubernetes.yml: |-
    - module: kubernetes
      metricsets:
        - node
        - system
        - pod
        - container
        - volume
      period: 10s
      host: ${NODE_NAME}
      hosts: ["localhost:10255"]
```

iv. Modify configurations related to metricbeat-deployment-modules to obtain the metrics for kube-state-metrics and Kubernetes events.

> **Notice** The Metricbeat service is deployed in the kube-system namespace. The kube-state-metrics service is deployed in the arms-prom namespace by default. The two services belong to different namespaces. Therefore, specify the hosts parameter in the format of kube-state-metrics.<namespace>:8080. If the Metricbeat and kube-state-metrics services are deployed in the same namespace, set the hosts parameter to kube-state-metrics:8080.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: metricbeat-deployment-modules
  namespace: kube-system
  labels:
    k8s-app: metricbeat
data:
  # This module requires `kube-state-metrics` up and running under `kube-system` na
mespace
  kubernetes.yml: |-
    - module: kubernetes
      metricsets:
        - state_node
        - state_deployment
        - state_replicaset
        - state_pod
        - state_container
      period: 10s
      host: ${NODE_NAME}
      hosts: ["kube-state-metrics.arms-prom:8080"]
    # Uncomment this to get k8s events:
    - module: kubernetes
      metricsets:
        - event
```

v. Create a role and assign the role to Metricbeat to implement role-based access control (RBAC). This ensures that Metricbeat can obtain the resource information of the ACK cluster.

```
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: metricbeat
subjects:
- kind: ServiceAccount
  name: metricbeat
  namespace: kube-system
roleRef:
  kind: ClusterRole
  name: metricbeat
  apiGroup: rbac.authorization.k8s.io
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRole
metadata:
  name: metricbeat
  labels:
    k8s-app: metricbeat
rules:
- apiGroups: [""]
  resources:
  - nodes
  - namespaces
  - events
  - pods
  verbs: ["get", "list", "watch"]
- apiGroups: ["extensions"]
  resources:
  - replicasets
  verbs: ["get", "list", "watch"]
- apiGroups: ["apps"]
  resources:
  - statefulsets
  - deployments
  verbs: ["get", "list", "watch"]
- apiGroups:
  - ""
  resources:
  - nodes/stats
  verbs:
  - get
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: metricbeat
  namespace: kube-system
  labels:
    k8s-app: metricbeat
---
```

3. Deploy Metricbeat in the ACK cluster and view the resources of the cluster.

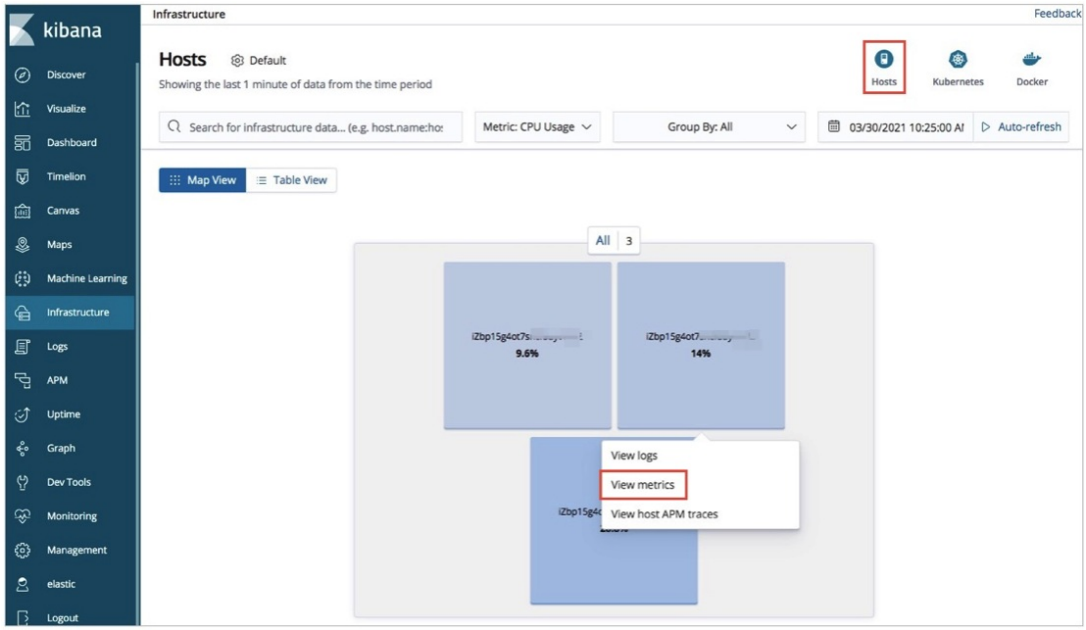
Use the kubectl client to run the following commands:

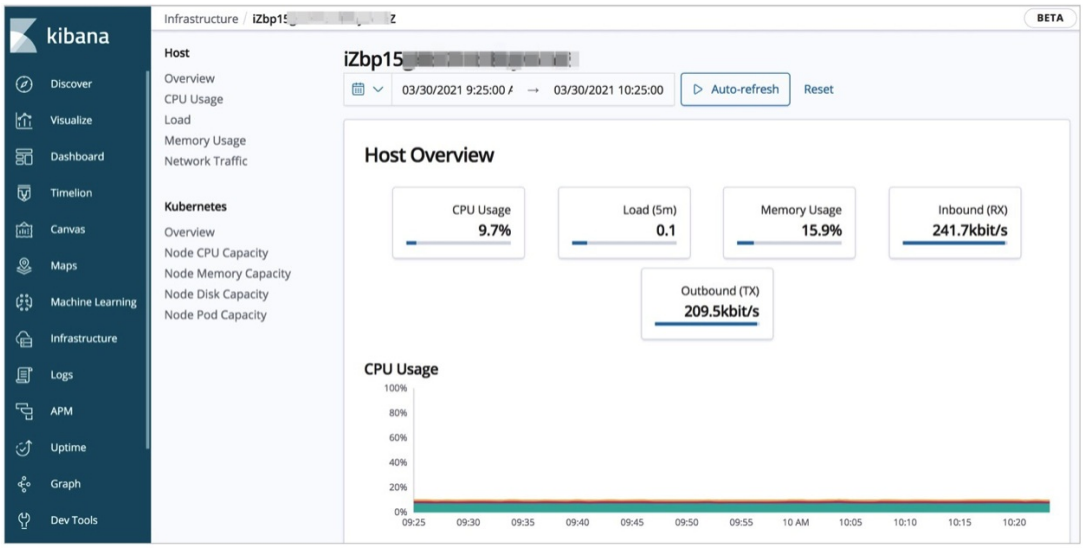
```
kubectl apply -f metricbeat-kubernetes.yaml
kubectl get pods -n kube-system
```

Notice You must make sure that the resources in the pods are running. Otherwise, data may not be displayed in the Kibana console.

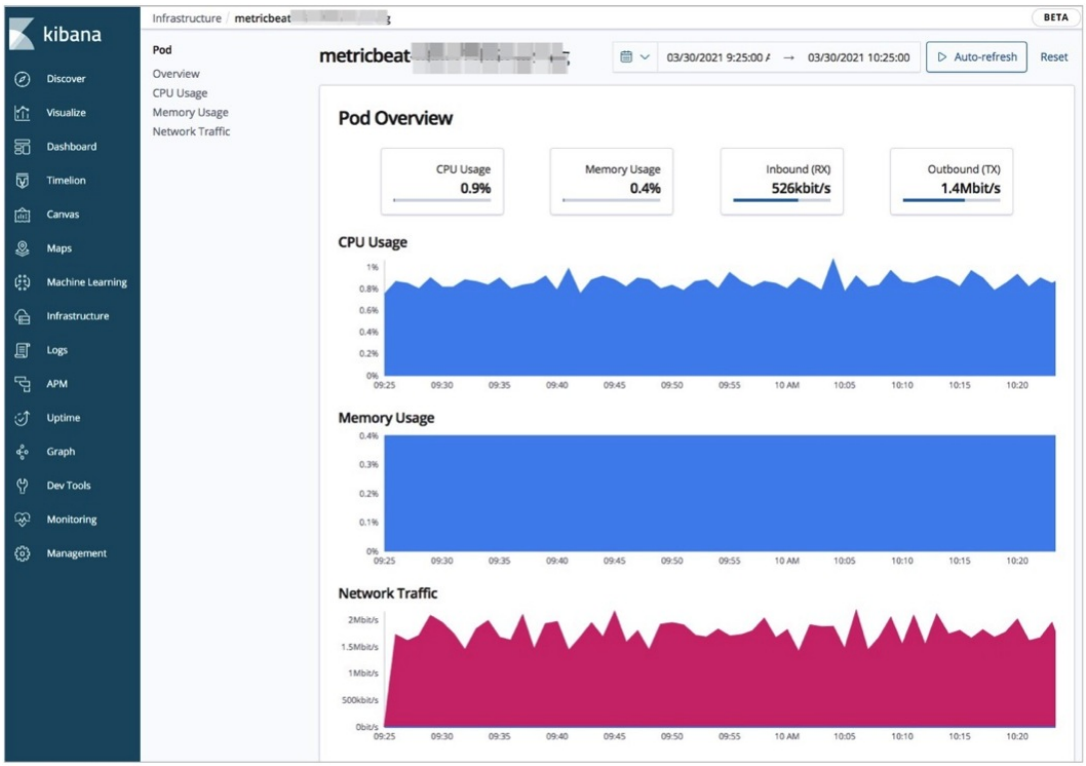
4. View the monitored data in the Kibana console.
   i. Log on to the Kibana console of your Elasticsearch cluster.

      For more information, see Log on to the Kibana console.
   ii. In the left-side navigation pane, click **Infrastructure**.
   iii. View the statistics on the metrics of the hosts and pods in the ACK cluster.
      - View the statistics on the metrics of the hosts: On the Infrastructure page, click **Hosts** in the upper-right corner. On the **Map View** tab, click a host and select **View metrics**. Then, you can view the statistics on the CPU, load, and memory metrics.


kibana
Infrastructure
Feedback
Hosts
Default
Showing the last 1 minute of data from the time period
Hosts
Kubernetes
Docker
Search for infrastructure data... (e.g. host.name:ho
Metric: CPU Usage
Group By: All
03/30/2021 10:25:00 Al
Auto-refresh
Map View
Table View
All 3
9.6%
14%
View logs
View metrics
View host APM traces
Discover
Visualize
Dashboard
Timelion
Canvas
Maps
Machine Learning
Infrastructure
Logs
APM
Uptime
Graph
Dev Tools
Monitoring
Management
elastic
Logout

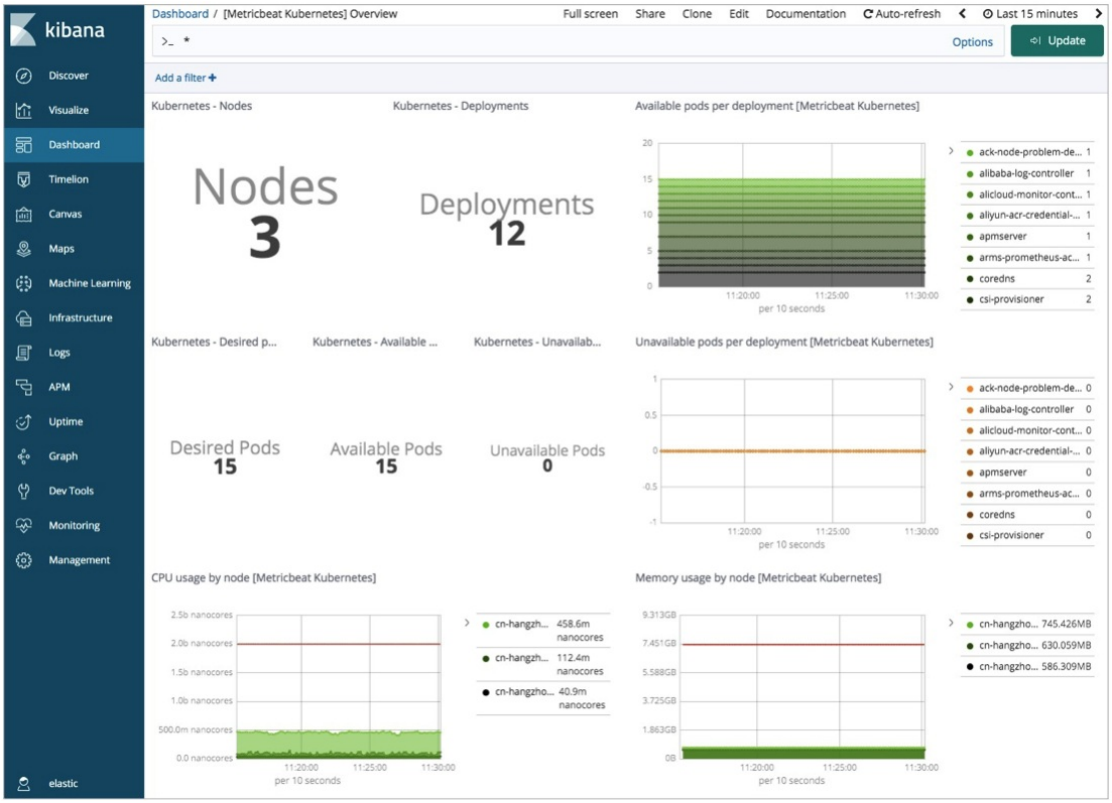

kibana
Infrastructure
BETA
Host
Overview
CPU Usage
Load
Memory Usage
Network Traffic
Kubernetes
Overview
Node CPU Capacity
Node Memory Capacity
Node Disk Capacity
Node Pod Capacity
03/30/2021 9:25:00
03/30/2021 10:25:00
Auto-refresh
Reset
Host Overview
CPU Usage
9.7%
Load (5m)
0.1
Memory Usage
15.9%
Inbound (RX)
241.7kbit/s
Outbound (TX)
209.5kbit/s
CPU Usage
100%
80%
60%
40%
20%
0%
09:25
09:30
09:35
09:40
09:45
09:50
09:55
10 AM
10:05
10:10
10:15
10:20

- View the statistics on the metrics of the pods in the ACK cluster: On the Infrastructure page, click **Kubernetes** in the upper-right corner. On the **Map View** tab, click a pod and select **View metrics**. Then, you can view the statistics on the CPU, memory, and network metrics.



iv. View the overall statistics on the resources of the ACK cluster.

Click **Dashboard** in the left-side navigation pane. On the Dashboards page, click **[Metricbeat Kubernetes] Overview**. Then, you can view the overall statistics on the resources of the ACK cluster.



Use Filebeat to collect logs

In this example, the DaemonSet controller is used to deploy Filebeat. The DaemonSet controller ensures that each node of the ACK cluster runs a pod to collect data. The resources in the Filebeat configuration file are deployed in the kube-system namespace. If you want to change the namespace in which the resources are deployed, you can modify the configuration file.

1. Download the Filebeat configuration file.

   Use the kubectl client to access the ACK cluster and download the Filebeat configuration file.

   ```
   curl -L -O https://raw.githubusercontent.com/elastic/beats/6.8/deploy/kubernetes/filebeat-kubernetes.yaml
   ```

2. Modify the Filebeat configuration file.

i. Modify the configurations in kind: DaemonSet.

```
env:
- name: ELASTICSEARCH_HOST
  value: es-cn-nif23p3mo0065****.elasticsearch.aliyuncs.com
- name: ELASTICSEARCH_PORT
  value: "9200"
- name: ELASTICSEARCH_USERNAME
  value: elastic
- name: ELASTICSEARCH_PASSWORD
  value: ****
- name: KIBANA_HOST
  value: es-cn-nif23p3mo0065****-kibana.internal.elasticsearch.aliyuncs.com
- name: KIBANA_PORT
  value: "5601"
- name: NODE_NAME
  valueFrom:
     fieldRef:
        fieldPath: spec.nodeName
```

Parameter	Description
ELASTICSEARCH_HOST	The internal endpoint of your Elasticsearch cluster.
ELASTICSEARCH_PORT	The private port of your Elasticsearch cluster.
ELASTICSEARCH_USERNAME	The username of your Elasticsearch cluster. The default value of this parameter is elastic.
ELASTICSEARCH_PASSWORD	The password that corresponds to the elastic username.
KIBANA_HOST	The internal endpoint of Kibana.
KIBANA_PORT	The private port of Kibana.
NODE_NAME	The hosts in the ACK cluster.

ii. Change the ConfigMap configurations in name: filebeat-config and configure the Kibana output information to use the environment variables configured in the configuration file.

```
output.elasticsearch:
      hosts: ['${ELASTICSEARCH_HOST:elasticsearch}:${ELASTICSEARCH_PORT:9200}']
      username: ${ELASTICSEARCH_USERNAME}
      password: ${ELASTICSEARCH_PASSWORD}
setup.kibana:
      host: "https://${KIBANA_HOST}:${KIBANA_PORT}"
```

iii. Configure Filebeat to collect logs from the containers in the ACK cluster.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: filebeat-inputs
  namespace: kube-system
  labels:
    k8s-app: filebeat
data:
  kubernetes.yml: |-
    - type: docker
      containers.ids:
      - "*"
      processors:
        - add_kubernetes_metadata:
            host: ${NODE_NAME}
            in_cluster: true
---
```

3. Deploy Filebeat in the ACK cluster and view the resources of the cluster.

   Use the kubectl client to run the following commands:

```
kubectl apply -f filebeat-kubernetes.yaml
kubectl get pods -n kube-system
```

> **Notice** You must make sure that the resources in the pods are running. Otherwise, data may not be displayed in the Kibana console.

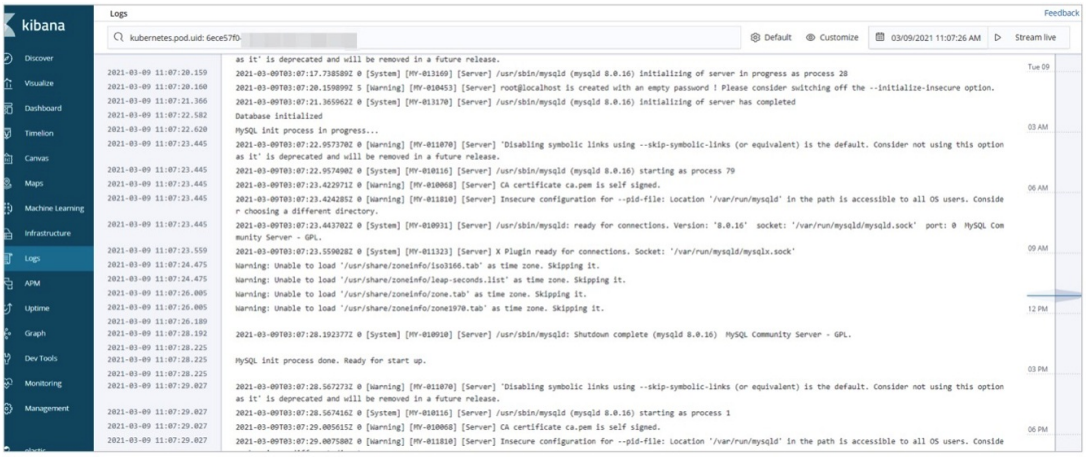
4. View real-time logs in the Kibana console.

   i. Log on to the Kibana console of your Elasticsearch cluster.

   For more information, see Log on to the Kibana console.

ii. View the logs of the hosts and pods in the ACK cluster.

- View the logs of the hosts: On the Infrastructure page, click **Hosts** in the upper-right corner. On the **Map View** tab, click a host and select **View logs**. Then, you can view the real-time logs of the host.
- View the logs of the pods in the ACK cluster: On the Infrastructure page, click **Kubernetes** in the upper-right corner. On the **Map View** tab, click a pod and select **View logs**. Then, you can view the real-time logs of the pod.



Use Elastic APM to monitor the performance of applications

Elastic APM is an application performance monitoring system built on Elastic Stack. Elastic APM allows you to monitor software services and applications in real time. To implement monitoring and facilitate troubleshooting, Elastic APM collects detailed performance information about response time for requests, database queries, calls to caches, and external HTTP requests. Elastic APM also automatically collects unhandled errors and exceptions. Errors are grouped based on the stacktrace. This helps you identify new errors and understand the number of times specific errors occur.

For more information about Elastic APM, see Elastic APM Overview.

1. Deploy APM Server to a container.

   In this example, APM Server is deployed in the ACK cluster. You can use a ConfigMap controller to define the apm-server.yml file and start APM Server by initializing the pods. Then, you can use a Service object to implement service self-discovery and load balancing.

   i. Configure the apm-server-yml file.

   The file contains the following code:

```
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: apm-deployment-config
  namespace: kube-system
  labels:
    k8s-app: apmserver
data:
  apm-server.yml: |-
      apm-server.host: "0.0.0.0:8200"
      output.elasticsearch:
         hosts: ['${ELASTICSEARCH_HOST:elasticsearch}:${ELASTICSEARCH_PORT:9200}']
```

```
        username: ${ELASTICSEARCH_USERNAME}
        password: ${ELASTICSEARCH_PASSWORD}
    setup.kibana:
        host: "https://${KIBANA_HOST}:${KIBANA_PORT}"
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: apmserver
  namespace: kube-system
  labels:
    k8s-app: apmserver
spec:
  selector:
    matchLabels:
      k8s-app: apmserver
  template:
    metadata:
      labels:
        k8s-app: apmserver
    spec:
      serviceAccountName: apmserver
      hostNetwork: true
      dnsPolicy: ClusterFirstWithHostNet
      containers:
      - name: apmserver
        image: docker.elastic.co/apm/apm-server:6.8.14
        args: [
          "-c", "/etc/apm-server.yml",
          "-e",
        ]
        env:
        - name: ELASTICSEARCH_HOST
          value: es-cn-oew20i5h90006****.elasticsearch.aliyuncs.com
        - name: ELASTICSEARCH_PORT
          value: "9200"
        - name: ELASTICSEARCH_USERNAME
          value: elastic
        - name: ELASTICSEARCH_PASSWORD
          value: ****
        - name: KIBANA_HOST
          value: es-cn-oew20i5h90006****-kibana.internal.elasticsearch.aliyuncs.com
        - name: KIBANA_PORT
          value: "5601"
        - name: NODE_NAME
          valueFrom:
            fieldRef:
              fieldPath: spec.nodeName
        securityContext:
          runAsUser: 0
        resources:
          limits:
            memory: 50Mi
          requests:
```

```
            cpu: 20m
            memory: 30Mi
        volumeMounts:
        - name: config
          mountPath: /etc/apm-server.yml
          readOnly: true
          subPath: apm-server.yml
      volumes:
      - name: config
        configMap:
          defaultMode: 0600
          name: apm-deployment-config
---
apiVersion: v1
kind: Service
metadata:
  name: apmserver
  namespace: kube-system
  labels:
    k8s-app: apmserver
spec:
  clusterIP: None
  ports:
  - name: http-metrics
    port: 8200
    targetPort: 8200
  selector:
    k8s-app: apmserver
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: apmserver
  namespace: kube-system
  labels:
    k8s-app: apmserver
---
```

Notice

- When you use the Deployment controller to deploy APM Server, the docker.elastic.co/apm/apm-server:6.8.14 image is used to deploy pods. The version of the image must be consistent with that of the Elasticsearch cluster.
- Port 8200 is exposed to the ACK cluster by using the Service object. This ensures that APM agents can communicate with APM Server.

Parameter	Description
ELASTICSEARCH_HOST	The internal endpoint of your Elasticsearch cluster.
ELASTICSEARCH_PORT	The private port of your Elasticsearch cluster.

Parameter	Description
ELASTICSEARCH_USERNAME	The username of your Elasticsearch cluster. The default value of this parameter is elastic.
ELASTICSEARCH_PASSWORD	The password that corresponds to the elastic username.
KIBANA_HOST	The internal endpoint of Kibana.
KIBANA_PORT	The private port of Kibana.
NODE_NAME	The hosts in the ACK cluster.

ii. Deploy APM Server to a container and view the status of APM Server.

Use the kubectl client to run the following commands:

```
kubectl apply -f apm-server.yml
kubectl get pods -n kube-system
```

Notice You must make sure that the resources in the pods are running. Otherwise, data may not be displayed in the Kibana console.

2. Configure APM agents.

In this example, Spring Boot is used to create a simple Spring application and compress the application into a JAR package. Then, the JAR package and the latest Java agent that is downloaded from Maven Central Repository are uploaded to APM Server. For more information, see Spring Boot and Maven Central.

i. Log on to a node of the ACK cluster and create a Dockerfile named myapply in the working directory.

The Dockerfile must contain the following information:

```
FROM frolvlad/alpine-oraclejdk8
MAINTAINER peterwanghao.com
VOLUME /tmp
ADD spring-boot-0.0.1-SNAPSHOT.jar spring-boot-0.0.1-SNAPSHOT.jar
ADD elastic-apm-agent-1.21.0.jar elastic-apm-agent-1.21.0.jar
EXPOSE 8080
ENTRYPOINT ["java","-javaagent:/elastic-apm-agent-1.21.0.jar","-Delastic.apm.service_name=my-application","-Delastic.apm.server_url=http://apmserver:8200","-Delastic.apm.application_packages=com.example","-jar","/spring-boot-0.0.1-SNAPSHOT.jar"]
```

The parameters and Java commands that are used to start the pods are defined in ENTRYPOINT. The following table describes these parameters.

Parameter	Description
-javaagent	The JAR package of the APM agents.
-Delastic.apm.service_name	The name of the Service object. The name can contain letters, digits, hyphens (-), underscores (_), and spaces.
-Delastic.apm.server_url	The URL of APM Server. *http://apmserver:8200* is specified in the apm-server.yml file.
-Delastic.apm.application_packages	The basic software package of the application.
-jar	The JAR package of the application.

ii. Use the `docker build` command and the Dockerfile myapply to build an image.

Run the following command in the current path:

```
docker build -t myapply .
```

iii. Load the built image to other containers.

iv. Configure the deployment file for the pods. The file is named my-application.yaml.

The file contains the following code:

```
---
apiVersion: v1
kind: Pod
metadata:
  name: my-apply
  namespace: kube-system
  labels:
    app: my-apply
spec:
  containers:
    - name: my-apply
      image: myapply:latest
      ports:
        - containerPort: 8080
      imagePullPolicy: Never
---
apiVersion: v1
kind: Service
metadata:
  name: my-apply
  namespace: kube-system
  labels:
    app: my-apply
spec:
  type: NodePort
  ports:
  - name: http-metrics
    port: 8080
    nodePort: 30000
  selector:
    app: my-apply
---
```

> **Note** image specifies the built image.

v. Use the kubectl client to run the following command to deploy the pods:

```
kubectl apply -f my-application.yaml
```

vi. Use a curl command to access the host over port 30000 after all resources in the pods are running.

Run the following curl command:

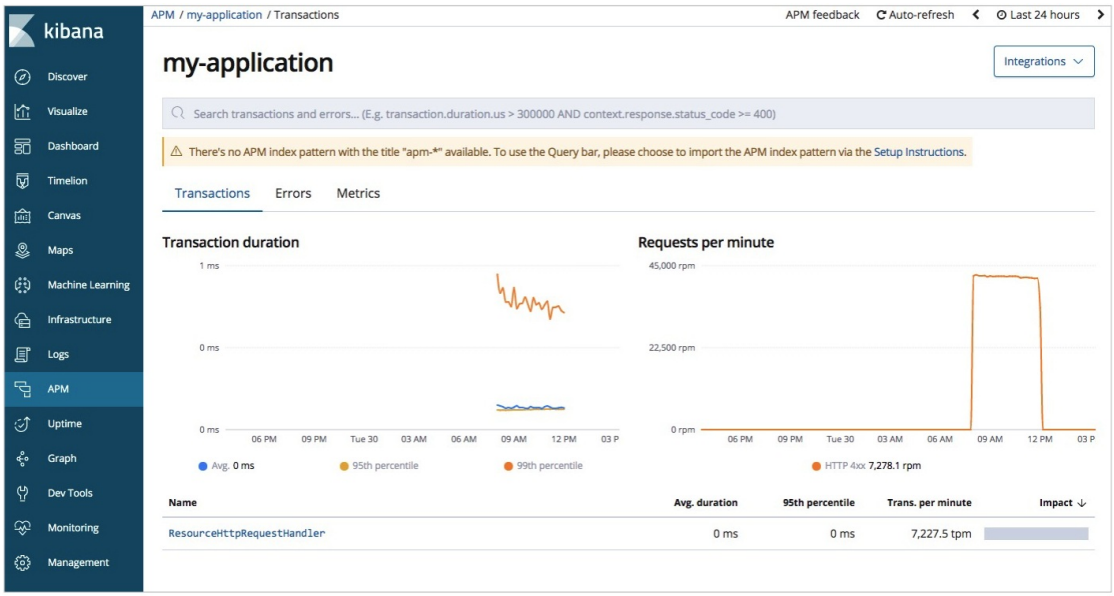
```
curl http://10.7.XX.XX:30000
```

> **Note** 10.7.XX.XX specifies the IP address of the node in the ACK cluster.

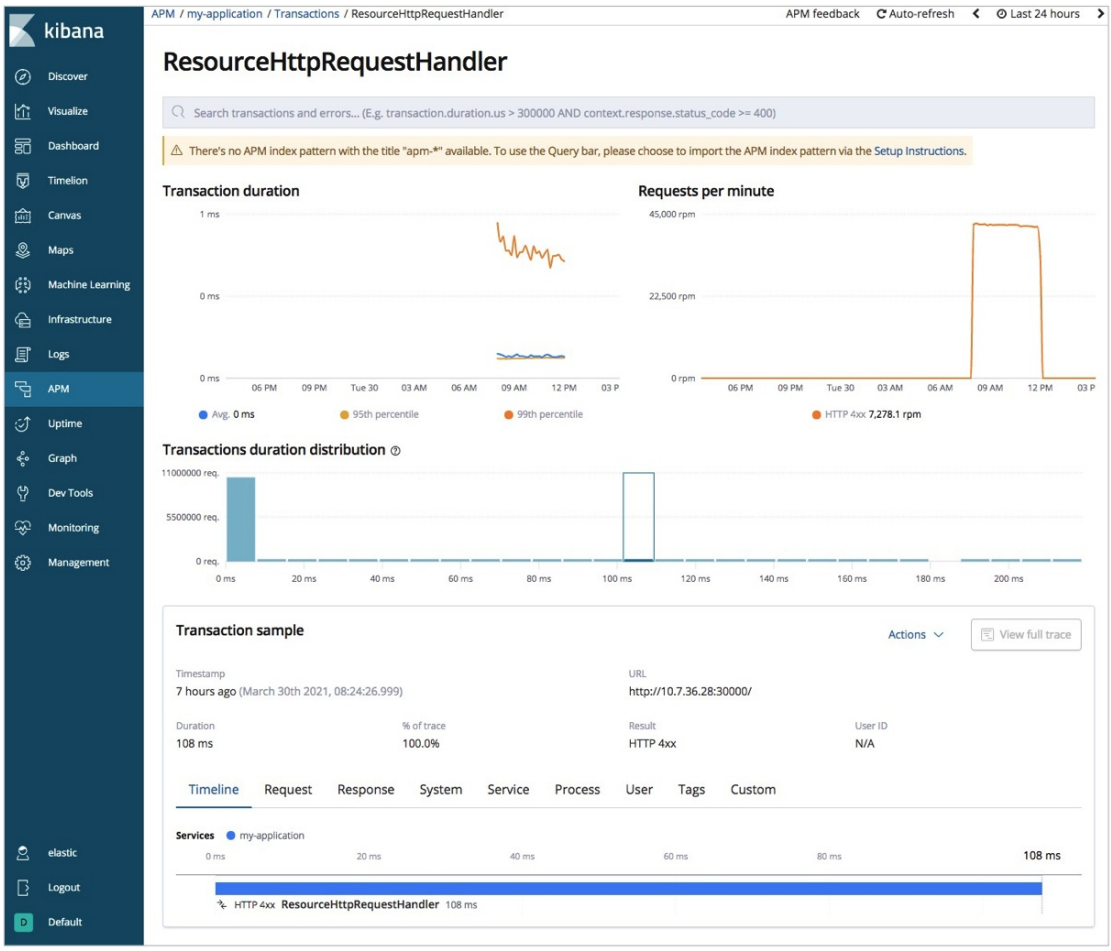
If the access to the host is successful, APM agents are deployed.

3. View the monitoring data obtained by Elastic APM in the Kibana console.
   i. Log on to the Kibana console of your Elasticsearch cluster.

      For more information, see Log on to the Kibana console.
   ii. Click **APM** in the left-side navigation pane.
   iii. Find the application whose performance is monitored and click the application name. Then, you can view the overall performance statistics of the application. In this example, my-application is used.

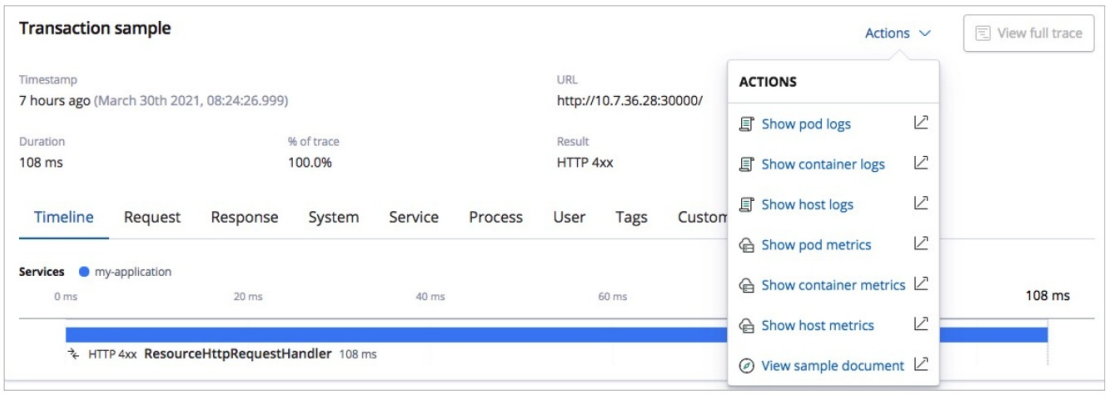


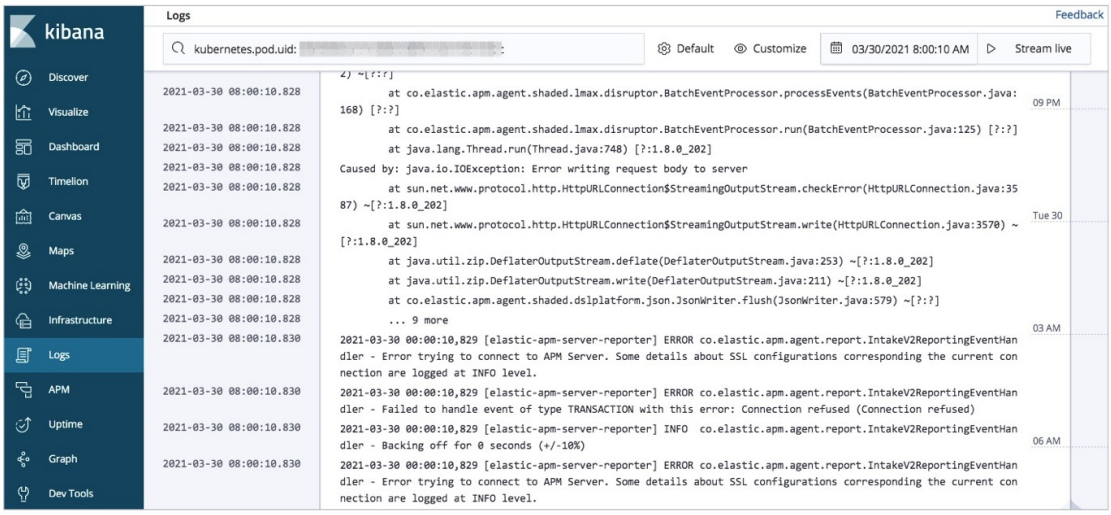
   iv. Click the interface that is requested. Then, you can view the detailed request information.


kibana
Discover
Visualize
Dashboard
Timelion
Canvas
Maps
Machine Learning
Infrastructure
Logs
APM
Uptime
Graph
Dev Tools
Monitoring
Management
elastic
Logout
Default
APM / my-application / Transactions / ResourceHttpRequestHandler
APM feedback
Auto-refresh
Last 24 hours
ResourceHttpRequestHandler
Search transactions and errors... (E.g. transaction.duration.us > 300000 AND context.response.status_code >= 400)
There's no APM index pattern with the title "apm-*" available. To use the Query bar, please choose to import the APM index pattern via the Setup Instructions.
Transaction duration
Requests per minute
Avg. 0 ms
95th percentile
99th percentile
HTTP 4xx 7,278.1 rpm
Transactions duration distribution
Transaction sample
Actions
View full trace
Timestamp
7 hours ago (March 30th 2021, 08:24:26.999)
URL
http://10.7.36.28:30000/
Duration
108 ms
% of trace
100.0%
Result
HTTP 4xx
User ID
N/A
Timeline
Request
Response
System
Service
Process
User
Tags
Custom
Services
my-application
108 ms
HTTP 4xx ResourceHttpRequestHandler 108 ms

v. View the statistics on the logs and metrics of the hosts and pods.

Click **Actions** and select **Show pod logs** or **Show pod metrics**. Then, you can view the statistics on the logs or metrics.





FAQ

- Problem description: The resources.requests parameter is set to a large value in the configuration file of the ACK cluster. As a result, pods fail to be started.

  Solution: Change the value of the resource.requests parameter. The resources.requests parameter must be specified in the configuration files of Metricbeat, Filebeat, and Elastic APM. We recommend that you set this parameter to an appropriate value based on the specifications of the ACK cluster.

- Problem description: Errors keep occurring when I deploy Metricbeat, Filebeat, and Elastic APM. The error message is similar to `no matches for kind "DaemonSet" in version "extensions/v1beat1"`.

  Solution: Use the apps/v1 API version. This is because that the official configuration file uses the extensions/v1beta1 API version for DaemonSets and Deployments. However, in ACK V1.18 and later, the extensions/v1beta1 API version is deprecated for DaemonSets, Deployments, and ReplicaSets.

9.6. Data management and visualization

9.6.1. Use Terraform to manage Alibaba Cloud Elasticsearch clusters

Terraform allows you to use code to allocate resources such as physical machines. You can use Terraform to write a configuration file to purchase a cloud server or apply for resources such as Alibaba Cloud Elasticsearch and Object Storage Service (OSS). This topic describes how to use Terraform to manage your Alibaba Cloud Elasticsearch clusters, such as creating, updating, viewing, or deleting a cluster.

Context

You can install and configure Terraform by using the following methods:

- Install and configure Terraform in the local PC. This method is used in this topic.
- Use Terraform in Cloud Shell.

Install and configure Terraform

1. Download the software package that is suitable for your OS from the official Terraform website.

   In this example, Terraform is installed and configured in a Linux OS. If you do not have a Linux OS, you can purchase an Alibaba Cloud Elastic Compute Service (ECS) instance. For more information, see Step 1: Create an ECS instance.

2. Decompress the package to the */usr/local/bin* directory.

   If you want to decompress the package to another directory, define a global path for the package by using one of the following methods:

   - Linux: See How to define a global path in Linux.
   - Windows: See How to define a global path in Windows.
   - macOS: See How to define a global path in macOS.

3. Run the `terraform` command to verify the path.

   ```
   terraform
   Usage: terraform [-version] [-help] <command> [args]
   ```

```
[root@elastic64 ~]# terraform
Usage: terraform [-version] [-help] <command> [args]

The available commands for execution are listed below.
The most common, useful commands are shown first, followed by
less common or more advanced commands. If you're just getting
started with Terraform, stick with the common commands. For the
other commands, please read the help and docs before usage.

Common commands:
    apply              Builds or changes infrastructure
    console            Interactive console for Terraform interpolations
    destroy            Destroy Terraform-managed infrastructure
    env                Workspace management
    fmt                Rewrites config files to canonical format
    get                Download and install modules for the configuration
    graph              Create a visual graph of Terraform resources
    import             Import existing infrastructure into Terraform
    init               Initialize a Terraform working directory
    output             Read an output from a state file
    plan               Generate and show an execution plan
    providers          Prints a tree of the providers used in the configuration
    refresh            Update local state file against real resources
    show               Inspect Terraform state or plan
    taint              Manually mark a resource for recreation
    untaint            Manually unmark a resource as tainted
    validate           Validates the Terraform files
    version            Prints the Terraform version
    workspace          Workspace management
```

4. Create a Resource Access Management (RAM) user and grant permissions to the user.

   For higher flexibility and security in permission management, we recommend that you create a RAM user and grant the required permissions to the RAM user.

   i. Log on to the RAM console.

   ii. Create a RAM user named Terraform and an AccessKey pair for the user.

   For more information, see Create a RAM user.

   > **Notice** We recommend that you do not use the AccessKey pair of your Alibaba Cloud account to configure Terraform.

   iii. Grant the required permissions to the RAM user.

   In this example, the RAM user Terraform is granted the `AliyunElasticsearchFullAccess` and `AliyunVPCFullAccess` permissions. For more information, see Grant permissions to a RAM user.

5. Create a test directory.

   You must create an independent directory for each Terraform project. For this example, create a test directory named `terraform-test`.

   ```
   mkdir terraform-test
   ```

6. Go to the `terraform-test` directory.

   ```
   cd terraform-test
   ```

7. Create a configuration file and configure identity authentication information.

Terraform reads all the *.tf and *.tfvars files in the directory when it is running. You can write different configurations to these files base on your business requirements. The following table lists the frequently used configuration files.

Configuration file	Description
provider.tf	Used to configure providers.
terraform.tfvars	Used to configure the variables required to configure providers.
varable.tf	Used to configure universal variables.
resource.tf	Used to define resources.
data.tf	Used to define package files.
output.tf	Used to define output files.

For example, when you create the provider.tf file, you can configure identity authentication information in the following format:

```
vim provider.tf
```

```
provider "alicloud" {
    region      = "cn-hangzhou"
    access_key  = "LTA**********NO2"
    secret_key   = "MOk8x0**********wwff"
    }
```

For more information, see alicloud_elasticsearch_instance.

8. Create a folder named plugh in the current directory, download a provider package, and then decompress the package to the plugh folder.

```
mkdir -p
```

9. Initialize the working directory and use `-plugin-dir` to specify the path that is used to store the provider.

```
terraform init -plugin-dir=./plugh/
```

If the `Terraform has been successfully initialized` message is returned, the working directory is initialized.

> **Notice** After you create a working directory and a configuration file for a Terraform project, you must initialize the working directory.

Create an Alibaba Cloud Elasticsearch cluster

1. Create a configuration file named elastic.tf in the test directory.
2. Configure the elastic.tf file to create a multi-zone Elasticsearch V6.7 cluster of the Standard Edition. You can refer to the following script to configure the elastic.tf file:

```
resource "alicloud_elasticsearch_instance" "instance" {
  description          = "testInstanceName"
  instance_charge_type = "PostPaid"
  data_node_amount     = "2"
  data_node_spec       = "elasticsearch.sn2ne.large"
  data_node_disk_size  = "20"
  data_node_disk_type  = "cloud_ssd"
  vswitch_id           = "vsw-bp1f7r0ma00pf9h21****"
  password             = "es_password"
  version              = "6.7_with_X-Pack"
  master_node_spec     = "elasticsearch.sn2ne.large"
  zone_count           = "1"
}
```

The following table describes the parameters supported by providers.

Parameter	Required	Description
`description`	No	The description of the cluster name.
`instance_charge_type`	No	The billing method of the cluster. Valid values: `PrePaid` and `PostPaid`. Default value: PostPaid.
`period`	No	The billing cycle of the cluster. Unit: months. This parameter is valid only when `instance_charge_type` is set to `PrePaid`. Valid values: 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 24, and 36. Default value: 1.
`data_node_amount`	Yes	The number of data nodes in the cluster. Valid values: 2 to 50.
`data_node_spec`	Yes	The specifications of each data node.
`data_node_disk_size`	Yes	The disk space. Different types of disks provide different storage space: ○ `cloud_ssd`: If the disk type is the standard SSD (cloud_ssd), the maximum value of this parameter is 2048, which indicates 2 TiB of storage space. ○ `cloud_efficiency`: If the disk type is the ultra disk (cloud_efficiency), the maximum value of this parameter is 5120, which indicates 5 TiB of storage space. Ultra disks are cost-effective and can be used in scenarios such as logging and analyzing large amounts of data. If you want to specify a size greater than 2,048 GiB for an ultra disk, you can set the value of this parameter only to 2560, 3072, 3584, 4096, 4608, or 5120.

Parameter	Required	Description
`data_node_disk_type`	Yes	The disk type. Valid values: `cloud_ssd` and `cloud_efficiency`.
`vswitch_id`	Yes	The ID of the vSwitch.
`password`	No	The password that is used to access the cluster. It must be 8 to 32 characters in length and can contain letters, digits, and special characters. The following special characters are allowed: `! @ # $ % ^ & * ( ) _ + - =`
`kms_encrypted_password`	No	The encrypted password of Key Management Service (KMS). You do not need to set this parameter if you set `password`. You must set either `password` or `kms_encrypted_password`.
`kms_encryption_context`	No	The KMS encryption context. This parameter is valid only when `kms_encrypted_password` is specified. This parameter is used to decrypt the cluster that is created or updated with `kms_encrypted_password`. For more information, see Encryption context.
`version`	Yes	The version of the cluster. Valid values: `5.5.3_with_X-Pack`, `6.3_with_X-Pack`, and `6.7_with_X-Pack`.
`private_whitelist`	No	The IP address whitelist for access to the cluster over a virtual private cloud (VPC).
`kibana_whitelist`	No	The IP address whitelist for access to the Kibana console.
`master_node_spec`	No	The specifications of each dedicated master node.
`advancedDedicateMaster`	No	Specifies whether to create dedicated master nodes. Default value: false. Valid values: ○ true: Create dedicated master nodes. If the cluster is deployed across zones and dedicated master nodes are enabled for the cluster, you must set this parameter to *true*. ○ false: Do not create dedicated master nodes.
`zone_count`	No	The number of zones. Valid values: 1 to 3. The value of `data_node_amount` must be a multiple of this value.

For more information, see alicloud_elasticsearch_instance.

> **Notice**
>
> - `kms_encrypted_password` and `kms_encryption_context` are available only for provider V1.57.1 or later. `zone_count` is available only for provider V1.44.0 or later.
> - If you want to purchase nodes other than data nodes, call the CreateInstance operation to create an Elasticsearch cluster for which you want to purchase nodes other than data nodes. For more information, see createInstance. For example, if you want to create a multi-zone cluster that contains dedicated master nodes, add `advancedDedicateMaster="true"` to the script.

3. Run the `terraform plan` command to view the operations that will be performed.
If the command is successfully run, the following result is returned:

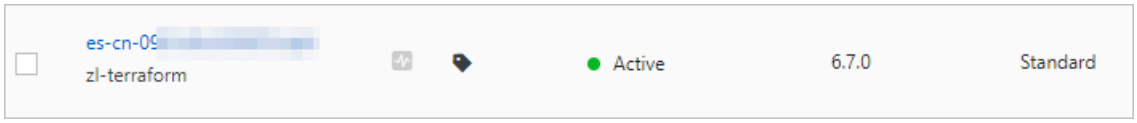
```
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.
------------------------------------------------------------------------
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create
Terraform will perform the following actions:
  # alicloud_elasticsearch_instance.instance will be created
  + resource "alicloud_elasticsearch_instance" "instance" {
      + description          = "testInstanceName"
      + data_node_amount     = 2
      + data_node_disk_size  = 20
      + data_node_disk_type  = "cloud_ssd"
      + data_node_spec       = "elasticsearch.sn2ne.large"
      + domain               = (known after apply)
      + id                   = (known after apply)
      + instance_charge_type = "PostPaid"
      + kibana_domain        = (known after apply)
      + kibana_port          = (known after apply)
      + kibana_whitelist     = (known after apply)
      + master_node_spec     = "elasticsearch.sn2ne.large"
      + password             = (sensitive value)
      + port                 = (known after apply)
      + private_whitelist    = (known after apply)
      + public_whitelist     = (known after apply)
      + status               = (known after apply)
      + version              = "6.7_with_X-Pack"
      + vswitch_id           = "vsw-bp1f7r0ma00pf9h2l****"
      + zone_count           = 1
    }
Plan: 1 to add, 0 to change, 0 to destroy.
------------------------------------------------------------------------
Note: You didn't specify an "-out" parameter to save this plan, so Terraform
can't guarantee that exactly these actions will be performed if
"terraform apply" is subsequently run.
```

4. Run the `terraform apply` command to run the configuration file in the working directory and enter *yes*.

   If the command is successfully run, the following result is returned:

```
Plan: 1 to add, 0 to change, 0 to destroy.
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.
  Enter a value: yes
 alicloud_elasticsearch_instance.instance: Creating...
 alicloud_elasticsearch_instance.instance: Still creating... [10s elapsed]
alicloud_elasticsearch_instance.instance: Still creating... [20s elapsed]
 ..............
 Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

5. Log on to the Elasticsearch console to view the newly created Elasticsearch cluster.



Change cluster configurations

1. Go to the test directory and modify the elastic.tf configuration file.

   For example, change the value of `data_node_disk_size` to `50`.

```
resource "alicloud_elasticsearch_instance" "instance" {
  instance_charge_type = "PostPaid"
  data_node_amount     = "2"
  data_node_spec       = "elasticsearch.sn2ne.large"
  data_node_disk_size  = "50"
  data_node_disk_type  = "cloud_ssd"
  vswitch_id           = "vsw-bp1f7r0ma00pf9h21****"
  password             = "es_password"
  version              = "6.7_with_X-Pack"
  master_node_spec     = "elasticsearch.sn2ne.large"
  zone_count           = "1"
}
```

> **Notice**
>
> - After a cluster is created, you cannot change the value of `version`.
> - You can modify only one configuration item in a request. For example, if you modify both `data_node_spec` and `data_node_disk_size`, the system reports an error.

2. Run the `terraform plan` command to view cluster information.
3. Run the `terraform apply` command. Wait until the configuration upgrade of the cluster is complete.

Import the Elasticsearch cluster

If your Elasticsearch cluster is not created by using Terraform, you can run commands to import the cluster to the state directory of Terraform.

1. Create a file named main.tf in the test directory.

   ```
   vim main.tf
   ```

2. Declare the cluster and specify the storage path of the cluster that you want to import to the state directory.

   ```
   resource "alicloud_elasticsearch_instance" "test" {}
   ```

3. Import the cluster.

   ```
   terraform import alicloud_elasticsearch_instance.test  es-cn-0pp1f1y5g000h****
   ```

   If the command is successfully run, the following result is returned:

   ```
   alicloud_elasticsearch_instance.test: Importing from ID "es-cn-0pp1f1y5g000h****"...
   alicloud_elasticsearch_instance.test: Import prepared!
     Prepared alicloud_elasticsearch_instance for import
   alicloud_elasticsearch_instance.test: Refreshing state... [id=es-cn-0pp1f1y5g000h****]
   Import successful!
   The resources that were imported are shown above. These resources are now in
   your Terraform state and will henceforth be managed by Terraform.
   ```

   > **Note** For more information about how to import and manage existing clusters, see Manage existing cloud resources.

View all the managed clusters

Run the `terraform show` command to view all the managed clusters and their attribute values in the state directory.

```
# alicloud_elasticsearch_instance.instance:
resource "alicloud_elasticsearch_instance" "instance" {
    data_node_amount     = 2
    data_node_disk_size  = 20
    data_node_disk_type  = "cloud_ssd"
    data_node_spec       = "elasticsearch.sn2ne.large"
    domain               = "es-cn-dssf9op81lz4q****.elasticsearch.aliyuncs.com"
    id                   = "es-cn-dssf9op81lz4q****"
    instance_charge_type = "PostPaid"
    kibana_domain        = "es-cn-dssf9op81lz4q****.kibana.elasticsearch.aliyuncs.com"
    kibana_port          = 5601
    kibana_whitelist     = []
    master_node_spec     = "elasticsearch.sn2ne.large"
    password             = (sensitive value)
    port                 = 9200
    private_whitelist    = []
    public_whitelist     = []
    status               = "active"
    version              = "6.7.0_with_X-Pack"
    vswitch_id           = "vsw-bp1f7r0ma00pf9h2l****"
    zone_count           = 1
}
# alicloud_elasticsearch_instance.test:
resource "alicloud_elasticsearch_instance" "test" {
    data_node_amount     = 3
    data_node_disk_size  = 51
    data_node_disk_type  = "cloud_ssd"
    data_node_spec       = "elasticsearch.r5.large"
    domain               = "es-cn-0pp1f1y5g000h****.elasticsearch.aliyuncs.com"
    id                   = "es-cn-0pp1f1y5g000h****"
    instance_charge_type = "PostPaid"
    kibana_domain        = "es-cn-0pp1f1y5g000h****.kibana.elasticsearch.aliyuncs.com"
    kibana_port          = 5601
    kibana_whitelist     = []
    port                 = 9200
    private_whitelist    = []
    public_whitelist     = []
    status               = "active"
    version              = "6.7.0_with_X-Pack"
    vswitch_id           = "vsw-bp1f7r0ma00pf9h2l****"
    zone_count           = 1
    timeouts {}
}
```

Delete a cluster

Warning After a cluster is deleted, it cannot be recovered, and all the data stored on the cluster is deleted.

Go to the test directory, run the `terraform destroy` command, and then enter *yes* to delete the cluster.

```
# terraform destroy
alicloud_elasticsearch_instance.instance: Refreshing state... [id=es-cn-v3x49h5397fau****]
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  - destroy
Terraform will perform the following actions:
  # alicloud_elasticsearch_instance.instance will be destroyed
  - resource "alicloud_elasticsearch_instance" "instance" {
      - data_node_amount     = 2 -> null
      - data_node_disk_size  = 20 -> null
      - data_node_disk_type  = "cloud_ssd" -> null
      - data_node_spec       = "elasticsearch.sn2ne.large" -> null
      - domain               = "es-cn-v3x49h5397fau****.elasticsearch.aliyuncs.com" -> null
      - id                   = "es-cn-v3x49h5397fau****" -> null
      - instance_charge_type = "PostPaid" -> null
      - kibana_domain        = "es-cn-v3x49h5397fau****.kibana.elasticsearch.aliyuncs.com"
-> null
      - kibana_port          = 5601 -> null
      - kibana_whitelist     = [] -> null
      - master_node_spec     = "elasticsearch.sn2ne.large" -> null
      - password             = (sensitive value)
      - port                 = 9200 -> null
      - private_whitelist    = [] -> null
      - public_whitelist     = [] -> null
      - status               = "active" -> null
      - version              = "6.7.0_with_X-Pack" -> null
      - vswitch_id           = "vsw-bp1f7r0ma00pf9h2l****" -> null
      - zone_count           = 1 -> null
    }
Plan: 0 to add, 0 to change, 1 to destroy.
Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.
  Enter a value: yes
alicloud_elasticsearch_instance.instance: Destroying... [id=es-cn-v3x49h5397fau****]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****,
10s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****,
20s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****,
30s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****,
40s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****,
50s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****,
1m0s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****,
1m10s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****,
1m20s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****,
1m30s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****,
1m40s elapsed]
```

```
1m40s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****,
1m50s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****,
2m0s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****,
2m10s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****,
2m20s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****,
2m30s elapsed]
alicloud_elasticsearch_instance.instance: Still destroying... [id=es-cn-v3x49h5397fau****,
2m40s elapsed]
alicloud_elasticsearch_instance.instance: Destruction complete after 10m2s
Destroy complete! Resources: 1 destroyed.
```

9.6.2. Use the _split API to split an index into a new index with more primary shards

If performance issues are caused by inappropriate shard configurations when you use an Elasticsearch cluster, you can use the `_split` API to split indexes in the Elasticsearch cluster into new indexes with more primary shards in online mode. For example, if the number of primary shards for an index is small, large amounts of data may be stored in each primary shard. As a result, the cluster performance may be affected. This topic describes how to use the `_split` API to split an existing index into a new index with more primary shards.

Context

After an index is created, you cannot change the number of primary shards for the index. In most cases, if you want to change the number of primary shards for an existing index, you need to call the reindex API to reindex data, which is time-consuming. To resolve this issue, Elasticsearch provides the _split API in Elasticsearch V6.X and later versions. You can use this API to split an existing index into a new index with more primary shards in online mode. For more information about the API, see Split index API.

The following descriptions provide information about performance tests performed on the reindex API and `_split` API:

- Test environment:
  - Nodes: five data nodes, each of which offers 8 vCPUs and 16 GiB of memory
  - Data volume: 183 GiB of data stored in an index
  - Number of shards: five primary shards for the original index, 20 primary shards for the new index, and no replica shards for both indexes
- Test results

Method	Consumed time	Resource usage
reindex API	2.5 hours	The write QPS in the cluster is excessively high, and the resource usage of the data nodes is high.

Method	Consumed time	Resource usage
`_split` API	3 minutes	The CPU utilization of each data node is approximately 78%, and the minute-average load of each data node is approximately 10.

Prerequisites

- The Elasticsearch cluster is healthy, and the load of the cluster is normal.
- The number of primary shards that can be obtained after the index is split is evaluated based on the number of data nodes in and the disk space of the Elasticsearch cluster. For more information, see Shard evaluation.
- Data write operations are disabled for the index. The Elasticsearch cluster does not contain an index that is named the same as the new index.
- The Elasticsearch cluster has sufficient disk space to store the new index.

Procedure

1.
2.
3. On the Console tab of the page that appears, run the following command to create an index. In the command, configure the index.number_of_routing_shards parameter to specify the number of routing shards and the index.number_of_shards parameter to specify the number of primary shards.

   The number of primary shards for the new index must be a factor of the value of the index.number_of_routing_shards parameter and a multiple of the value of the index.number_of_shards parameter. In this example, an index named dest1 is created in an Elasticsearch V7.10 cluster. The index.number_of_routing_shards parameter is set to 24, and the index.number_of_shards parameter is set to 2. In this case, the number of primary shards that can be obtained after the index is split is 4, 6, 8, 12, or 24.

   > **Note** You must replace dest1 in the following command based on your business requirements.

```
PUT /dest1
{
  "settings": {
    "index": {
      "number_of_routing_shards": 24,
      "number_of_shards":2
    }
  }
}
```

Parameter	Description

Parameter	Description
number_of_routing_shards	The number of routing shards. This parameter defines the number of times the original index can be split or the numbers of primary shards that can be obtained after the split. When you create an index, you must make sure that the number of primary shards configured for the index is a factor of the value of this parameter. **Note** ◦ If you want to split an index in an Elasticsearch cluster of a version earlier than V7.0, you must configure the index.number_of_routing_shards parameter in the command that is used to create the index, and the maximum value of this parameter is 1024. By default, for an Elasticsearch cluster of V7.0 or later, the value of the index.number_of_routing_shards parameter is related to the number of primary shards for the index that you want to split in the cluster. If you want to split an index in an Elasticsearch cluster of V7.0 or later and this parameter is not configured in the command used to create the index, the index is split by a factor of 2 by default, and a maximum of 1,024 primary shards can be obtained after the split. For example, if the number of primary shards for the original index is 1, the number of primary shards obtained after the split can be a number from 1 to 1,024. If the number of primary shards for the original index is 5, the number of primary shards that can be obtained after the split is 10, 20, 40, 80, 160, 320, or 640. The maximum number of primary shards that can be obtained after the split is 640. ◦ When you split an index whose primary shards have been shrunk down by using the _shrink API, you must make sure that the number of primary shards obtained after the split is a multiple of the number of primary shards before the split. For example, if the number of primary shards before the split is 5, the number of primary shards obtained after the split must be a multiple of 5, such as 10, 15, 20, 25, and 30. The number of primary shards obtained after the split cannot exceed 1,024.
number_of_shards	The number of primary shards for the index.

4. Insert data.

Note The following data is used only for testing.

```
POST /dest1/_doc/_bulk
{"index":{}}
{"productName":"Daily Wealth Management for Comprehensive Health","annual_rate":"3.2200%","describe":"180-day wealth management product. Minimum investment of USD 20,000. Low-risk investment. Select whether to receive push messages for returns."}
{"index":{}}
{"productName":"Western Tongbao","annual_rate":"3.1100%","describe":"90-day wealth management product. Minimum investment of USD 10,000. Daily push messages when returns are credited to your account."}
{"index":{}}
{"productName":"Anxiang Livestock Industry","annual_rate":"3.3500%","describe":"270-day wealth management product. Minimum investment of USD 40,000. Daily push messages when returns are immediately credited to your account."}
{"index":{}}
{"productName":"Monthly 5G Device Purchase Profit","annual_rate":"3.1200%","describe":"90-day wealth management product. Minimum investment of USD 12,000. Daily push messages when returns are credited to your account."}
{"index":{}}
{"productName":"New Energy Power Wealth Management","annual rate":"3.0100%","describe":"30-day wealth management product. Minimum investment of USD 8,000. Daily push messages for returns."}
{"index":{}}
{"productName":"Microcredit Profit","annual_rate":"2.7500%","describe":"3-day popular wealth management product. No service fees. Minimum investment of USD 500. Push messages for returns."}
```

5. Disable data write operations for the index.

```
PUT /dest1/_settings
{
  "settings": {
    "index.blocks.write": true
  }
}
```

6. Split the original index into a new index with more primary shards and enable data write operations for the new index.

```
POST dest1/_split/dest3
{
  "settings": {
    "index.number_of_shards": 12,
    "index.blocks.write": null
  }
}
```

In this example, the original index dest1 is split into the new index dest3 by using the `_split` API. The number of primary shards for the new index is 12, and data write operations are enabled for the new index.

> **Notice**
>
> - The number of primary shards for the original index is 2, and the index.number_of_routing_shards parameter is set to 24. In this case, the number of primary shards for the new index must be a multiple of 2 and cannot exceed 24. Otherwise, an error is reported in the Kibana console.
> - During the split, the system merges the segments on nodes. This operation consumes the computing resources of the Elasticsearch cluster and increases the loads on the cluster. Therefore, before you split an index, you must make sure that your Elasticsearch cluster has sufficient disk space. We recommend that you split indexes during off-peak hours.
> - You must replace dest1 and dest3 in the preceding commands based on your business requirements.

7. View the result.

   Call the `_cat recovery` API to query the index split progress. If no recoveries about shard split are returned and the Elasticsearch cluster is healthy, the index split is complete.

   - Query the index split progress

     ```
     GET _cat/recovery?v&active_only
     ```

     If no index that is waiting to be split is displayed in the index column in the returned result, no recoveries about index split exist.

   - Query the health status of the Elasticsearch cluster

     ```
     GET _cluster/health
     ```

     If the returned result contains `"status" : "green"`, the Elasticsearch cluster is healthy.

FAQ

Q: Why are the CPU utilization and minute-average load of each data node in my Elasticsearch cluster not reduced after the index split operation is complete?

A: When you split an index, the system reroutes the documents in the index, and the new index contains a large number of docs.deleted documents. If you run the `GET _nodes/hot_threads` command, you can view that a merge operation is being performed on the original index. The merge operation consumes a large number of computing resources of the Elasticsearch cluster. We recommend that you split indexes during off-peak hours.

9.6.3. Use the _shrink API to shrink an index into a new index with fewer primary shards

Before you create an index in an Elasticsearch cluster, you must determine the number of primary shards to be configured for the index based on the actual volume of business data. If you have only a small amount of business data but configure a large number of primary shards for the index, resources may be excessively consumed and QPS or write throughput may be affected. In this case, we recommend that you reduce the number of primary shards configured for the index. This topic describes how to use the `_shrink` API to shrink an existing index into a new index with fewer primary shards.

Context

When you use an Elasticsearch cluster, you must take note of the total number of shards configured for indexes in the cluster and shard configuration for each index in the cluster. The larger the total number of shards configured for indexes in the cluster, the more the file handles occupied by the shards, and the more the resources consumed in the cluster. In addition, inappropriate shard configuration such as excessive primary shards affects query and write operations.

If you configure an excessive number of primary shards for an index, you can use the reindex API to reduce the number of primary shards. However, this method requires a long period of time. To reduce time costs, open source Elasticsearch provides the `_shrink` API. When you use the _shrink API, a shrink operation is performed. You must complete the following steps when you use the _shrink API to reduce the number of primary shards for an index:

1. Create a new index. The number of primary shards for the new index is less than that for the original index, but the other settings for the new index are the same as those for the original index. Relocate all shards for the original index to the same node in the Elasticsearch cluster and make sure that the reserved disk space for the node is greater than the size of data stored on all primary shards for the original index.
2. Create hard links to link segments from the original index to the new index.
3. Recover the new index. This operation is similar to opening a closed index.

The following descriptions provide information about performance tests that are performed on the reindex API and `_shrink` API:

- Test environment:
  - Data nodes: five data nodes, each of which offers 8 vCPUs and 16 GiB of memory
  - Data volume: 182 GiB of data stored in an index
  - Number of shards: 30 primary shards for the original index, 5 primary shards for the new index, and no replica shards for both indexes
- Test results

Method	Consumed time	Resource usage
reindex API	3 hours and 22 minutes	The write QPS in the cluster is excessively high, and the resource usage of the data nodes is high.
`_shrink` API	15 minutes	The computing resource usage of the node on which the shrink operation is performed is high.

Prerequisites

- The Elasticsearch cluster is healthy, and the load of the cluster is normal.
- The number of primary shards to be obtained after the shrink operation is evaluated based on the number of data nodes in the Elasticsearch cluster and the disk space of the Elasticsearch cluster. For more information, see Shard evaluation.
- The original index is in a normal state indicated by the color green.
- The number of documents stored in the original index does not exceed 2,147,483,519.
- The Elasticsearch cluster does not contain an index that is named the same as the new index.

Procedure

1.
2.
3. On the Console tab of the page that appears, run the following command to disable data write operations for the original index, set the number of replica shards for the original index to 0, and relocate all shards for the original index to the same node in the Elasticsearch cluster.

   In this example, an original index named shrink5 is used. You must replace the name of the original index in the following command based on your business requirements.

```
PUT shrink5/_settings
{
    "index.routing.allocation.require._name": "es-cn-zvp25yhyy000y****-1ab7****-0001",
    "index.blocks.write": true,
    "index.number_of_replicas": 0
}
```

Parameter	Description
index.routing.allocation.require._name	The name of the node to which you want to relocate shards. You can run the `GET _cat/nodes?v` command to obtain the name. **Note** Before you call the `_shrink` API to shrink an existing index into a new index with fewer primary shards, you must relocate all shards for the original index to the same node in the Elasticsearch cluster.
index.blocks.write	Specifies whether to disable data write operations for the original index. Set this parameter to true. The value true indicates that data write operations are disabled for the original index. **Note** Before you call the `_shrink` API to shrink an existing index into a new index with fewer primary shards, you must disable data write operations for the original index.
index.number_of_replicas	The number of replica shards for the original index.

4. Call the `_shrink` API to shrink the original index into a new index with fewer primary shards.

   The following command provides an example on how to shrink the original index shrink5 that has 30 primary shards to the new index shrink_hk5e_cn that has five primary shards. You must replace the name of the original index and that of the new index in the following command based on your business requirements.

```
POST shrink5/_shrink/shrink_hk5e_cn
{
  "settings": {
    "index.blocks.write": null,
    "index.number_of_shards": 5,
    "index.number_of_replicas": 0,
    "index.routing.allocation.require._name": null
  }
}
```

Parameter	Description
index.blocks.write	Specifies whether to disable data write operations for the new index. Set this parameter to null. This way, the settings that are copied from the original index are cleared.
index.number_of_shards	The number of primary shards for the new index. **Notice** ◦ After the shrink operation is triggered, the CPU utilization and minute-average load of the node on which the shrink operation is performed are high. We recommend that you shrink indexes during off-peak hours. ◦ The number of primary shards for the original index must be greater than that for the new index. The number of primary shards for the original index must be divisible by the number of primary shards for the new index. For example, if the number of primary shards for the original index is 8, the number of primary shards for the new index can be 4, 2, or 1. If the number of primary shards for the original index is 15, the number of primary shards for the new index can be 5, 3, or 1. If the number of primary shards for the original index is a prime number, the number of primary shards for the new index can only be 1.
index.number_of_replicas	The number of replica shards for the new index.
index.routing.allocation.require._name	The name of the node to which you want to relocate shards. Set this parameter to null. This way, the settings that are copied from the original index are cleared.

5. View the result.

   Call the `_cat recovery` API to query the index shrink progress. If no recoveries about index shrink are returned and the Elasticsearch cluster is healthy, the index shrink is complete.

   ◦ Query the index shrink progress

   ```
   GET _cat/recovery?v&active_only
   ```

If no index that is waiting to be shrunk is displayed in the index column in the returned result, no recoveries about index shrink exist.

- Query the health status of the Elasticsearch cluster

```
GET _cluster/health
```

If the returned result contains `"status" : "green"`, the Elasticsearch cluster is healthy.

FAQ

Q: Why are hard links instead of symbolic links used?

A: Hard links ensure the independence of the new index. If you use symbolic links, and you delete the original index after data is written to the new index, data in the new index is also deleted. Hard links ensure that data in the new index is not deleted.

9.6.4. Use Curator

Curator is an index management tool provided by open-source Elasticsearch. This tool allows you to create, delete, and disable indexes. It also allows you to merge index segments. This topic describes how to install Curator, use the singleton command line interface (CLI), schedule a task by using crontab, separate hot and cold data, and migrate indexes from hot nodes to warm nodes.

Install Curator

Before you install Curator, make sure that you have completed the following preparations:

- Create an Alibaba Cloud Elasticsearch cluster.
- Create an Alibaba Cloud Elastic Compute Service (ECS) instance.

  This topic uses an ECS instance that runs 64-bit CentOS 7.3 as an example. The created ECS instance must be in the same region, zone, and Virtual Private Cloud (VPC) as the Elasticsearch cluster.

Connect to the ECS instance and run the following command to install Curator:

```
pip install elasticsearch-curator
```

Note We recommend that you install Curator 5.6.0 that is compatible with Alibaba Cloud Elasticsearch V5.5.3 and V6.3.2. For more information about the compatibility between Curator and Alibaba Cloud Elasticsearch, see Version Compatibility.

Run the following command to check the version of Curator:

```
curator --version
```

If the command is successfully executed, the following result is returned:

```
curator, version 5.6.0
```

Note For more information about Curator, see Curator Index Management.

Use the singleton CLI

You can run the **curator_cli** command to perform a single action. For more information, see Singleton Command Line Interface.

> **Note**
> - The **curator_cli** command allows you to perform only one action at a time.
> - Some actions such as Alias and Restore cannot be performed on the singleton CLI.

Schedule a task by using crontab

You can use crontab and **curator** commands to schedule the actions in a task.

The following code provides an example of the **curator** command:

```
curator [OPTIONS] ACTION_FILE
Options:
  --config PATH  Path to configuration file. Default: ~/.curator/curator.yml
  --dry-run      Do not perform any changes.
  --version      Show the version and exit.
  --help         Show this message and exit.
```

When you run the **curator** command, you must specify the config.yml and action.yml files.

Separate hot and cold data

For more information, see "Hot-Warm" Architecture in Elasticsearch 5.x.

Migrate indexes from hot nodes to warm nodes

1. Create a *config.yml* file in the */usr/curator/* directory. Example:

```
client:
  hosts:
    - http://es-cn-0pxxxxxxxxxxx234.elasticsearch.aliyuncs.com
  port: 9200
  url_prefix:
  use_ssl: False
  certificate:
  client_cert:
  client_key:
  ssl_no_validate: False
  http_auth: user:password
  timeout: 30
  master_only: False
logging:
  loglevel: INFO
  logfile:
  logformat: default
  blacklist: ['elasticsearch', 'urllib3']
```

   - `hosts`: Set the value to the internal or public endpoint of the Elasticsearch cluster. The internal endpoint is used in this example.

- `http_auth`: Set the value to the username and password that are used to access the Elasticsearch cluster.

2. Create an *action.yml* file in the */usr/curator/* directory. Example:

```
actions:
  1:
    action: allocation
    description: "Apply shard allocation filtering rules to the specified indices"
    options:
      key: box_type
      value: warm
      allocation_type: require
      wait_for_completion: true
      timeout_override:
      continue_if_exception: false
      disable_action: false
    filters:
    - filtertype: pattern
      kind: prefix
      value: logstash-
    - filtertype: age
      source: creation_date
      direction: older
      timestring: '%Y-%m-%dT%H:%M:%S'
      unit: minutes
      unit_count: 30
```

In this example, indexes that are created on `hot` nodes 30 minutes ago and start with `logstash-` are migrated to `warm` nodes. You can also configure an *action.yml* file as required.

3. Check whether the **curator** command runs normally.

```
curator --config /usr/curator/config.yml /usr/curator/action.yml
```

If the curator command runs normally, information similar to the following code is returned:

```
2019-02-12 20:11:30,607 INFO      Preparing Action ID: 1, "allocation"
2019-02-12 20:11:30,612 INFO      Trying Action ID: 1, "allocation": Apply shard alloca
tion filtering rules to the specified indices
2019-02-12 20:11:30,693 INFO      Updating index setting {'index.routing.allocation.req
uire.box_type': 'warm'}
2019-02-12 20:12:57,925 INFO      Health Check for all provided keys passed.
2019-02-12 20:12:57,925 INFO      Action ID: 1, "allocation" completed.
2019-02-12 20:12:57,925 INFO      Job completed.
```

4. Enable the **curator** command to run at 15-minute intervals.

```
*/15 * * * * curator --config /usr/curator/config.yml /usr/curator/action.yml
```

9.6.5. Use the rollup mechanism to summarize traffic data

This topic describes how to use the rollup mechanism to summarize traffic data.

For time series data, data volumes increase over time. If you want to store large volumes of data, the storage costs will linearly increase. In this scenario, you can use the rollup mechanism of Elasticsearch to store data at a fraction of the cost. The following procedure demonstrates how to use the rollup mechanism to summarize Logstash traffic data.

Prerequisites

- You have the manage or manage_rollup permission.

  To use the rollup mechanism, you must have the manage or manage_rollup permission. For more information, see Security privileges.

- You have created an Alibaba Cloud Elasticsearch instance.

  For more information, see Create an Alibaba Cloud Elasticsearch cluster. This topic uses an Alibaba Cloud Elasticsearch V7.4 instance of the Standard Edition as an example.

> **Note** The rollup commands listed in this topic are of Elasticsearch V7.4. For more information about commands of Elasticsearch V6.x, see rollup job descriptions.

Context

Requirements:

- Elasticsearch provides hourly summaries of the networkoutTraffic and networkinTraffic fields at intervals of 15 minutes. The networkoutTraffic and networkinTraffic fields correspond to a specific instance ID.
- Elasticsearch uses charts presented on the Kibana console to visualize the data of the networkoutTraffic and networkinTraffic fields.

In this topic, the index that is prefixed by monitordata-logstash-sls-* is used as an example. * indicates the date in the format of YYYY-MM-DD. This type of index is generated on a daily basis. Mapping format of the index:

```
"monitordata-logstash-sls-2020-04-05" : {
    "mappings" : {
      "properties" : {
        "@timestamp" : {
          "type" : "date"
        },
        "__source__" : {
          "type" : "text",
          "fields" : {
            "keyword" : {
              "type" : "keyword",
              "ignore_above" : 256
            }
          }
        },
        "disk_type" : {
          "type" : "text",
          "fields" : {
            "keyword" : {
              "type" : "keyword",
              "ignore_above" : 256
            }
```

```
          }
        },
        "host" : {
          "type" : "keyword"
        },
        "instanceId" : {
          "type" : "keyword"
        },
        "metricName" : {
          "type" : "keyword"
        },
        "monitor_type" : {
          "type" : "keyword"
        },
        "networkinTraffic" : {
          "type" : "double"
        },
        "networkoutTraffic" : {
          "type" : "double"
        },
        "node_spec" : {
          "type" : "keyword"
        },
        "node_stats_node_master" : {
          "type" : "keyword"
        },
        "resource_uid" : {
          "type" : "keyword"
        }
      }
    }
  }
}
```

Note You can run the commands provided in this topic in the Kibana console. For more information, see Log on to the Kibana console.

Procedure

1. Step 1: Create a rollup job
2. Step 2: Start the rollup job and view the job information
3. Step 3: Query the data of the rollup index
4. Step 4: Create a rollup index pattern
5. Step 5: Create a chart for traffic monitoring in the Kibana console
6. Step 6: Create a traffic monitoring dashboard in the Kibana console

Step 1: Create a rollup job

This step provides the instructions on how to run a job, when to index a document, and which queries are performed on rollup indexes. The following example uses the `PUT _rollup/job` command to define rollup jobs within an hour.

```
PUT _rollup/job/ls-monitordata-sls-1h-job1
{
    "index_pattern": "monitordata-logstash-sls-*",
    "rollup_index": "monitordata-logstash-rollup-1h-1",
    "cron": "0 */15 * * * ?",
    "page_size" :1000,
    "groups" : {
      "date_histogram": {
        "field": "@timestamp",
        "fixed_interval": "1h"
      },
      "terms": {
        "fields": ["instanceId"]
      }
    },
    "metrics": [
        {
            "field": "networkoutTraffic",
            "metrics": ["sum"]
        },
        {
            "field": "networkinTraffic",
            "metrics": ["sum"]
        }
    ]
}
```

Parameter	Required	Type	Description
`index_pattern`	Yes	string	The index or index pattern of the rollup job. Wildcards (*) are supported.
`rollup_index`	Yes	string	The index of the rollup summary. Wildcards are not supported, and a complete name is required.
`cron`	Yes	string	The interval between rollup jobs. It is independent of the interval at which data is rolled up.
`page_size`	Yes	integer	The number of bucket results that are processed on each iteration of the rollup index. A larger value indicates faster processing and higher memory usage during the processing.
`groups`	Yes	object	Allows you to define the grouping fields and aggregation methods for jobs.

Parameter	Required	Type	Description
└ `date_histogram`	Yes	object	Allows you to roll up the date field to a time-based bucket.
└ `field`	Yes	string	The date field you want to roll up.
└ `fixed_interval`	Yes	time units	The interval at which data is rolled up. For example, if this parameter is set to 1h, the date field specified by the field parameter is rolled up on an hourly basis. This parameter specifies the minimum interval at which data is rolled up.
`terms`	No	object	None.
└ `fields`	Yes	string	The terms field set. Fields in this array can be of the keyword or numberic type, and arranged with no order required.
`metrics`	No	object	None.
└ `field`	Yes	string	The field of the metrics you want to collect. In the preceding code, this parameter is set to networkoutTraffic and networkinTraffic.
└ `metrics`	Yes	array	The operator you want to use for aggregation. If this parameter is set to sum, the sum of the networkinTraffic field is calculated. This parameter can be set to min, max, sum, average, or value count.

Note └ indicates a child parameter.

For more information about these parameters, see Create rollup jobs API. Note the following points when you configure parameters:

- If `index_pattern` is set to a wildcard pattern, make sure that the value of index_pattern is different from that of `rollup_index`. Otherwise, an error is returned.
- The mapping of rollup_index is of the object type. Make sure that index_pattern is not set to the same value as rollup_index. Otherwise, an error is returned.
- The rollup job supports only date histogram aggregation, histogram aggregation, and terms aggregation. For more information, see Rollup aggregation limitations.

Step 2: Start the rollup job and view the job information

1. Start the rollup job.

```
POST _rollup/job/ls-monitordata-sls-1h-job1/_start
```

2. View the configuration, statistics, and status of the rollup job.

```
GET _rollup/job/ls-monitordata-sls-1h-job1/
```

For more information, see Get rollup jobs API.

If the command is executed successfully, the following result is returned:

```
{
     ........
      "status" : {
        "job_state" : "indexing",
        "current_position" : {
          "@timestamp.date_histogram" : 1586775600000,
          "instanceId.terms" : "ls-cn-ddddez****"
        },
        "upgraded_doc_id" : true
      },
      "stats" : {
        "pages_processed" : 3,
        "documents_processed" : 11472500,
        "rollups_indexed" : 3000,
        "trigger_count" : 1,
        "index_time_in_ms" : 766,
        "index_total" : 3,
        "index_failures" : 0,
        "search_time_in_ms" : 68559,
        "search_total" : 3,
        "search_failures" : 0
      }
}
```

Step 3: Query the data of the rollup index

When the rollup job is executed, the structure of the rollup document is different from that of the raw data. The rollup query port rebuilds the Query DSL into a pattern that matches the rollup document, obtains the response, and restores the Query DSL to the pattern expected by the client that is used for the original query.

1. Use match_all to obtain all data of the rollup index.

```
GET monitordata-logstash-rollup-1h-1/_search
{
  "query": {
    "match_all": {}
  }
}
```

- Only one rollup index can be specified for a query. Fuzzy match is not supported. Multiple indexes can be specified for a real-time data query.
- The following queries are supported: term queries, terms queries, range queries, match all queries, and any compound queries. Compound queries are combinations of queries, including Boolean queries, boosting queries, and constant score queries. For more limits, see Rollup search limitations.

2. Use `_rollup_search` to obtain the sum of networkoutTraffic.

```
GET /monitordata-logstash-rollup-1h-1/_rollup_search
{
    "size": 0,
    "aggregations": {
        "sum_temperature": {
            "sum": {
                "field": "networkoutTraffic"
            }
        }
    }
}
```

`_rollup_search` supports subsets of common search operation features:

- query: the Query DSL parameter with specific limits. For more information, see Rollup search limitations and Rollup aggregation limitations.
- aggregations: the aggregation parameter.

`_rollup_search` does not support the following features:

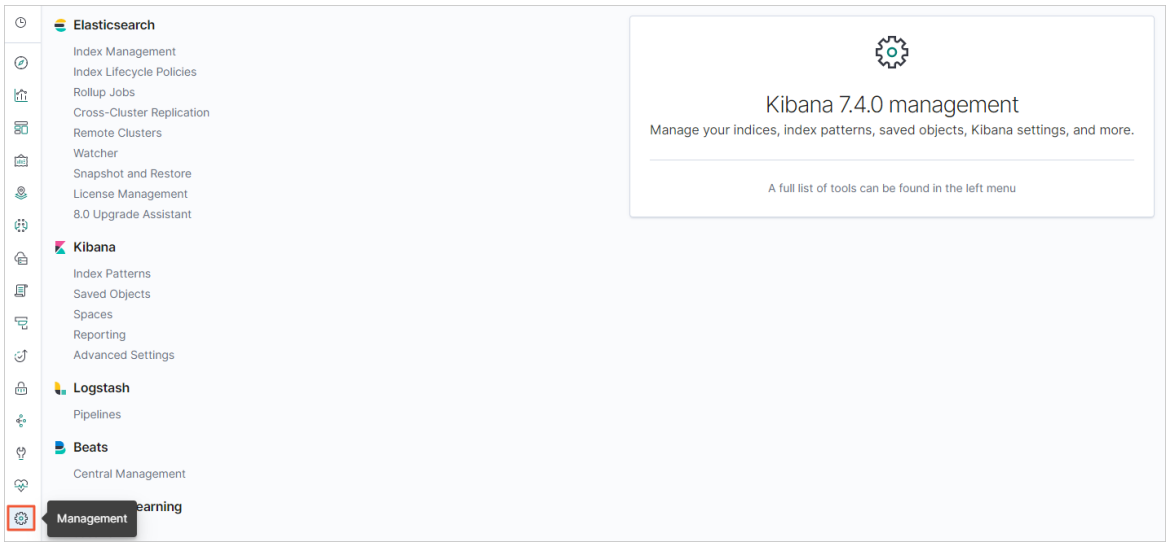
- size: Set this parameter to 0 or do not specify this parameter. This is because rollup is used only for data aggregation and the query result cannot be returned.
- Parameters such as highlighter, suggestors, post_filter, profile, and explain are not supported.

Step 4: Create a rollup index pattern

1. Log on to the Kibana console.

   For more information, see Log on to the Kibana console.

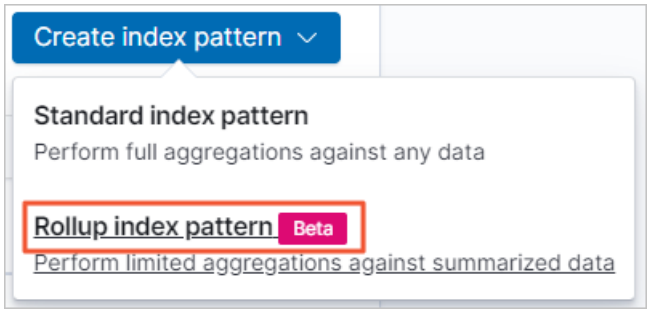
2. In the left-side navigation pane, click the **Management** icon.



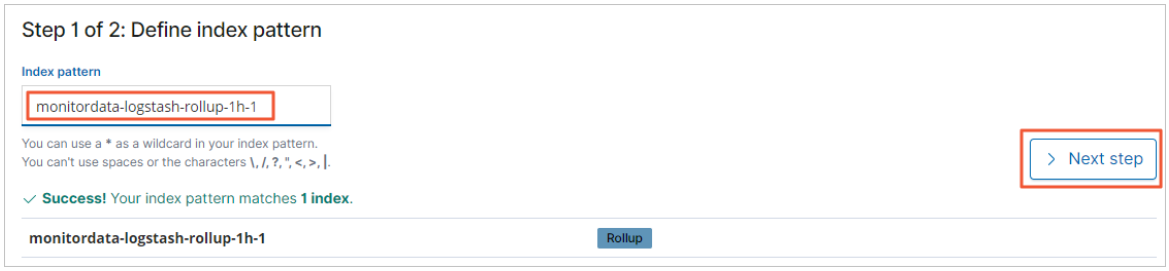
3. In the **Kibana** area, click **Index Patterns**.
4. (Optional)Close the **About index patterns** page.

> **Note** Skip this step if this is not the first time you created an index pattern.

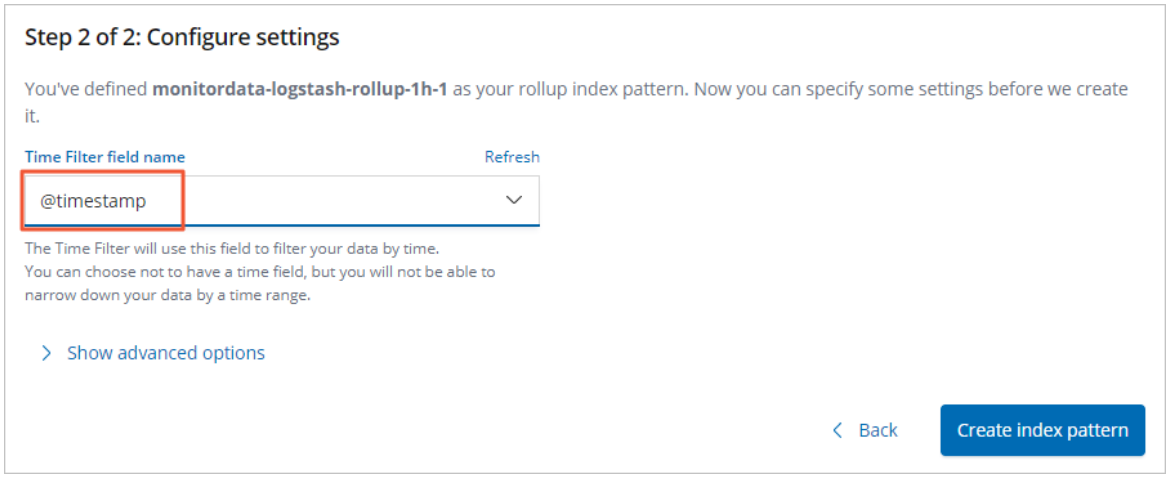
5. Choose **Create index pattern > Rollup index pattern**.



6. In the Index pattern field, enter an index pattern name such as monitordata-logstash-rollup-1h-1, and then click **Next step**.



7. From the **Time Filter field name** drop-down list, select **@timestamp**.



8. Click **Create index pattern**.

Step 5: Create a chart for traffic monitoring in the Kibana console

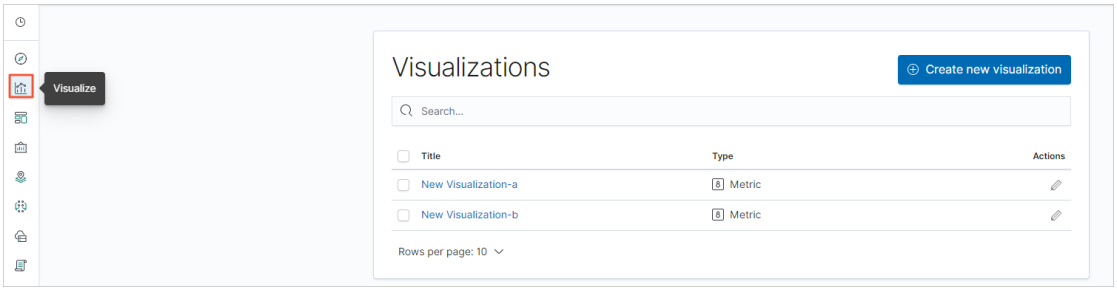
The following procedure demonstrates how to create networkinTraffic and networkoutTraffic charts for the rollup index in the Kibana console.

1. Log on to the Kibana console.

   For more information, see Log on to the Kibana console.

2. Create a line chart.

i. In the left-side navigation pane, click the **Visualize** icon.



ii. Click **Create new visualization**.

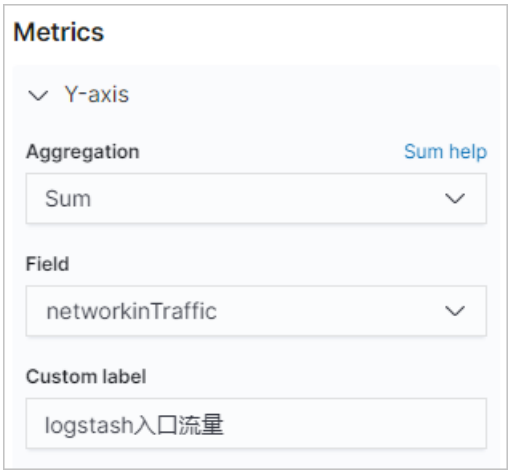
iii. In the **New Visualization** dialog box that appears, click **Line**.

iv. In the index pattern list, click the created rollup index pattern.

3. Specify parameters in **Metrics** and **Buckets**.

i. In the **Metrics** section, click >.

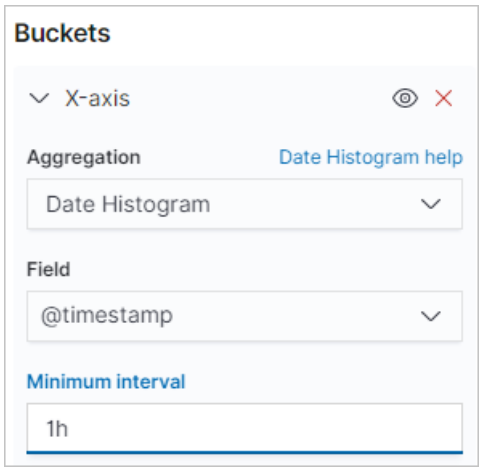
ii. Specify **Y-axis** parameters.



Parameter	Description
Aggregation	Set the parameter value to **Sum**.
Field	Set the parameter value to networkinTraffic or networkoutTraffic.
Custom label	Enter a custom Y-axis label.

iii. In the **Buckets** section, choose **Add > X-axis**.

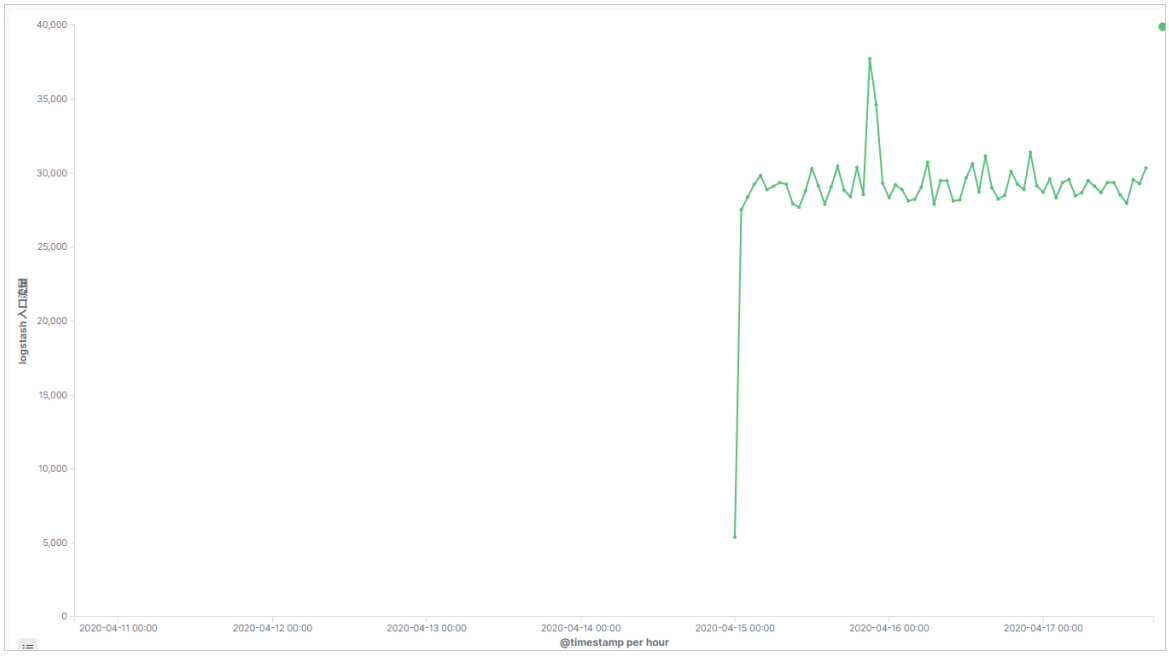
iv. Specify **X-axis** parameters.



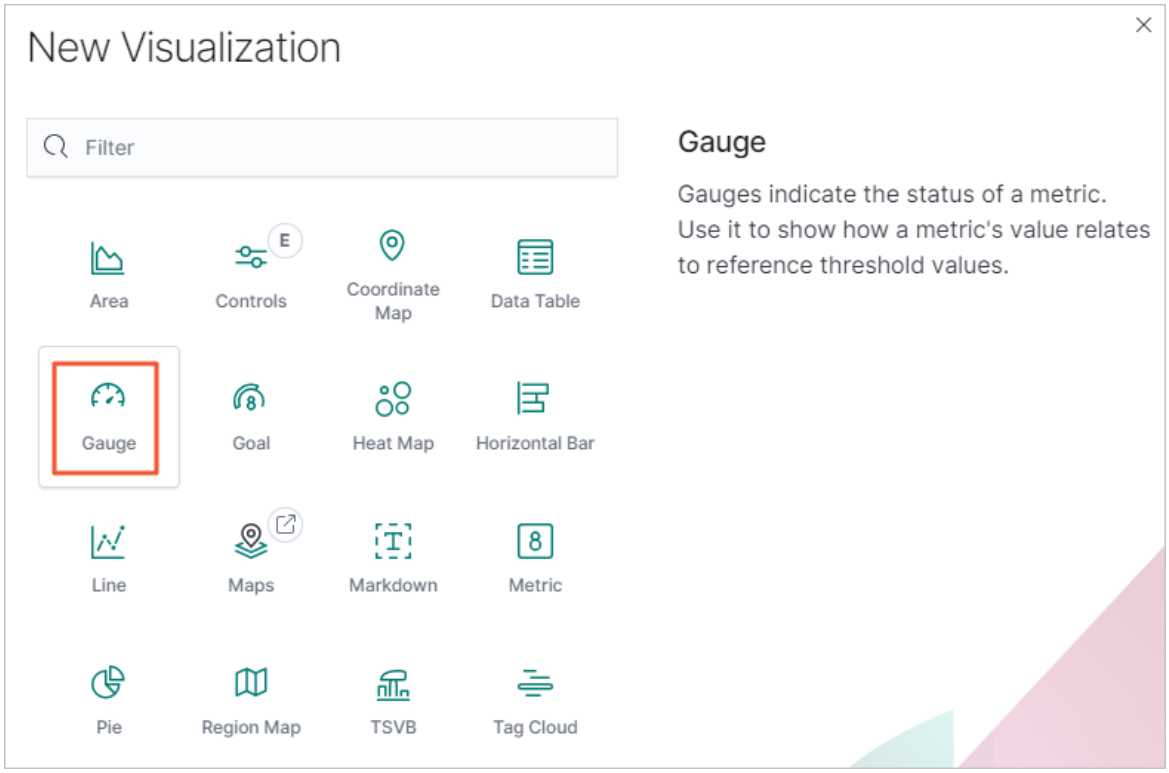
Parameter	Description
Aggregation	Set the parameter value to `date_histogram` defined for `group` in Step 1: Create a rollup job.
Field	Set the parameter value to **@timestamp**.
Minimum interval	The default value is the aggregation time granularity defined in the rollup job. The value must be an integer multiple of the rollup interval, such as 2h or 3h.

v. Click the ▷ icon.

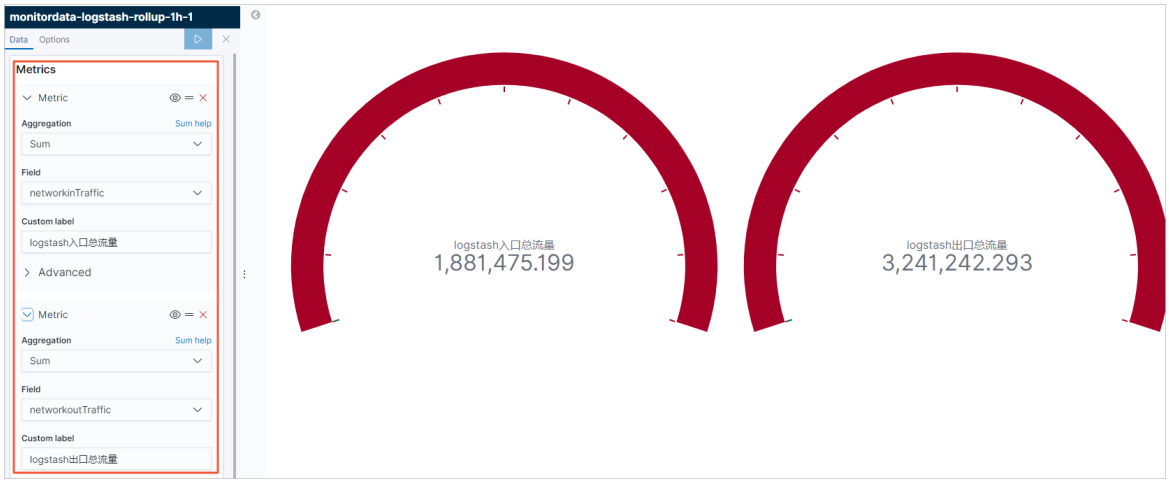
4. In the top navigation bar, click **Save**.
   After the configuration is successful, you can view the line chart, as shown in the following figure.



5. Create a gauge chart in the same way.

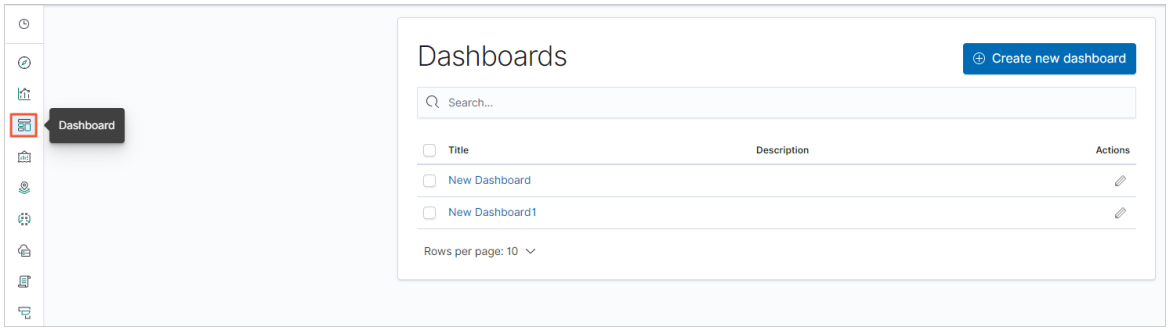


6. Configure parameters for the gauge chart. Sample configurations:



Step 6: Create a traffic monitoring dashboard in the Kibana console

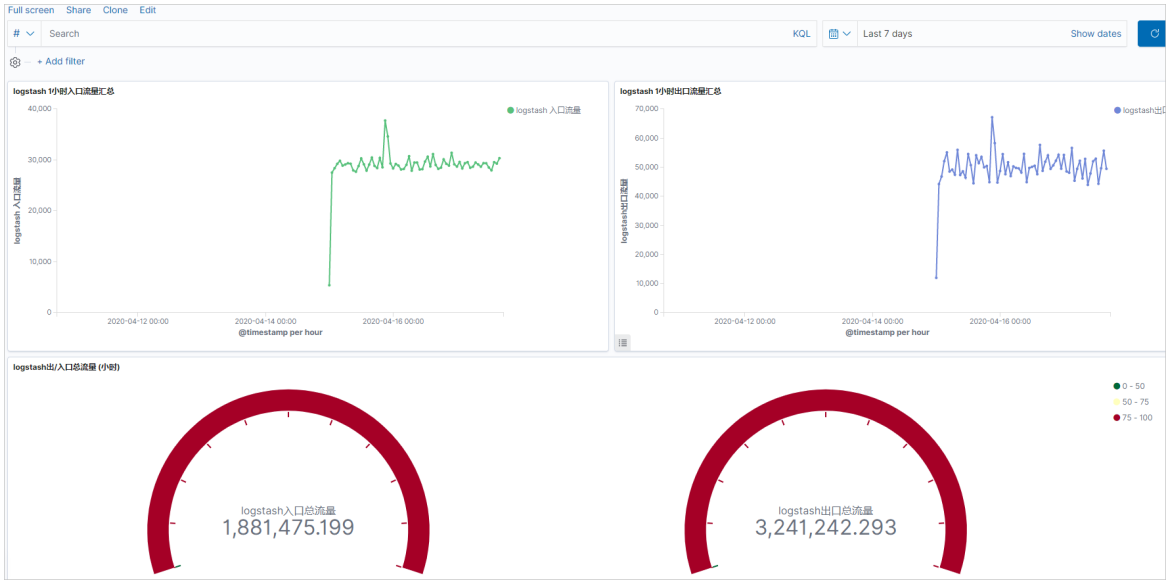
1. In the Kibana console, click the **Dashboard** icon in the left-side navigation pane.



2. Click **Create new dashboard**.

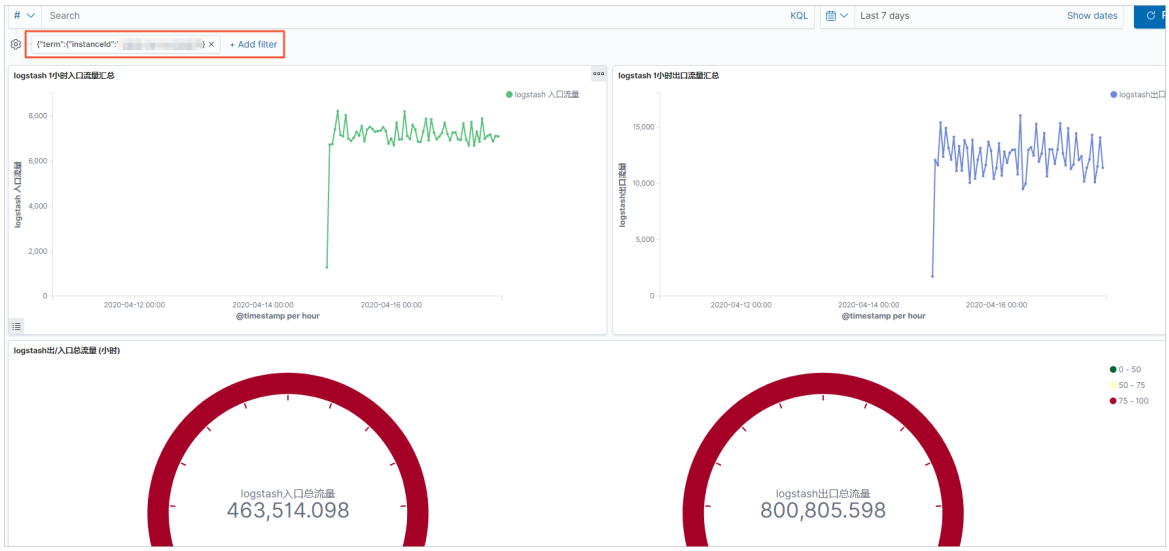
3. In the top navigation bar, click **Add**.
4. On the **Add panels** page, click the chart configured in Step 5.
5. Close the **Add panels** page. In the top navigation bar, click **Save**.
6. Modify the dashboard name and click **Confirm Save**.

   After the dashboard configuration is saved, you can view the dashboard.



7. Click **+ Add filter**, select a filter item, configure the filter conditions, and click **Save**.

   In this step, term is used as a filter item to query networkoutTraffic and networkinTraffic of an instance. The following figure shows the final presentation of dashboards.



9.6.6. Use Cerebro to access an Elasticsearch cluster

Use Cerebro to access an Elasticsearch cluster

In addition to Kibana, curl commands, and clients, you can use third-party plug-ins or tools such as Elasticsearch-Head and Cerebro to access an Elasticsearch cluster. The Elasticsearch-Head plug-in is not maintained in versions later than Elasticsearch 5.x. Therefore, we recommend that you use Cerebro to access your Elasticsearch cluster. This topic describes how to use Cerebro to access an Elasticsearch cluster.

Prerequisites

- An Alibaba Cloud Elasticsearch cluster is created.

  For more information, see Create an Alibaba Cloud Elasticsearch cluster.

- An Alibaba Cloud Elastic Compute Service (ECS) instance is created. This instance must reside in the same virtual private cloud (VPC) as the Elasticsearch cluster.

  For more information, see Create an instance by using the wizard. The ECS instance is used to install Cerebro.

  > **Note** If your ECS instance resides in a different VPC from your Elasticsearch cluster, or you want to install Cerebro on an on-premises machine, you can access the Elasticsearch cluster over the Internet. In this case, take note of the following items:
  >
  > - Access over the Internet is less secure than access over an internal network.
  > - Network latency may cause services to be unstable.
  > - You must turn on Public Network Access for your Elasticsearch cluster and configure a whitelist for access to the Elasticsearch cluster over the Internet. For more information, see Configure a public or private IP address whitelist for an Elasticsearch cluster.

- The JDK is installed on the ECS instance. The JDK version must be 1.8 or later.

Context

- Cerebro is a third-party tool.
- You can use Cerebro to access the Elasticsearch cluster over the Internet by using the public endpoint and the related port of this cluster.

Procedure

1. Connect to the ECS instance.

   For more information, see 连接ECS实例.

2. Download and decompress the Cerebro installation package.
   - Run the following command to download the Cerebro installation package:

     ```
     wget https://github.com/lmenezes/cerebro/releases/download/v0.9.0/cerebro-0.9.0.tgz
     ```

   - Run the following command to decompress the Cerebro installation package:

     ```
     tar -zxvf cerebro-0.9.0.tgz
     ```

3. Modify the configuration file of Cerebro and associate Cerebro with the Elasticsearch cluster that you want to access.
   i. Open the application.conf file.

   ```
   vim cerebro-0.9.0/conf/application.conf
   ```

ii. Configure `hosts` based on the following instructions.

```
# A list of known hosts
hosts = [
  #{
  #  host = "http://localhost:9200"
  #  name = "Localhost cluster"
  #  headers-whitelist = [ "x-proxy-user", "x-proxy-roles", "X-Forwarded-For" ]
  #}
  # Example of host with authentication
  {
    host = "http://es-cn-r[redacted].elasticsearch.aliyuncs.com:9200"
    name = "es-cn-[redacted]"
    auth = {
      username = "elastic"
      password = "[redacted]"
    }
  }
```

Note You can associate Cerebro with multiple Elasticsearch clusters. Multiple clusters are separated with commas (,).

Parameter	Description
host	The URL that is used to access the Elasticsearch cluster. Specify the URL in the format of `http://<Internal endpoint of the Elasticsearch cluster>:9200`. You can obtain the internal endpoint from the Basic Information page of the cluster. For more information, see View the basic information of a cluster.
name	The ID of the Elasticsearch cluster. You can obtain the ID from the Basic Information page of the cluster. For more information, see View the basic information of a cluster.
username	The username that is used to access the Elasticsearch cluster. Default value: elastic. **Notice** To ensure system security, we recommend that you do not use the elastic username. You can use a custom username instead. Before you use a custom username, you must create a role for it and grant the required permissions to the role. For more information, see Use the RBAC mechanism provided by Elasticsearch X-Pack to implement access control.
password	The password that corresponds to the username. The password that corresponds to the elastic username is specified when you create your Elasticsearch cluster. If you forget the password, you can reset it. For more information about the precautions and procedures for resetting a password, see Reset the access password for an Elasticsearch cluster.

iii. Start Cerebro after you save the modifications.

```
cd cerebro-0.9.0
bin/cerebro
```

After Cerebro is started, the result shown in the following figure is returned.

```
[root@VM01 cerebro-0.9.0]# bin/cerebro
[info] play.api.Play - Application started (Prod) (no global state)
[info] p.c.s.AkkaHttpServer - Listening for HTTP on /0.0.0.0:9000
```

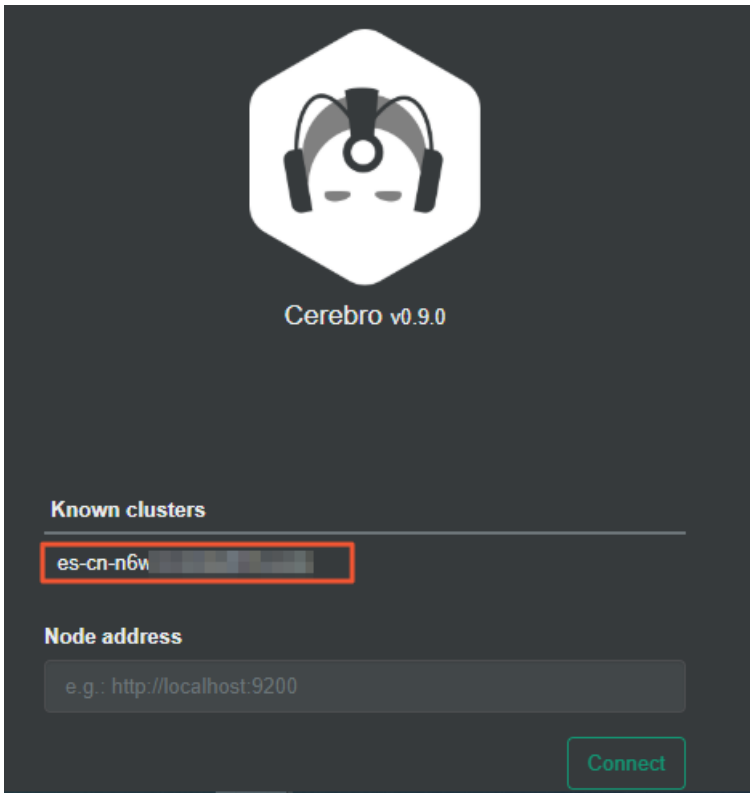
4. Use Cerebro to access the Elasticsearch cluster.

i. Configure a security group for the ECS instance. On the **Inbound** tab, add the IP address of the Elasticsearch cluster that you want to access and set Port Range to 9000.

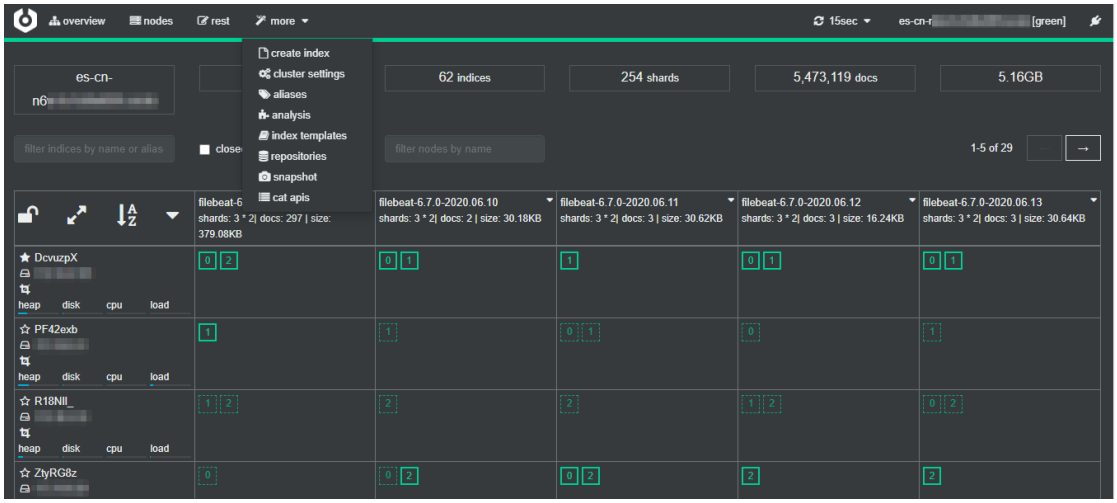
For more information, see Add a security group rule.

ii. Enter http://<Public IP address of the ECS instance>:9000 in the address bar of a browser.

iii. On the logon page of Cerebro, click the ID of the Elasticsearch cluster that you want to access.



iv. In the Cerebro console, view the status and the number of indexes, shards, and documents of the cluster and perform operations as required.



Note For more information about the usage notes of Cerebro, see Getting Started with Cerebro.

9.7. Cluster alerting

9.7.1. Configure a DingTalk chatbot to receive alert notifications from X-Pack Watcher

Configure X-Pack Watcher

X-Pack Watcher is a monitoring and alerting service developed for Elasticsearch. If you configure X-Pack Watcher for your cluster, X-Pack Watcher can trigger actions when specific conditions are met. For example, if the logs index contains errors, X-Pack Watcher triggers the system to send alert notifications by using emails or DingTalk messages. This topic describes how to configure a DingTalk chatbot to receive alert notifications from X-Pack Watcher.

Background information

X-Pack Watcher allows you to create watches. A watch consists of a trigger, an input, a condition, and actions.

- Trigger

  Determines when a watch starts to run. You must configure a trigger for each watch. X-Pack Watcher allows you to create various types of triggers. For more information, see Schedule Trigger.

- Input

  Loads data to the payload of a watch. Inputs are used as filters to match the specified type of index data. For more information, see Inputs.

- Condition

  Controls whether a watch performs actions.

- Actions

Determines the actions that a watch performs when the specified condition is met. In this example, the webhook action is used.

Prerequisites

- A single-zone Alibaba Cloud Elasticsearch cluster is created.

  For more information, see Create an Alibaba Cloud Elasticsearch cluster.

  > **Note** In the original network architecture, X-Pack Watcher is available only for single-zone Elasticsearch clusters. In the new network architecture, X-Pack Watcher is available for both single-zone Elasticsearch clusters and multi-zone Elasticsearch clusters.

- X-Pack Watcher is enabled for the Elasticsearch cluster. By default, X-Pack Watcher is disabled.

  For more information, see Configure the YML file.

- An Elastic Compute Service (ECS) instance is created in your virtual private cloud (VPC), and the required applications are deployed on the ECS instance.

  For more information, see Create an instance by using the wizard.

  > **Note**
  > - The ECS instance is used as a backend server to receive requests that are forwarded by a Server Load Balancer (SLB) instance. The ECS instance can be deployed in a zone that is different from the SLB instance but must be deployed in the same VPC and region as the SLB instance.
  > - X-Pack Watcher cannot directly access the Internet. It must use the internal endpoint of your Elasticsearch cluster to access the Internet. In this case, you can enable source network address translation (SNAT) for or associate an elastic IP address (EIP) with an ECS instance that is deployed in a VPC. This way, you can use the ECS instance as a proxy to forward requests.

Precautions

The network architecture of Alibaba Cloud Elasticsearch in different regions has been adjusted since October 2020. The adjustment has the following impacts on clusters:

- Clusters that are created before October 2020 are deployed in the original network architecture. In this architecture, clusters are deployed in the VPCs that are created by users. If you want a cluster that is deployed in this architecture to access the Internet, you can use an ECS instance for which SNAT is enabled or use an NGINX proxy to forward requests.
- Clusters that are created in October 2020 or later are deployed in the new network architecture. If you want to use X-Pack Watcher for an Elasticsearch cluster that is created in October 2020 or later, you must first use the PrivateLink service to establish private connections between VPCs. For more information, see Configure a private connection for an Elasticsearch cluster. If you want a cluster that is deployed in the new network architecture to access the Internet, you can configure an NGINX proxy to forward requests.

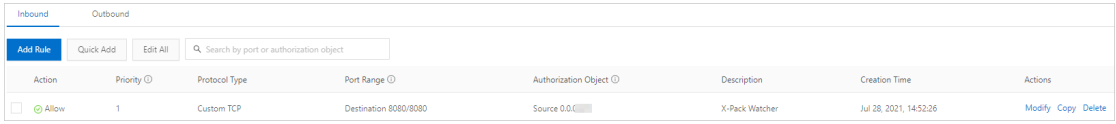
Procedure

1. Configure a private connection to the Elasticsearch cluster and obtain the domain name of the related endpoint. The domain name is used to access external services.

For more information, see Configure a private connection for an Elasticsearch cluster.

> **Note** This step is required only for a cluster that is deployed in the new network architecture.

2. Configure a security group rule for the ECS instance.
   i. Log on to the ECS console.
   ii. In the left-side navigation pane, click **Instances**.
   iii. On the Instances page, find the ECS instance and choose **More > Network and Security Group > Configure Security Group** in the **Actions** column.
   iv. On the **Security Groups** tab, find your security group and click **Add Rules** in the **Actions** column.
   v. On the **Inbound** tab, click **Add Rule**.
   vi. Configure parameters.

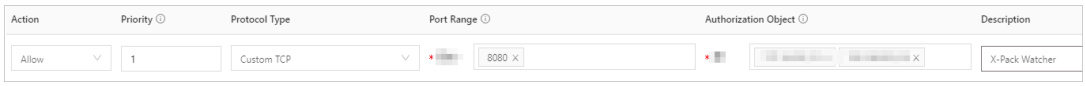


Parameter	Description
Action	Select **Allow**.
Priority	Retain the default value.
Protocol Type	Select **Custom TCP**.
Port Range	Set this parameter to the port that you frequently use. If you want to configure an NGINX proxy, you must configure this parameter. In this example, port 8080 is used.
Authorization Object	Enter the IP addresses of all the nodes in the Elasticsearch cluster. **Note** For more information about how to obtain the IP addresses of the nodes, see View the basic information of nodes.
Description	The description of the rule.

   vii. Click **Save**.
3. Configure an NGINX proxy.
   i. Install NGINX on the ECS instance.

ii. Configure the nginx.conf file.

Replace the `server` configuration in the nginx.conf file with the following code.




```
server
  {
    listen 8080;# Listening port
    server_name localhost;# Domain name
    index index.html index.htm index.php;
    root /usr/local/webserver/nginx/html;# Website directory
      location ~ .*\.(php|php5)?$
    {
      #fastcgi_pass unix:/tmp/php-cgi.sock;
      fastcgi_pass 127.0.0.1:9000;
      fastcgi_index index.php;
      include fastcgi.conf;
    }
    location ~ .*\.(gif|jpg|jpeg|png|bmp|swf|ico)$
    {
      expires 30d;
      # access_log off;
    }
    location / {
      proxy_pass <Webhook URL of the DingTalk chatbot>;
    }
    location ~ .*\.(js|css)?$
    {
      expires 15d;
      # access_log off;
    }
    access_log off;
  }
```

Replace <Webhook URL of the DingTalk chatbot> with the webhook URL of the DingTalk chatbot that you configured to receive alert notifications.

> **Note** To obtain the webhook URL of the DingTalk chatbot, create an alert group in DingTalk. In the upper-right corner of the DingTalk group, click the Group Settings icon. In the Group Settings panel, click Group Assistant. In the Group Assistant panel, click Add Robot. In the ChatBot dialog box, click the Add icon on the right side of Add Robot to add a chatbot that you can access by using a webhook. Then, you can view the webhook URL of the DingTalk chatbot.

iii. Reload the NGINX configuration file and restart NGINX.

```
/usr/local/webserver/nginx/sbin/nginx -s reload            # Reload the NGINX con
figuration file.
/usr/local/webserver/nginx/sbin/nginx -s reopen            # Restart NGINX.
```

4. Create a watch for alerting.

i. Log on to the Kibana console of the Elasticsearch cluster.

> **Note** For more information, see Log on to the Kibana console.

ii. In the left-side navigation pane, click **Dev Tools**.

iii. On the **Console** tab of the page that appears, run the following command to create a watch.

In this example, a watch named `log_error_watch` is created to search the `logs` index for `errors` every `10 seconds`. If more than `0` errors are found, an alert is triggered.

```
PUT _xpack/watcher/watch/log_error_watch
{
  "trigger": {
    "schedule": {
      "interval": "10s"
    }
  },
  "input": {
    "search": {
      "request": {
        "indices": ["logs"],
        "body": {
          "query": {
            "match": {
              "message": "error"
            }
          }
        }
      }
    }
  },
  "condition": {
    "compare": {
      "ctx.payload.hits.total": {
        "gt": 0
      }
    }
  },
  "actions" : {
  "test_issue" : {
    "webhook" : {
      "method" : "POST",
      "url" : "http://<yourAddress>:8080",
      "body" : "{\"msgtype\": \"text\", \"text\": { \"content\": \"An error is fo
und. Handle the error immediately.\"}}"
    }
  }
}
}
```

Parameters

Parameter	Network architecture type	Value	Description
<yourAddress>	New network architecture	Domain name of the endpoint	In the new network architecture, private connections need to be established between VPCs, and the domain name of the related endpoint is used to forward requests. **Notice** You must set the parameter to the domain name of the related endpoint rather than the domain name of the related endpoint service. For more information about how to obtain the domain name of an endpoint, see View the domain name of an endpoint.
	Original network architecture	IP address of the NGINX proxy	The NGINX proxy in the same VPC as the Elasticsearch cluster is used to forward requests over the Internet.
		Webhook URL of the DingTalk chatbot	The SNAT feature must be enabled. This feature enables an ECS instance in a VPC to access the Internet if no public IP address is associated with the ECS instance.

Notice

- If the error `No handler found for uri [/_xpack/watcher/watch/log_error_watch_2] and method [PUT]` is returned after you run the preceding command, X-Pack Watcher is disabled for the Elasticsearch cluster. In this case, enable X-Pack Watcher and run the command again. For more information, see Configure the YML file.
- When you create a DingTalk chatbot, you must configure security settings. This is because the body parameter in the preceding code must be specified based on the security settings. For more information, see Configure security settings. In this example, **Security Settings** is set to **Custom Keywords** and the error keyword is specified. In this case, the DingTalk chatbot sends alert notifications only if the content field in the body parameter contains error.

If you no longer require this watch, you can run the following command to delete the watch:

```
DELETE _xpack/watcher/watch/log_error_watch
```