

Elasticsearch

Best Practices

Best Practices

Build a visualized O&M system with Beats

This topic describes how to use Metricbeat to collect metrics from a development machine (this topic takes a MacBook as an example), send metrics to Alibaba Cloud Elasticsearch (ES), and view a corresponding dashboard in Kibana.

Background

Beats is a platform for single-purpose data shippers. After you install Beats, the lightweight Beats agents send data from your instances to target outputs, such as Logstash or Elasticsearch.

As an agent of Beats, Metricbeat is a lightweight shipper designed to collect metrics from your systems and services and send the metrics to your target output, such as Elasticsearch.

1. Purchase and configure an ES instance

If you do not have an ES instance, you must activate the Elasticsearch service and purchase an ES instance. Then, data in your local development machine (that is, the MacBook) can then be sent to an Alibaba Cloud ES instance through a private IP address or a public IP address provided by your ES instance.

Notes:

If you are using an ES public IP address, you must click the corresponding Public Address switch to turn it on to access your ES instance over the Internet, and add the public IP address of your development machine (that is, the MacBook) to the **Public IP Address Whitelist** under the Network and Snapshots area.

If you are using an ES private IP address, you need to purchase an Alibaba Cloud Elastic Compute Service (ECS) instance in the same **VPC** and **Region** as your ES instance.

1. Log on to the ES console.
2. Click **Manage** at the right of the target ES instance and then, in the left-side navigation pane, click **Network and Snapshots**.
3. Click the corresponding Public Address switch to turn it on.

Add the public IP address of your MacBook to the Public IP Address whitelist.

Notes:

If you are using a public network, you need to add the jump server IP of the Internet node to the whitelist. If the jump server IP cannot be obtained, we recommend that you configure 0.0.0.0/1, 128.0.0.0/1 to open as many IP addresses as possible (this topic takes the recommended configuration as an example). **However, this configuration will fully expose your ES instance to the Internet.**

In the left-side navigation pane, click **Basic Information** and record the public IP address of your ES instance.

6. Modify the YAML configuration by setting **Create Index Automatically** to **Enable** (by default, it is set to Disable). You must restart your ES instance for the change to take effect.

2. Download and configure Metricbeat

Download the package of your required version of Metricbeat:

- MacOS
- Linux 32-bit
- Linux 64-bit
- Windows 32-bit
- Windows 64-bit

Unzip your package and open the Metricbeat file:

Open and edit the **Elasticsearch output** section of metricbeat.yml by uncommenting the corresponding content:

Notes:

ES provides the following access control information

- hosts: the public IP or private IP of your ES instance.
- protocol: Set to http.
- username: By default, it is **elastic**.

- password: the logon password you set when you purchased your ES instance.

3. Start Metricbeat

Use Metricbeat to send data to your ES instance.

```
./metricbeat -e -c metricbeat.yml
```

4. View the dashboard in Kibana

Open the Kibana console (integrated in your ES instance) and then go to the dashboard page.

Note:

If you have not created an **Index Pattern** in Kibana, the corresponding information may not be displayed on the dashboard. To resolve this issue, create an Index Pattern and then go to the dashboard page to view the corresponding content.

List of various related metrics.

CPU metrics.

Note:

You can configure data to be refreshed every five seconds, generate a corresponding report, and connect to webhooks to trigger alerts when exceptions are detected.

Cloud data import

Import data from Alibaba Cloud to Alibaba Cloud ES (offline)

Alibaba Cloud stores an abundance of cloud storage and database products. If you want to analyze and search for data in these products, visit **Data Integration**, which allows you to synchronize offline data to Elasticsearch every 5 minutes.

Supported data source

- Alibaba Cloud database (MySQL, PostgreSQL, SQL Server, PPAS, MongoDB, and HBase)
- Alibaba Cloud DRDS
- Alibaba Cloud MaxCompute (ODPS)
- Alibaba Cloud OSS
- Alibaba Cloud Table Store
- Self-developed HDFS, Oracle, FTP, DB2, and self-developed versions of the previous cloud databases

Note:

Data synchronization may produce public network traffic cost.

Procedure

Take the following steps to import offline data.

- Prepare an ECS instance that can interact with Elasticsearch within a VPC. This ECS instance will obtain data sources and execute a job to write ES data (the job is centrally issued by Data Integration).
- You need to activate the Data Integration service and register the ECS instance to the Data Integration service as an executable job resource.
- Configure a data synchronization script and make it run periodically.

Steps

Buy an ECS instance that is in the same VPC as the Elasticsearch service. Allocate a public IP address to the ECS instance or enable the elastic IP address for the ECS instance. To lower costs, you can use an existing ECS instance. For how to buy an ECS instance, see [Buy an ECS instance](#).

Note:

CentOS 6, CentOS 7, and AliyunOS are recommended.

If the added ECS instance needs to run MaxCompute or synchronization tasks, verify whether the current Python version of the ECS instance is 2.6 or 2.7 (The Python version of CentOS 5 is 2.4 while those of other operating systems are later than 2.6).

- Ensure that the ECS instance has a public IP address.

Log on to the **Data Integration console** to open the workbench.

If Data Integration or DataWorks is not enabled, follow the instructions to activate the Data Integration service. This is a **paid service**, so check the quoted price against your budget.

Go to the **Project Management-Scheduling Resource Management** page of the Data Integration service to configure the ECS instance in the VPC as a scheduling resource. For more information, see [Scheduling resource configuration](#).

Configure the data synchronization script in the Data Integration service. For the configuration procedure, see [Data synchronization script configuration](#).

For the instructions on configuring Elasticsearch, see [ES Writer](#).

Note:

- The synchronization script configuration includes three parts: Reader is the configuration of upstream data source (cloud product ready for data synchronization), Writer is the configuration of ES, and setting refers to the synchronization configurations such as packet loss rate and maximum concurrency.
- The `accessId` and `accessKey` of ES Writer are the Elasticsearch user name and password, respectively.

After configuring the script, submit the data synchronization job. Set the job execution cycle and click **Ok**.

Note:

- If you are configuring a periodic scheduling, set the parameters such as Job Start Time, Execution Interval, and Job Lifecycle in this window.
- A periodic job is executed at 00:00 on the next day according to the rule you have configured.

After the submission, go to the **O&M Center-Task Scheduling** page to find the submitted job, and change its scheduling resource from default to the scheduling resource you have configured.

Import real-time data

This function is currently under development and will become available in the future.

Synchronize Hadoop and ES data with DataWorks

This topic describes how to use the data synchronization feature of DataWorks to migrate data from Hadoop to Alibaba Cloud Elasticsearch (ES), and analyze the data. You can also use Java codes to synchronize data.

Prerequisites

Create a Hadoop cluster

You must create a Hadoop cluster to perform data migration. This topic uses the Alibaba Cloud E-MapReduce service (EMR) to create a Hadoop cluster. For more information, see [Create a cluster](#).

Specifically, the following EMR Hadoop version information is used:

- EMR version: EMR-3.11.0
- Cluster type: HADOOP
- Services: HDFS2.7.2 / YARN2.7.2 / Hive2.3.3 / Ganglia3.7.2 / Spark2.3.1 / HUE4.1.0 / Zeppelin0.8.0 / Tez0.9.1 / Sqoop1.4.7 / Pig0.14.0 / ApacheDS2.0.0 / Knox0.13.0

Additionally, this topic uses a VPC network for the Hadoop cluster, sets the region to China East 1 (Hangzhou), sets a public and private IP for the ECS master nodes, and selects non-high availability (non-HA) mode.

Elasticsearch

Log on to the [Elasticsearch console](#) and select the same region and VPC network as the EMR cluster. For information about purchasing an ES instance, see [Purchase and configuration](#).

Subscription

Pay-As-You-Go

Region

China

China (Beijing)

China

China

Asia Pacific SOU 1 (Mumbai)

Asia Pacific SE 1 (Singapore)

China (Hong Kong)

US West 1 (Silicon Valley)

Asia Pacific SE 3 (Kuala Lumpur)

Germany

日本

亚太东南 2 (澳大)

Asia Pacific SE 5 (Jakarta)

Zone

Hangzhou Zone B

Version

5.5.3 with X-Pack

6.3 with X-Pack

Network Type

VPC

VPC

emr_test_vpc

Create VPC/Subnet (Switch). Refresh the page after the creation is complete

VSwitch

No available VSwitches. Create a VSwitch>>>>

Instance Type

1Core2G

1Core2G Instance type is intended for testing only. It is not suitable for the production environment and is excluded from the SLA after-sales guarantee.

Amount

3

Two node cluster has the risk of split-brain, please choose very carefully

Username

elastic

Used to access Elasticsearch and log on to Kibana.

Password

Please enter your password

The password can be 8 to 32 characters in length and must contain three of the following conditions: uppercase letters, lowercase letters, numbers, and special characters (!@#\$%^&*()_+=).

Please confirm your password

DataWorks

Create a DataWorks project and set the region to **China East 1 (Hangzhou)**. The following example uses the project bigdata_DOC.

Prepare data

To create test data in the Hadoop cluster, follow these steps:

Log on to the EMR console, go to **Old EMR Scheduling**, and in the left-side navigation pane, click **Notebook**.

Click **File->New notebook**. In this example, a notebook named **es_test_hive** is created. Set the default type to **Hive**. The attached cluster is the EMR Hadoop cluster created.

Enter the syntax for creating a Hive table.

```
CREATE TABLE IF NOT
EXISTS hive_esdoc_good_sale(

create_time timestamp,

category STRING,

brand STRING,

buyer_id STRING,

trans_num BIGINT,

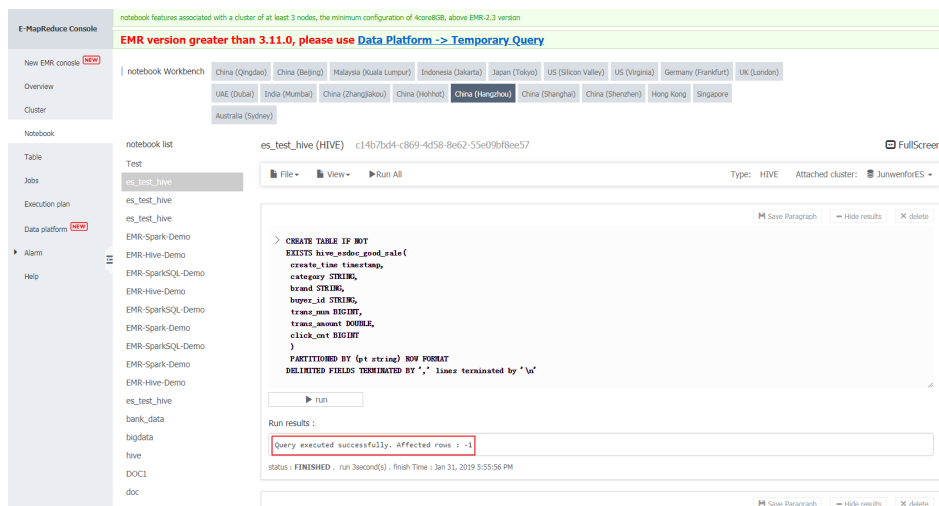
trans_amount DOUBLE,

click_cnt BIGINT

)

PARTITIONED BY (pt string) ROW FORMAT
DELIMITED FIELDS TERMINATED BY ',' lines terminated by '\n'
```

Click **Run**. If the message **Query executed successfully** displays, then the table **hive_esdoc_good_sale** was created successfully in the EMR Hadoop cluster, as shown in the following figure.



Insert test data. You can import data from OSS, or other data sources, or insert data manually. This example inserts data manually. The script for inserting data is as follows:

```
insert into
hive_esdoc_good_sale PARTITION(pt =1 ) values('2018-08-21','Jacket ','Brand A','lilei',3,500.6,7),('2018-08-
22','Fresh food','Brand B','lilei',1,303,8),('2018-08-22','Jacket ','Brand C','hanmeimei',2,510,2),('2018-08-
22','Bathroom accessory','Brand A','hanmeimei',1,442.5,1),('2018-08-22','Fresh food','Brand
```

```
D,'hanmeimei',2,234,3),('2018-08-23','Jacket ','Brand B','jimmy',9,2000,7),('2018-08-23','Fresh food','Brand A','jimmy',5,45.1,5),('2018-08-23','Jacket ','Brand E','jimmy',5,100.2,4),('2018-08-24','Fresh food','Brand G','peiqi',10,5560,7),('2018-08-24','Bathroom accessory','BrandF','peiqi',1,445.6,2),('2018-08-24','Jacket ','Brand A','ray',3,777,3),('2018-08-24','Bathroom accessory','Brand G','ray',3,122,3),('2018-08-24','Jacket ','Brand C','ray',1,62,7) ;
```

After data is inserted successfully, run the `select * from hive_esdoc_good_sale where pt =1;` statement, and then check that the data is already in the EMR Hadoop cluster table.



Synchronize data

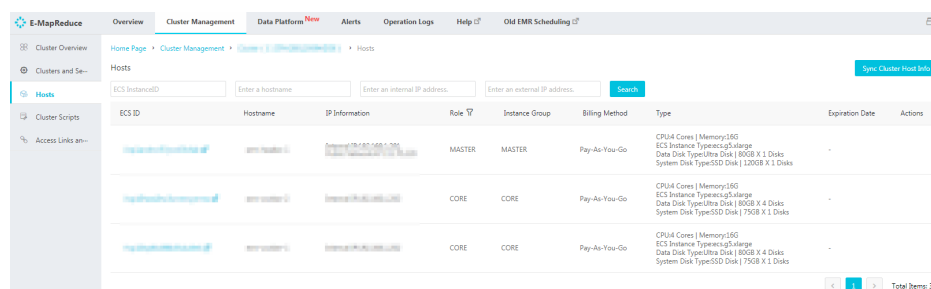
Note: Because the network environment of the DataWorks project is generally not connected to that of the Hadoop cluster core nodes, you can customize your resource groups to run the synchronization task of DataWorks on Hadoop cluster master nodes (this is because Hadoop cluster master and core nodes are often interconnected).

View core nodes of the EMR Hadoop cluster.

In the EMR console, at the top of the menu bar, click **Cluster Management**.

Locate your target cluster and click **Manage** at its right side.

In the left-side navigation pane, click **Hosts** to view the master nodes and core nodes, as shown in the following figure.



The master node name of a Non-HA EMR Hadoop cluster is generally **emr-header-1**, and the core node name is generally **emr-worker-X**.

Click the ECS ID of the master node in the preceding figure to go to its Instance Details page.

Click **Connect** to connect to the ECS instance.

Note: You can also run the `hadoop dfsadmin -report` command to view core node information.

Note: The ECS master node logon password is the password you set when you created your EMR Hadoop cluster.

```
DFS Remaining: 665931456512 (620.20 GB)
DFS Used: 209780736 (200.06 MB)
DFS Used%: 0.03%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0

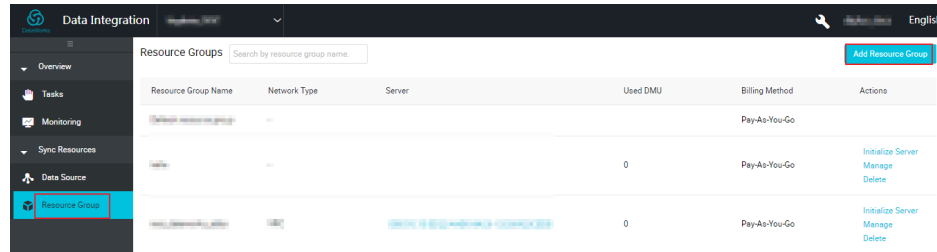
-----
Live datanodes (2):

Name: 192.168.1.206:50010 (emr-worker-2.cluster-77026)
Hostname: emr-worker-2.cluster-77026
Decommission Status : Normal
Configured Capacity: 333373341696 (310.48 GB)
DFS Used: 104890368 (100.03 MB)
Non DFS Used: 302723072 (288.70 MB)
DFS Remaining: 332965728256 (310.10 GB)
DFS Used%: 0.03%
DFS Remaining%: 99.88%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Sat Sep 29 17:37:46 CST 2018

Name: 192.168.1.205:50010 (emr-worker-1.cluster-77026)
Hostname: emr-worker-1.cluster-77026
Decommission Status : Normal
Configured Capacity: 333373341696 (310.48 GB)
DFS Used: 104890368 (100.03 MB)
Non DFS Used: 302723072 (288.70 MB)
DFS Remaining: 332965728256 (310.10 GB)
DFS Used%: 0.03%
DFS Remaining%: 99.88%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Sat Sep 29 17:37:46 CST 2018
```

Create a custom resource group

In the DataWorks console, go to the **Data Integration** page, select **Resource Group** -> **Add resources Group**. For more information about custom resource group, see [Add a scheduling resource](#).



Enter the name of the resource group and the server information. The server is the master node of your EMR cluster.

The 'Add Resource Group' dialog box shows a four-step process: Create Resource Group, Add Server, Install Agent, and Test Connectivity. The 'Add Server' step is active. It displays the 'Network Type' as VPC and lists fields for 'Server 1': ECS UUID, Server IP, Machine CPU (Cores), and Machine RAM (GB). Each field has a placeholder text and a help icon. An 'Add Server' button is at the bottom left, and 'Previous' and 'Next' buttons are at the bottom right.

Add Resource Group

Create Resource Group → **Add Server** → Install Agent → Test Connectivity

* Network Type: **VPC** ?

Server 1

* ECS UUID: Enter a UUID rather than server name. ?

* Server IP: Enter the internal IP address of the machine. ?

* Machine CPU (Cores):

* Machine RAM (GB):

Add Server

Previous Next

Network type is a **proprietary network (VPC)**.

For a VPC network, you must enter the UUID of your ECS instance.

For a Classic network, you must enter the instance name. Currently, only DataWorks 2.0 in the China East 2 (Shanghai) region supports adding a Classic network scheduling resource. For other regions, regardless of whether you are using a Classic network or VPC network, the network type must be selected as VPC network when you add a scheduling resource group.

ECS UUID: Log on to the EMR cluster master node and run `dmidecode | grep UUID` to obtain the returned value.

Machine IP: the public IP of the master node-**Machine CPU:** the CPU of the master node-**Memory size:** memory of the master node

You can obtain the preceding information from the configuration information section by clicking the master node ID in the ECS console.

After completing the **Add server** step, you must ensure that the networks of master node and DataWorks are interconnected. If you are using an ECS server, you need to set a server security group. If you are using a private IP, see **Add security group settings**. If you are using a public IP address, you can directly set the Internet ingress and egress under Security Group Rules. This example uses an EMR cluster in a VPC network that is in the same region as DataWorks, which means no security group needs to be set.

Install the agent as prompted. When the **available** status appears, it indicates that you successfully added a resource group.

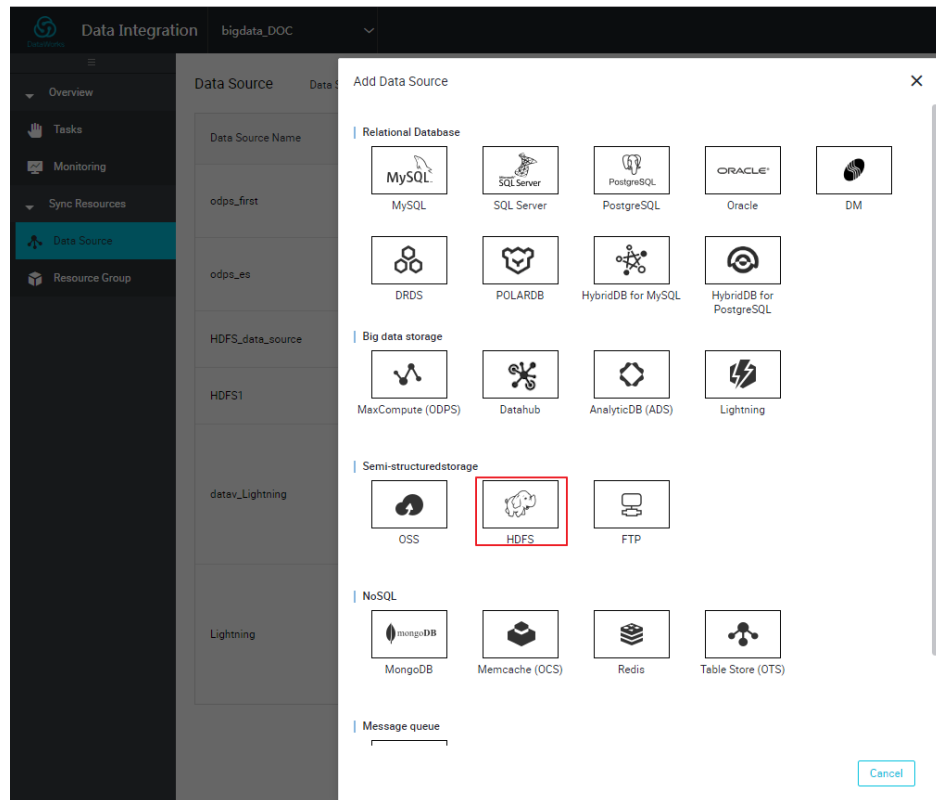
This example uses a VPC network, which means you do not need to open port 8000.

If the status is **unavailable**, log on to the master node and run the `tail -f /home/admin/alisa-tasknode/logs/heartbeat.log` command to check whether the heartbeat message between DataWorks and the master node has timed out.

```
[root@emr-header-1 logs]# hdfs dfs -ls /user/hive/warehouse/hive_doc_good_sale/
Found 1 items
drwxr-x--x  - hive hadoop          0 2018-09-03 17:46 /user/hive/warehouse/hive_doc_good_sale/pt=1
[root@emr-header-1 logs]# tail -f /home/admin/alisa-tasknode/logs/heartbeat.log
2018-09-06 21:47:34,448 INFO [pool-6-thread-1] [HeartbeatReporter.java:104] [] - heartbeat start, current status:2
2018-09-06 21:47:34,465 INFO [pool-6-thread-1] [HeartbeatReporter.java:133] [] - heartbeat end# cost time:0.025s
2018-09-06 21:47:39,465 INFO [pool-6-thread-1] [HeartbeatReporter.java:104] [] - heartbeat start, current status:2
2018-09-06 21:47:39,491 INFO [pool-6-thread-1] [HeartbeatReporter.java:133] [] - heartbeat end# cost time:0.026s
2018-09-06 21:47:44,491 INFO [pool-6-thread-1] [HeartbeatReporter.java:104] [] - heartbeat start, current status:2
2018-09-06 21:47:44,515 INFO [pool-6-thread-1] [HeartbeatReporter.java:133] [] - heartbeat end# cost time:0.024s
2018-09-06 21:47:49,516 INFO [pool-6-thread-1] [HeartbeatReporter.java:104] [] - heartbeat start, current status:2
2018-09-06 21:47:49,538 INFO [pool-6-thread-1] [HeartbeatReporter.java:133] [] - heartbeat end# cost time:0.022s
2018-09-06 21:47:54,539 INFO [pool-6-thread-1] [HeartbeatReporter.java:104] [] - heartbeat start, current status:2
2018-09-06 21:47:54,555 INFO [pool-6-thread-1] [HeartbeatReporter.java:133] [] - heartbeat end# cost time:0.016s
```

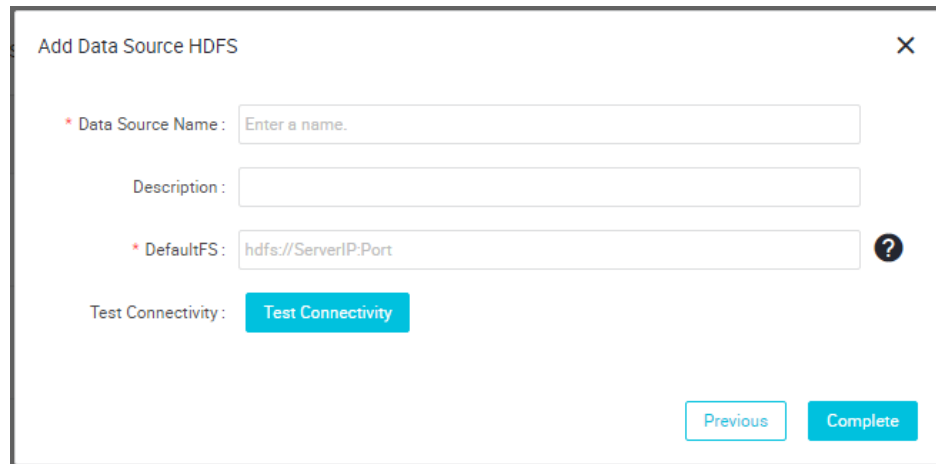
Create a data source

In the Data Integration page of DataWorks, click **Data Sources**->**New source**, and select **HDFS**.



In the **New HDFS Data Sources** panel, set the **Name** and **defaultFS** parameters.

For a EMR Hadoop cluster, if it is a non-HA cluster, the address is set to `hdfs://emr-header-1的IP:9000`. If it is an HA cluster, the address is set to `hdfs://emr-header-1的IP:8020`. In this example, `emr-header-1` and DataWorks are connected through a VPC network, so an intranet IP is set, and the test connectivity is unavailable.



Add Data Source HDFS

* Data Source Name:

Description:

* DefaultFS: ?

Test Connectivity:

Configure a data synchronization task

In the left-side navigation pane of the Data Integration page, click **Sync Tasks**, select **New->Script Mode**.

In the **Import template** panel, select the following data source type:

- a. After the template is imported, the synchronization task is converted to the script mode. The following figure shows the configuration script used in this topic. For more information, see [Script mode configuration](#). For information about Elasticsearch configuration rules, see [Configure Elasticsearch Writer](#).

```

1 {
2   "configuration": {
3     "reader": {
4       "plugin": "hdfs",
5       "parameter": {
6         "path": "/user/hive/warehouse/hive_esdoc_good_sale/",
7         "datasource": "HDFS_data_source",
8         "column": [
9           {
10            "index": 0,
11            "type": "string"
12          },
13          {
14            "index": 1,
15            "type": "string"
16          },
17          {
18            "index": 2,
19            "type": "string"
20          },
21          {
22            "index": 3,
23            "type": "string"
24          },
25          {
26            "index": 4,
27            "type": "long"
28          },
29          {
30            "index": 5,
31            "type": "double"
32          },
33          {
34            "index": 6,
35            "type": "long"
36          }
37        ],
38        "defaultFS": "hdfs://[redacted]:9000",
39        "fieldDelimiter": ",",
40        "encoding": "UTF-8",
41        "fileType": "text"
42      }
43    },
44    "writer": {
45      "plugin": "elasticsearch",
46      "parameter": {
47        "accessId": "[redacted]",
48        "endpoint": "http://es-cn-[redacted].com:9200",
49        "indexType": "elasticsearch",
50        "accessKey": "[redacted]",
51        "cleanup": true,
52        "discovery": false,
53        "column": [
54          {
55            "name": "create_time",
56            "type": "string"
57          },
58          {
59            "name": "category",
60            "type": "string"
61          },
62          {
63            "name": "brand",
64            "type": "string"
65          },
66          {
67            "name": "buyer_id",
68            "type": "string"
69          },
70          {
71            "name": "trans_num",
72            "type": "long"
73          },
74          {
75            "name": "trans_amount",
76            "type": "double"
77          },
78          {
79            "name": "click_cnt",
80            "type": "long"
81          }
82        ],
83        "index": "hive_doc_esgood_sale",
84        "batchSize": 1000,
85        "splitter": ",",
86      }
87    },
88    "setting": {
89      "errorLimit": {
90        "record": "1000"
91      },
92      "speed": {
93        "throttle": false,
94        "concurrent": 1,
95        "mbps": "1",
96        "dmu": 1
97      }
98    }
99  },
100  "type": "job",
101  "version": "1.0"
102 }

```

 Hdfs Reader

- i. The synchronization script configuration includes the following three parts: Reader, which is the configuration of the upstream data source (that is, the target cloud product for data synchronization); Writer, which is the configuration of your ES instance; and setting, which refers to synchronization configurations such as packet loss rate and maximum concurrency. In the script configuration, note the following:
 - i. The path parameter indicates the place where the data is stored in the Hadoop cluster. You can log on to the master node and run the `hdfs dfs -ls /user/hive/warehouse/hive_doc_good_sale` command to confirm the place. For a partition table, you do not need to specify the partitions. The data synchronization feature of DataWorks can automatically recurse to the partition path, as shown in the following figure.
 - ii. Because Elasticsearch does not support the timestamp type, the example used in this topic sets the type of the `creat_time` field to string.
- endpoint is the intranet or Internet IP address of your Elasticsearch instance. If you are using an intranet address, you need to add the IP into the Elasticsearch whitelist in the Elasticsearch cluster configuration page. If you are using an Internet IP, you need to configure the Elasticsearch public network access whitelist (including the server IP addresses of DataWorks and the IP of the resource group you use).
 - `accessId` and `accessKey` in Elasticsearch Writer are your Elasticsearch access user name (it is `elastic` by default) and password, respectively.
 - `index` is the index of your Elasticsearch instance through which you need to access Elasticsearch data.
 - When creating a synchronization task, in the default configuration script of DataWorks, the record field value of `errorLimit` is 0. You need to change the value to a larger number, such as 1,000.

After the preceding configurations are complete, in the upper right corner click **configuration tasks resources group**, and then click **run**.

If the prompt **Task run successfully** is displayed, it indicates that the task is synchronized successfully. If the task fails to run, copy the error logs for troubleshooting.

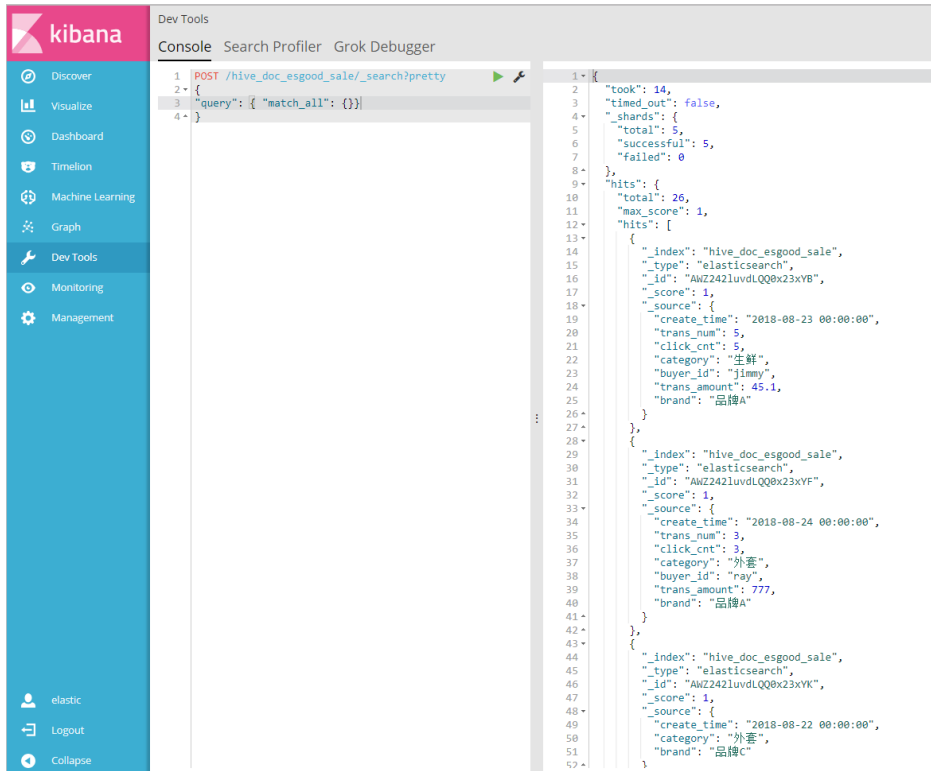
Verify the synchronization result

Go to the Elasticsearch console, click **Kibana console** in the upper right corner and then select **Dev Tools**.

Run the following command to view the synchronized data.

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

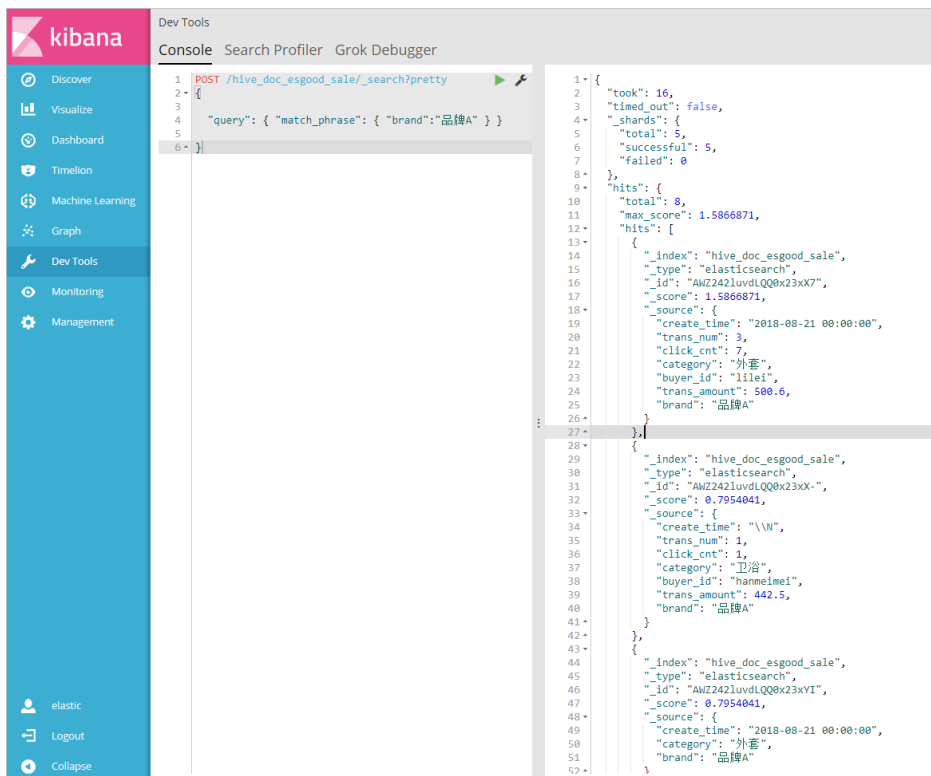
hive_doc_esgood_sale is the value of the index field when the data is synchronized.



Data query and analysis

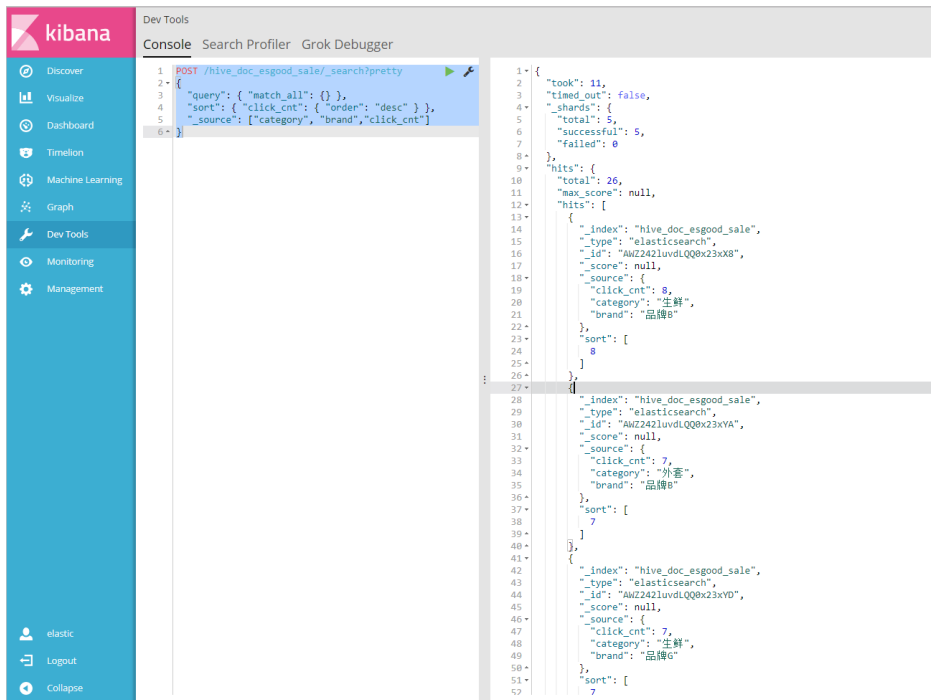
The following example returns all the documents of Brand A.

```
POST /hive_doc_esgood_sale/_search?pretty
{
  "query": { "match_phrase": { "brand": "Brand A" } }
}
```



The following example sorts various documents by **Clicks**, in order to view the popularity of all brands.

```
POST /hive_doc_esgood_sale/_search?pretty
{
  "query": { "match_all": {} },
  "sort": { "click_cnt": { "order": "desc" } },
  "_source": ["category", "brand", "click_cnt"]
}
```



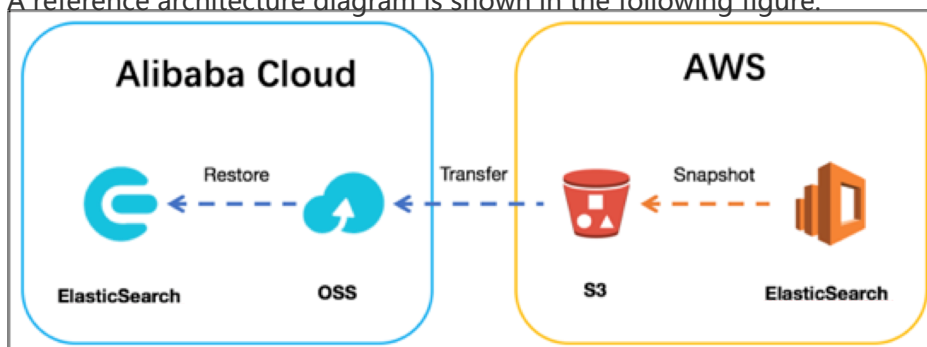
For more information about commands and access methods, see [Alibaba Cloud Elasticsearch documents](#) and [Elastic.co help center](#).

Elasticsearch index migration from AWS to Alibaba Cloud

Abstract

This document describes Elasticsearch (ES) index migration from AWS to Alibaba Cloud.

A reference architecture diagram is shown in the following figure:



Introduction

Concepts

- **Elasticsearch:** A distributed, RESTful search and analytics engine capable of solving a growing number of use cases. As the heart of the Elastic Stack, it centrally stores your data so you can discover the expected and uncover the unexpected.
- **Kibana:** Lets you visualize your Elasticsearch data and navigate the Elastic Stack.
- **Amazon Elasticsearch Service:** It's easy to deploy, secure, operate, and scale Elasticsearch for log analytics, full text search, application monitoring, and more. Amazon Elasticsearch Service is a fully managed service that delivers Elasticsearch' s easy-to-use APIs and real-time analytics capabilities alongside the availability, scalability, and security that production workloads require.
- **Alibaba Elasticsearch Service:** Alibaba Cloud' s Elasticsearch service. In this guide, we explain how to use Elasticsearch through our Alibaba Cloud China site. Have not onboard on International site.

Snapshot and Restore: You can store snapshots of individual indexes or an entire cluster in a remote repository like a shared file system, S3, or HDFS. These snapshots are great for backups because they can be restored relatively quickly. However, snapshots can only be restored to versions of Elasticsearch that can read the indexes:

- A snapshot of an index created in 5.x can be restored to 6.x.
- A snapshot of an index created in 2.x can be restored to 5.x.
- A snapshot of an index created in 1.x can be restored to 2.x.

Conversely, snapshots of indexes created in 1.x cannot be restored to 5.x or 6.x, and snapshots of indexes created in 2.x cannot be restored to 6.x.

Snapshots are incremental and can contain indexes created in various versions of Elasticsearch. If any indexes in a snapshot were created in an incompatible version, you will not be able restore the snapshot.

Solution overview

Elasticsearch (ES) indexes can be migrated with following steps:

Step 1: Create baseline indexes

1. Create a snapshot repository and associate it to an AWS S3 Bucket.

Create the first snapshot of the indexes to be migrated, which is a full snapshot.

The snapshot will be automatically stored in the AWS S3 bucket created in the first step.

Create an OSS Bucket on Alibaba Cloud, and register it to a snapshot repository of an Alibaba Cloud ES instance.

Use the OSSImport tool to pull the data from the AWS S3 bucket into the Alibaba Cloud OSS bucket.

Restore this full snapshot to the Alibaba Cloud ES instance.

Step 2: Periodic incremental snapshots

Repeat several incremental snapshot and restore.

Step 3: Final snapshot and service switchover

1. Stop services which can modify index data.
2. Create a final incremental snapshot of the AWS ES instance.
3. Transfer and restore the final incremental snapshot to an Alibaba Cloud ES instance.
4. Perform service switchover to the Alibaba Cloud ES instance.

Prerequisites

Elasticsearch service

- The version number of AWS ES is **5.5.2**, located in the Singapore region.
- The version number of Alibaba Cloud ES is **5.5.3**, located in Hangzhou.
- The demo index name is **movies**.

Manual Snapshot Prerequisites on AWS

Amazon ES takes daily automated snapshots of the primary index shards in a domain, and stores these automated snapshots in a preconfigured Amazon S3 bucket for 14 days at no additional charge to you. You can use these snapshots to restore the domain.

You cannot, however, use automated snapshots to migrate to new domains. Automated snapshots are read-only from within a given domain. **For migrations, you must use manual snapshots stored in your own repository (an S3 bucket).** Standard S3 charges apply for manual snapshots.

To create and restore index snapshots manually, you must work with IAM and Amazon S3. Verify that you have met the following prerequisites before you attempt to take a snapshot.

Prerequisite	Description
S3 bucket	Stores manual snapshots for your Amazon ES domain.
IAM role	Delegates permissions to Amazon

	Elasticsearch Service. The trust relationship for the role must specify Amazon Elasticsearch Service in the Principal statement. The IAM role also is required to register your snapshot repository with Amazon ES. Only IAM users with access to this role may register the snapshot repository.
IAM policy	Specifies the actions that Amazon S3 may perform with your S3 bucket. The policy must be attached to the IAM role that delegates permissions to Amazon Elasticsearch Service. The policy must specify an S3 bucket in a Resource statement.

S3 bucket

You need an S3 bucket to store manual snapshots. Make a note of its Amazon Resource Name (ARN). You need it for the following:

- Resource statement of the IAM policy that is attached to your IAM role.
- Python client that is used to register a snapshot repository.

The following example shows an ARN for an S3 bucket:

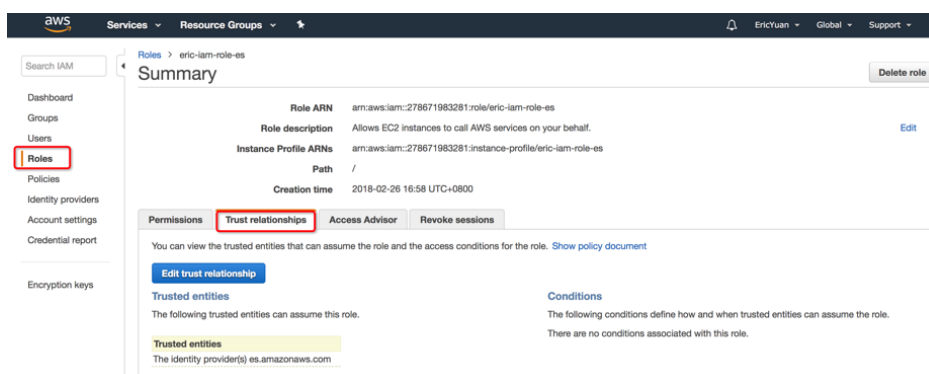
```
arn:aws:s3:::eric-es-index-backups
```

IAM role

You must have a role that specifies Amazon Elasticsearch Service, `es.amazonaws.com`, in a **ServiceStatement** in its trust relationship, as shown in the following example:

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

In the AWS IAM Console, you can find Trust Relationship details here:



Edit Trust Relationship

You can customize trust relationships by editing the following access control policy document.

Policy Document

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "",
6       "Effect": "Allow",
7       "Principal": {
8         "Service": "es.amazonaws.com"
9       },
10      "Action": "sts:AssumeRole"
11    }
12  ]
13 }
```

When you create an AWS service role by using the IAM Console, Amazon ES is not included in the **Select role type** list. However, you can still create the role by choosing **Amazon EC2**, following the steps to create the role, and then editing the role's trust relationships to `es.amazonaws.com` instead of `ec2.amazonaws.com`.

IAM Policy

You must attach an **IAM policy** to the **IAM role**. The policy specifies the S3 bucket that is used to store manual snapshots for 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/*"
  ]
}
]
}

```

You need to paste it in here:

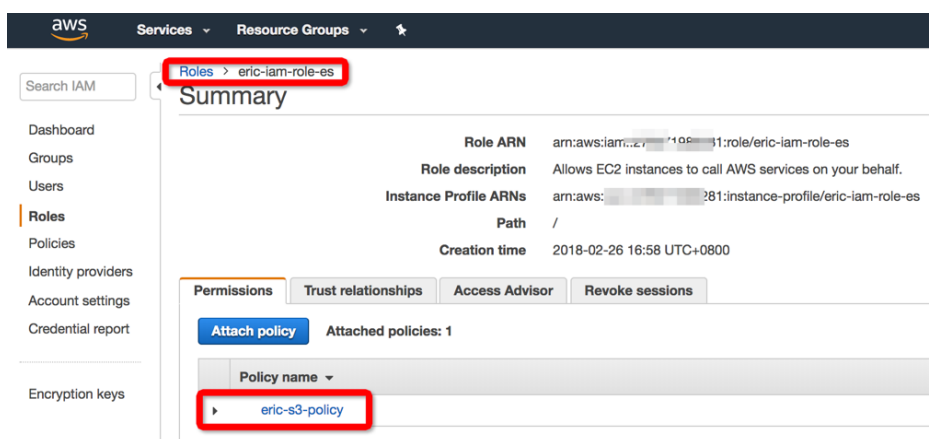
The screenshot shows the AWS IAM console interface. In the left-hand navigation menu, the 'Policies' link is highlighted with a red box. In the main content area, the breadcrumb 'Policies > eric-s3-policy' is also highlighted with a red box. Below the breadcrumb, the 'Summary' tab is selected, and the 'Permissions' sub-tab is active, also highlighted with a red box. The 'Policy summary' section shows the JSON policy document with line numbers 1 through 25. The policy allows 's3:ListBucket' and 's3:GetObject', 's3:PutObject', 's3:DeleteObject' actions on the resource 'arn:aws:s3:::eric-es-index-backups/*'.

You can make sure the policy is correct by looking at the **Policy summary** as follows:

The screenshot shows the 'Permissions' tab of the 'Policy summary' section. A red box highlights the table below the 'Policy summary' and 'JSON' tabs. The table has columns for 'Service', 'Access level', and 'Resource'. It shows that the policy allows access to S3 services with limited permissions (List, Read, Write) on multiple resources.

Service	Access level	Resource
S3	Limited: List, Read, Write	Multiple

Attach IAM Policy to IAM Role



Registering a manual snapshot directory

You must register the snapshot directory with Amazon Elasticsearch Service before you can take manual index snapshots. This one-time operation requires that you sign your AWS request with credentials for one of the users or roles specified in the IAM role's trust relationship, as described in Section **Manual snapshot prerequisites on AWS**.

You can't use curl to perform this operation because it doesn't support AWS request signing. Instead, use the sample Python client to register your snapshot directory.

Modify sample python client

Download a copy of the file "Sample Python Client.docx", then modify the values in yellow in that document to match your real values. Copy the contents of "Sample Python Client.docx" into a Python file called "snapshot.py" after you have finished editing.

Sample Python Client.docx

Variable name	Description
region	AWS Region where you created the snapshot repository
host	Endpoint for your Amazon ES domain
aws_access_key_id	IAM credential
aws_secret_access_key	IAM credential
path	Name of the snapshot repository
data: bucket; region; role_arn	Must include the name of the S3 bucket and the ARN for the IAM role that you created in Section Manual snapshot prerequisites on AWS . To enable server-side encryption with S3-managed keys for the snapshot repository, add "server_side_encryption": true to the settings JSON. Important

If the S3 bucket is in the **us-east-1** region, you need to use "endpoint": "s3.amazonaws.com" in place of "region": "us-east-1".

Install Amazon Web Services Library boto-2.48.0

This sample Python client requires that you install version 2.x of the **boto** package 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
```

Execute Python client to register snapshot directory

```
# python snapshot.py
```

Registering Snapshot Repository

Check result in **Kibana**->**Dev Tools** with request:

GET _snapshot



Snapshot and restore for the first time

Take a snapshot manually on AWS ES

The following commands are all performed on **Kibana**->**Dev Tools**, you can also perform them using **curl** from the Linux or Mac OSX command line.

- Take a snapshot with the name *snapshot_movies_1* only for the index **movies** in the repository eric-snapshot-repository.

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

- Check snapshot status

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

The screenshot shows the Kibana Dev Tools console. On the left, the console log shows the following commands:

```
1 PUT _snapshot/eric-snapshot-repository/snapshot_movies_1
2 {
3   "indices": "movies"
4 }
5 GET _snapshot/eric-snapshot-repository/snapshot_movies_1
6
7 PUT _snapshot/eric-snapshot-repository/snapshot_movies_2
8
9 delete movies
10
11 GET _search
12 {
13   "query": {
14     "match_all": {}
15   }
16 }
17
18 GET _snapshot
19 GET _snapshot/eric-snapshot-repository
20 GET _snapshot/eric-snapshot-repository/snapshot_movies_1
21 GET _snapshot/eric-snapshot-repository/snapshot_movies_2
```

On the right, the JSON response for the GET command is displayed:

```
{
  "snapshots": [
    {
      "snapshot": "snapshot_movies_1",
      "uuid": "8lgKLvgoSpSgwBhbD4hTWg",
      "version_id": "5050299",
      "version": "5.5.2",
      "indices": [
        "movies"
      ],
      "state": "SUCCESS",
      "start_time": "2018-02-28T03:00:44.591Z",
      "start_time_in_millis": 1519786844591,
      "end_time": "2018-02-28T03:00:46.236Z",
      "end_time_in_millis": 1519786846236,
      "duration_in_millis": 1645,
      "failures": [],
      "shards": {
        "total": 5,
        "failed": 0,
        "successful": 5
      }
    }
  ]
}
```

- Check snapshot files on the AWS S3 console

The screenshot shows the AWS S3 console for the bucket 'eric-es-index-backups'. The 'Overview' tab is selected. The table below lists the objects in the bucket:

Name	Last modified	Size	Storage class
indices	--	--	--
incompatible-snapshots	Feb 28, 2018 11:00:47 AM GMT+0800	29.0 B	Standard
index-0	Feb 28, 2018 11:00:47 AM GMT+0800	178.0 B	Standard
index.latest	Feb 28, 2018 11:00:47 AM GMT+0800	8.0 B	Standard
meta-BigKLvgoSpSgwBhbD4hTWg.dat	Feb 28, 2018 11:00:45 AM GMT+0800	337.0 B	Standard
snap-BigKLvgoSpSgwBhbD4hTWg.dat	Feb 28, 2018 11:00:47 AM GMT+0800	228.0 B	Standard

Pull snapshot data from AWS S3 to Alibaba Cloud OSS

In this step, you need to pull snapshot data from your AWS S3 bucket into Alibaba Cloud OSS. For more information, see [Migrate data from Amazon S3 to Alibaba Cloud OSS](#).

After data transfer, check stored snapshot data from the OSS console:

eric-oss-aws-es-snapshot-s3				Access Control List (ACL)	Private	Type	Standard	Region	China (Hangzhou)	Created At	02/21
Overview Files Basic Settings Domain Names Image Processing Event Notification Function Compute Intelligent Media Management											
Log Overview Basic Statistics Ranking Statistics API Statistics Object Access Statistics											
Upload Create Folder Fragments Authorize Batch operation Refresh											
<input type="checkbox"/>	File/Object Name	Size	Storage Class	Updated At							
<input type="checkbox"/>	indices/										
<input type="checkbox"/>	incompatible-snapshots	0.028KB		2018-02-28 11:06							
<input type="checkbox"/>	index-0	0.174KB		2018-02-28 11:06							
<input type="checkbox"/>	index.latest	0.0080KB		2018-02-28 11:06							
<input type="checkbox"/>	meta-BlgKLvgoSp8gwBhbD4hTWg.dat	0.328KB		2018-02-28 11:06							
<input type="checkbox"/>	snap-BlgKLvgoSp8gwBhbD4hTWg.dat	0.223KB		2018-02-28 11:06							

Restore snapshot to an Alibaba Cloud ES instance

Create snapshot repository

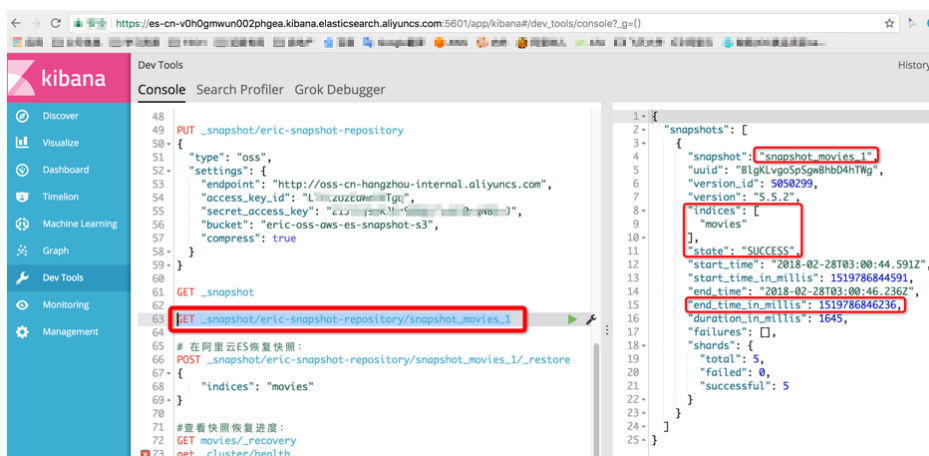
Perform the following request on **Kibana**->**Dev Tools** to create a snapshot repository with the same name: modify values as follows to match your real values.

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



After creating the snapshot directory, check the snapshot status for the snapshot named **snapshot_movies_1**, which was assigned in AWS ES manual snapshot step.

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



Note: Please record the start time and end time of this snapshot operation: It will be used when you transfer incremental snapshot data with the Alibaba Cloud OSSimport tool. For example:

"start_time_in_millis" : 1519786844591

"end_time_in_millis" : 1519786846236

Restore snapshots

Perform the following request on **Kibana->Dev Tools**.

POST _snapshot/eric-snapshot-repository/snapshot_movies_1/_restore

```

{
  "indexes": "movies"
}
  
```

GET movies/_recovery



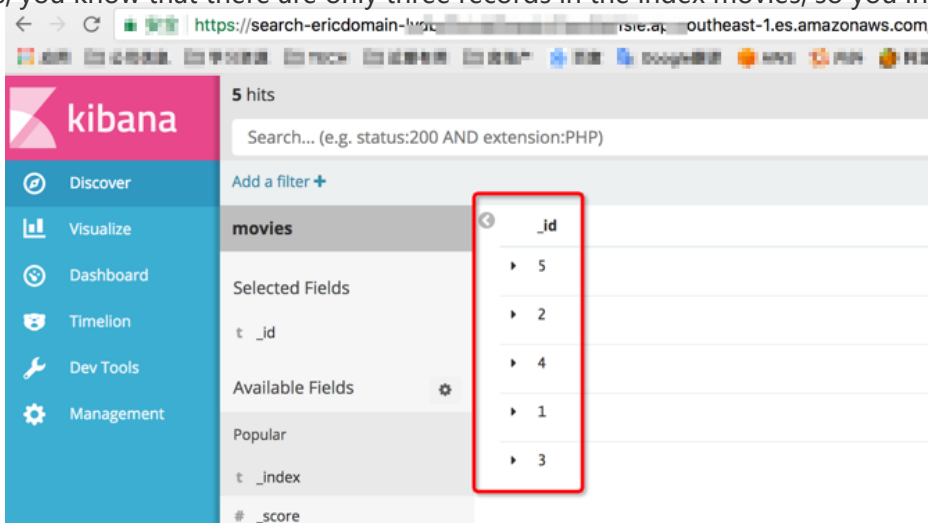
Check the availability of index movies on **Kibana->Dev Tools**, you can see there exist three records in the index movies, the number of records on your AWS ES instance.



Snapshot and restore for the last time

Create some sample data on AWS ES index movies

In the previous steps, you know that there are only three records in the index movies, so you insert



another two records.

You could also see the number of indexes using this request: GET movies/_count.



Take another snapshot manually

See section [Take a snapshot manually on AWS ES](#), then check the snapshot status:

```

1 get movies/_count
2
3 PUT _snapshot/eric-snapshot-repository/snapshot_movies_1
4 {
5   "indices": "movies"
6 }
7
8 GET _snapshot/eric-snapshot-repository/snapshot_movies_1
9
10 PUT _snapshot/eric-snapshot-repository/snapshot_movies_2
11 {
12   "indices": "movies"
13 }
14
15 GET _snapshot/eric-snapshot-repository/snapshot_movies_2
16
17
18 delete movies
19
20 GET _snapshot
21
22 GET _snapshot/eric-snapshot-repository
23 GET _snapshot/eric-snapshot-repository/snapshot_movies_1
24 GET _snapshot/eric-snapshot-repository/snapshot_movies_2
25
26 delete _snapshot/eric-es-index-backups
27 delete _snapshot/eric-snapshot-repository/snapshot_movies_1
28 delete _snapshot/eric-snapshot-repository/snapshot_movies_2

```

```

1 {
2   "snapshots": [
3     {
4       "snapshot": "snapshot_movies_2",
5       "uuid": "CWhIF7ShQZaKQJasPE70A",
6       "version_id": 5050299,
7       "version": "5.5.2",
8       "indices": [
9         "movies"
10      ],
11      "state": "SUCCESS",
12      "start_time": "2018-02-28T03:55:33.505Z",
13      "start_time_in_millis": 1519790133505,
14      "end_time": "2018-02-28T03:55:35.195Z",
15      "end_time_in_millis": 1519790135195,
16      "duration_in_millis": 1690,
17      "failures": [],
18      "shards": {
19        "total": 5,
20        "failed": 0,
21        "successful": 5
22      }
23    }
24  ]
25 }

```

Check the files listed in the S3 bucket:

Viewing 1 to 9				
<input type="checkbox"/> Name	Last modified	Size	Storage class	
<input type="checkbox"/> indices	--	--	--	
<input type="checkbox"/> snap-CWhIF7ShQZaKQJasPE70A.dat	Feb 28, 2018 11:55:36 AM GMT+0800	228.0 B	Standard	
<input type="checkbox"/> index.latest	Feb 28, 2018 11:55:36 AM GMT+0800	8.0 B	Standard	
<input type="checkbox"/> index-1	Feb 28, 2018 11:55:36 AM GMT+0800	274.0 B	Standard	
<input type="checkbox"/> meta-CWhIF7ShQZaKQJasPE70A.dat	Feb 28, 2018 11:55:34 AM GMT+0800	337.0 B	Standard	
<input type="checkbox"/> snap-BIqKLvgoSpSgwBhbD4hTWg.dat	Feb 28, 2018 11:00:47 AM GMT+0800	228.0 B	Standard	
<input type="checkbox"/> index-0	Feb 28, 2018 11:00:47 AM GMT+0800	178.0 B	Standard	
<input type="checkbox"/> incompatible-snapshots	Feb 28, 2018 11:00:47 AM GMT+0800	29.0 B	Standard	
<input type="checkbox"/> meta-BIqKLvgoSpSgwBhbD4hTWg.dat	Feb 28, 2018 11:00:45 AM GMT+0800	337.0 B	Standard	

If you check the folder **indexes**, you can also find some differences.

Pull incremental snapshot data from AWS S3 to Alibaba Cloud OSS

You can use the OSSImport tool to migrate data from S3 to OSS. Because there are 2 snapshot files stored in our S3 bucket now, we try to migrate only new files by modifying the value of **isSkipExistFile** in the configuration file **local_job.cfg**.

Filed	Meaning	Description
isSkipExistFile	Whether to skip the existing objects during data migration, a Boolean value.	<p>If it is set to true, the objects are skipped according to the size and LastModifiedTime.</p> <p>If it is set to false, the existing objects are overwritten. The default value is false. This option is invalid when jobType is set to audit.</p>

After the OSS Import migration job completes, you can see only 'new' files are migrated to OSS.

In your Alibaba Cloud OSS bucket:

eric-oss-aws-es-snapshot-s3 Access Control List (ACL) Private Type Standard Region China (Hangzhou) Created At 02/

Overview | **Files** | Basic Settings | Domain Names | Image Processing | Event Notification | Function Compute | Intelligent Media Manager

Log Overview | Basic Statistics | Ranking Statistics | API Statistics | Object Access Statistics

Upload Create Folder Fragments Authorize Batch operation Refresh

<input type="checkbox"/>	File/Object Name	Size	Storage Class	Updated At
<input type="checkbox"/>	indices/			
<input type="checkbox"/>	incompatible-snapshots	0.028KB		2018-02-28 11:06
<input type="checkbox"/>	index-0	0.174KB		2018-02-28 11:06
<input type="checkbox"/>	index-1	0.268KB		2018-02-28 14:06
<input type="checkbox"/>	index.latest	0.0080KB		2018-02-28 14:07
<input type="checkbox"/>	meta-BlgKLvgoSpSgwBhbD4hTWg.dat	0.329KB		2018-02-28 11:06
<input type="checkbox"/>	meta-CWhIF7ShQZaKQLJasPE70A.dat	0.329KB		2018-02-28 14:07
<input type="checkbox"/>	snap-BlgKLvgoSpSgwBhbD4hTWg.dat	0.223KB		2018-02-28 11:06
<input type="checkbox"/>	snap-CWhIF7ShQZaKQLJasPE70A.dat	0.223KB		2018-02-28 14:07

In our AWS S3 bucket:

Viewing 1 to 9

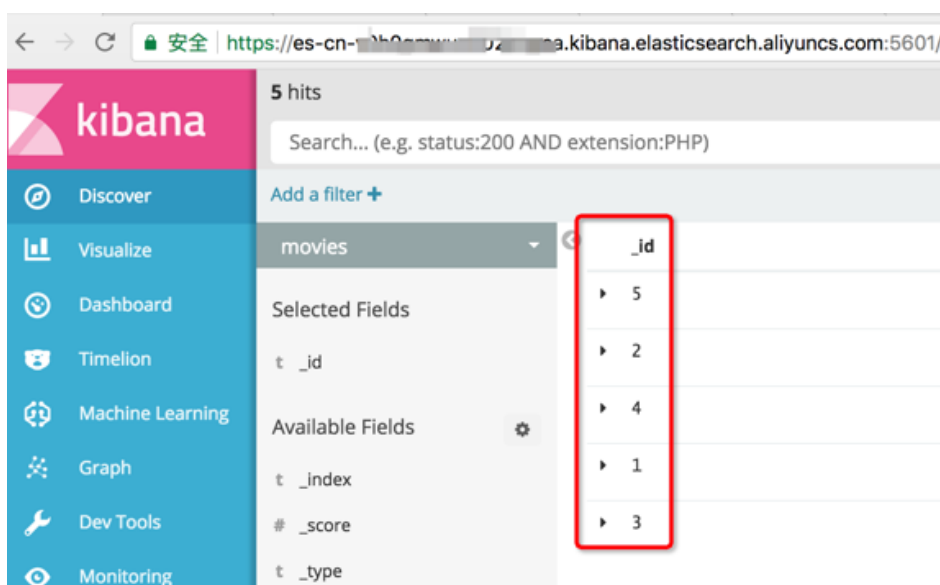
<input type="checkbox"/>	Name	Last modified	Size	Storage class
<input type="checkbox"/>	indices	--	--	--
<input type="checkbox"/>	incompatible-snapshots	Feb 28, 2018 11:00:47 AM GMT+0800	29.0 B	Standard
<input type="checkbox"/>	index-0	Feb 28, 2018 11:00:47 AM GMT+0800	178.0 B	Standard
<input type="checkbox"/>	index-1	Feb 28, 2018 11:55:36 AM GMT+0800	274.0 B	Standard
<input type="checkbox"/>	index.latest	Feb 28, 2018 11:55:36 AM GMT+0800	8.0 B	Standard
<input type="checkbox"/>	meta-BlgKLvgoSpSgwBhbD4hTWg.dat	Feb 28, 2018 11:00:45 AM GMT+0800	337.0 B	Standard
<input type="checkbox"/>	meta-CWhIF7ShQZaKQLJasPE70A.dat	Feb 28, 2018 11:55:34 AM GMT+0800	337.0 B	Standard
<input type="checkbox"/>	snap-BlgKLvgoSpSgwBhbD4hTWg.dat	Feb 28, 2018 11:00:47 AM GMT+0800	228.0 B	Standard
<input type="checkbox"/>	snap-CWhIF7ShQZaKQLJasPE70A.dat	Feb 28, 2018 11:55:36 AM GMT+0800	228.0 B	Standard

Restore an incremental snapshot

You can follow along with the steps from Section **Restore snapshots**, but the index **movies** needs to be closed firstly, then you have to restore the snapshot, and open the index again after restore:

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

After the restore procedure completes, you can see the count (5) of documents in the index **movies** is the same as it is in our AWS ES instance.



Conclusion

It is possible to migrate AWS Elasticsearch service data to Alibaba Cloud's Elasticsearch service by the snapshot and restore method.

This solution requires that the AWS ES instance is stopped first to prevent writes and requests during migration.

Further reading:

- <https://www.elastic.co/products/elasticsearch>
- <https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-snapshots.html>
- <https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/es-manageddomains-snapshots.html>
- <https://www.alibabacloud.com/help/doc-detail/64919.htm?spm=a3c0i.l31815en.b99.105.67c25139Y3tgnc>
- <https://github.com/zhichen/elasticsearch-repository-oss/wiki/OSS%E5%BF%AB%E7%85%A7%E8%BF%81%E7%A7%BB?spm=a2c4g.11186623.2.3.2acd85>
- <https://github.com/zhichen/elasticsearch-repository-oss/wiki/OSS%E5%BF%AB%E7%85%A7%E8%BF%81%E7%A7%BB?spm=a2c4g.11186623.2.3.wfCX30>
- <https://www.alibabacloud.com/help/doc-detail/56990.htm?spm=a3c0i.o56990en.a1.4.5baa6605O1C9yZ>

Data interconnection between ES-Hadoop

and Elasticsearch

You can directly write data to Alibaba Cloud Elasticsearch through ES-Hadoop based on Alibaba Cloud Elasticsearch and E-MapReduce. To complete this task, follow these steps:

Activate Alibaba Cloud Elasticsearch

This example uses the following Alibaba Cloud services:

- **VPC:** Transmitting data in a public network is not secure. To ensure a secure connection to your Alibaba Cloud Elasticsearch instances, you must deploy a VPC and a VSwitch in the specified region. Therefore, you must activate VPC.
- **OSS:** In this example, OSS is used to store the E-MapReduce log. You must activate OSS and create a bucket before you activate E-MapReduce.
- Elasticsearch
- E-MapReduce

Follow these steps to activate the corresponding Alibaba Cloud services:

Activate Alibaba Cloud VPC

- On the Alibaba Cloud website, choose Products > Networking > Virtual Private Cloud, and then click Activate Now.
- Log on to the VPC console, and click Create VPC to create a VPC.
- You can manage the VPC that you have created in the console.

Note:

For more information about Alibaba Cloud VPC, see [Virtual Private Cloud \(VPC\)](#).

Activate Alibaba Cloud Object Storage Service

- Log on to the Alibaba Cloud console, choose Product > Storage & CDN > Object Storage Service, and click Buy Now.
- Log on to the OSS console, click Create Bucket to create a bucket.

Note:

You must create the bucket in the same region where the E-MapReduce cluster is created. This example chooses the China (Hangzhou) region.

- iii. Create a bucket according to the instructions displayed on the page.

Activate Alibaba Cloud Elasticsearch

- i. On the Alibaba Cloud website, choose Product > Analytics & Big Data > Elasticsearch. The product page is displayed.

Note:

You can get a 30-day free trial.

- ii. After you have successfully activated Elasticsearch, you can view the newly created Elasticsearch instances in the Elasticsearch console.

Activate Alibaba Cloud E-MapReduce

- i. On the Alibaba Cloud website, choose Product > Analytics & Big Data > E-MapReduce. The product page is displayed.
- ii. Click **Buy Now**, and complete the configuration.
- iii. You can view the E-MapReduce clusters that you have created in the cluster list, and perform the following operations to verify the creation status.
 - i. You can remotely log on to the clusters through a public IP address:

```
ssh root@your public IP address
```

-Run the jps command to view background processes:

```
[root@emr-header-1 ~]# jps
16640 Bootstrap
17988 RunJar
19140 HistoryServer
18981 WebAppProxyServer
14023 Jps
15949 gateway.jar
16621 ZeppelinServer
1133 EmrAgent
15119 RunJar
17519 ResourceManager
1871 Application
19316 JobHistoryServer
1077 WatchDog
17237 SecondaryNameNode
16502 NameNode
16988 ApacheDsTanukiWrapper
18429 ApplicationHistoryServer
```

Create an MR job that writes data to Elasticsearch from E-MapReduce

We recommend that you use Maven to manage projects. To use Maven, follow these steps:

Install Maven.

Make sure that your computer has Maven installed.

Generate an engineering framework.

Run the following command in the root directory of the project:

```
mvn archetype:generate -DgroupId=com.aliyun.emrtoes -DartifactId=emrtoes -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

Maven will automatically generate an empty sample project named emrtoes, which is the same as the specified artifactId. The project contains a pom.xml file and an application class. The path of the class package is the same as the specified groupId.

Add Hadoop and ES-Hadoop dependencies.

Open the project with an IDE and edit the pom.xml file. Add the following content to the dependencies:

```
<dependency>
<groupId>org.apache.hadoop</groupId>
<artifactId>hadoop-mapreduce-client-common</artifactId>
<version>2.7.3</version>
</dependency>
<dependency>
<groupId>org.apache.hadoop</groupId>
<artifactId>hadoop-common</artifactId>
<version>2.7.3</version>
</dependency>
<dependency>
<groupId>org.elasticsearch</groupId>
<artifactId>elasticsearch-hadoop-mr</artifactId>
<version>5.5.3</version>
</dependency>
```

Add the packaging plugin.

Since a third-party database is used, you must package this database into a JAR package. Add the following maven-assembly-plugin coordinates to the pom.xml file:

```

    <plugins>
    <plugin>
    <artifactId>maven-assembly-plugin</artifactId>
    <configuration>
    <archive>
    <manifest>
    <mainClass>com.aliyun.emrtoes.EmrToES</mainClass>
    </manifest>
    </archive>
    <descriptorRefs>
    <descriptorRef>jar-with-dependencies</descriptorRef>
    </descriptorRefs>
    </configuration>
    <executions>
    <execution>
    <id>make-assembly</id>
    <phase>package</phase>
    <goals>
    <goal>single</goal>
    </goals>
    </execution>
    </executions>
    </plugin>

    <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-shade-plugin</artifactId>
    <version>3.1.0</version>
    <executions>
    <execution>
    <phase>package</phase>
    <goals>
    <goal>shade</goal>
    </goals>
    <configuration>
    <transformers>
    <transformer
    implementation="org.apache.maven.plugins.shade.resource.ApacheLicenseResourceTransformer">
    </transformer>
    </transformers>
    </configuration>
    </execution>
    </executions>
    </plugin>
  </plugins>

```

Write code.

Add a new class EmrToES.java that is parallel to the application class to the com.aliyun.emrtoes package. Add the following content:

```
package com.aliyun.emrtoes;
```

```
import org.apache.hadoop.conf.Configuration;
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 java.io.IOException;

public class EmrToES {

    public static class MyMapper extends Mapper<Object, Text, NullWritable, Text> {
        private Text line = new Text();

        @Override
        protected void map(Object key, Text value, Context context)
            throws IOException, InterruptedException {
            if (value.getLength() > 0) {
                line.set(value);
                context.write(NullWritable.get(), line);
            }
        }
    }

    public static void main(String[] args) throws IOException, ClassNotFoundException, InterruptedException {
        Configuration conf = new Configuration();
        String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();

        //Alibaba Cloud Elasticsearch X-PACK username and password
        conf.set("es.net.http.auth.user", "X-PACK username");
        conf.set("es.net.http.auth.pass", "X-PACK password");

        conf.setBoolean("mapred.map.tasks.speculative.execution", false);
        conf.setBoolean("mapred.reduce.tasks.speculative.execution", false);
        conf.set("es.nodes", "The private address of your Elasticsearch instance");
        conf.set("es.port", "9200");
        conf.set("es.nodes.wan.only", "true");
        conf.set("es.resource", "blog/yunqi");
        conf.set("es.mapping.id", "id");
        conf.set("es.input.json", "yes");

        Job job = Job.getInstance(conf, "EmrToES");
        job.setJarByClass(EmrToES.class);

        job.setMapperClass(MyMapper.class);
        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(EsOutputFormat.class);
        job.setMapOutputKeyClass(NullWritable.class);
        job.setMapOutputValueClass(Text.class);

        FileInputFormat.setInputPaths(job, new Path(otherArgs[0]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

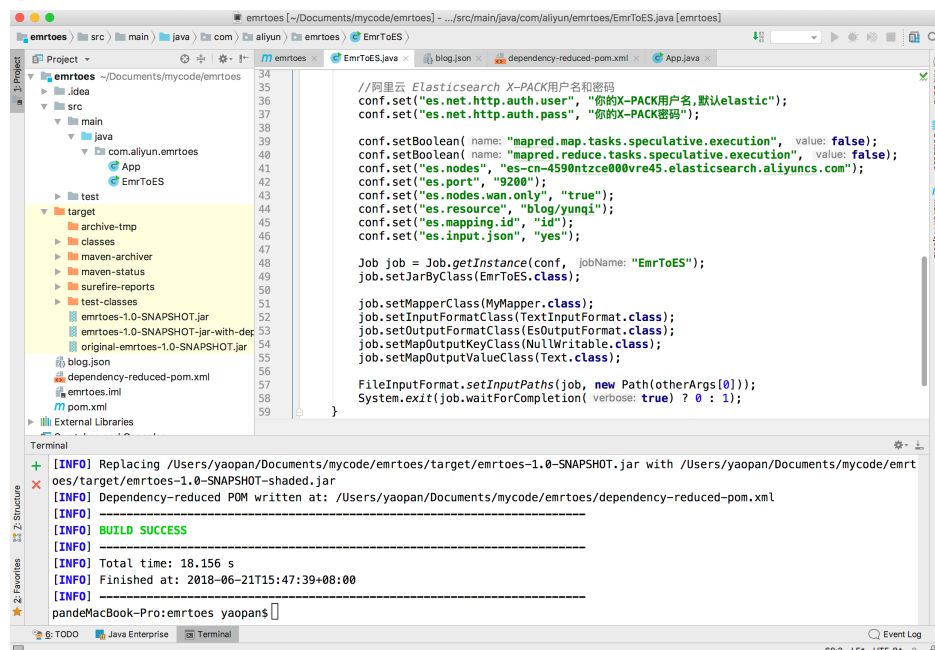
```
}
```

Compile and package.

Run the following command in the project directory:

```
mvn clean package
```

After you have run the command, you can view the JAR package named `emrtoes-1.0-SNAPSHOT-jar-with-dependencies.jar` of the job in the target directory of the project.



Complete the job in E-MapReduce

Test the data

Write the following data to `blog.json`:

```
{
  "id": "1",
  "title": "git introduction",
  "posttime": "2016-06-11",
  "content": "The main difference between svn and git..."
},
{
  "id": "2",
  "title": "Introduction and simple use of Java Generics",
  "posttime": "2016-06-12",
  "content": "Basic operations: CRUD..."
},
{
  "id": "3",
  "title": "Basic operations of SQL",
  "posttime": "2016-06-13",
  "content": "The main difference between svn and git..."
},
{
  "id": "4",
  "title": "Basic Hibernate framework",
  "posttime": "2016-06-14",
  "content": "Basic Hibernate framework..."
},
{
  "id": "5",
  "title": "Basics of Shell",
  "posttime": "2016-06-15",
  "content": "What is Shell?..."
}
```

Run the following scp remote copy command to upload the file to the Alibaba Cloud EMR cluster:

```
scp blog.json root@your EIP:/root
```

Upload blog.json to HDFS:

```
hadoop fs -mkdir /work
hadoop fs -put blog.json /work
```

Upload the JAR package

Upload the JAR package stored in the target directory of the Maven project to the Alibaba Cloud EMR cluster:

```
scp target/emrtoes-1.0-SNAPSHOT-jar-with-dependencies.jar root@YourIP:/root
```

Execute the MR job

Run the following command:

```
hadoop jar emrtoes-1.0-SNAPSHOT-jar-with-dependencies.jar /work/blog.json
```

If the job is successfully executed, the following message is displayed in the console:

```

1. root@emr-header-1:~ (ssh)
[root@emr-header-1 ~]# hadoop jar emrtoes-1.0-SNAPSHOT-jar-with-dependencies.jar /work/blog.json
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/apps/ecm/service/hadoop/2.7.2-1.2.11/package/hadoop-2.7.2-1.2.11/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/apps/ecm/service/tez/0.8.4/package/tez-0.8.4/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
18/06/21 15:53:18 INFO impl.TimelineClientImpl: Timeline service address: http://emr-header-1.cluster-67561:8188/ws/v1/timeline/
18/06/21 15:53:18 INFO client.RMProxy: Connecting to ResourceManager at emr-header-1.cluster-67561/192.168.0.103:8032
18/06/21 15:53:19 INFO input.FileInputFormat: Total input paths to process : 1
18/06/21 15:53:19 INFO mapreduce.JobSubmitter: number of splits:1
18/06/21 15:53:19 INFO Configuration.deprecation: mapred.reduce.tasks.speculative.execution is deprecated. Instead, use mapreduce.reduce.speculative
18/06/21 15:53:19 INFO Configuration.deprecation: mapred.map.tasks.speculative.execution is deprecated. Instead, use mapreduce.map.speculative
18/06/21 15:53:19 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1529566866753_0001
18/06/21 15:53:19 INFO impl.YarnClientImpl: Submitted application application_1529566866753_0001
18/06/21 15:53:20 INFO mapreduce.Job: The url to track the job: http://emr-header-1.cluster-67561:20888/proxy/application_1529566866753_0001/
18/06/21 15:53:20 INFO mapreduce.Job: Running job: job_1529566866753_0001
18/06/21 15:53:28 INFO mapreduce.Job: Job job_1529566866753_0001 running in uber mode : false
18/06/21 15:53:28 INFO mapreduce.Job: map 0% reduce 0%
18/06/21 15:53:34 INFO mapreduce.Job: map 100% reduce 0%
18/06/21 15:53:40 INFO mapreduce.Job: map 100% reduce 14%
18/06/21 15:53:41 INFO mapreduce.Job: map 100% reduce 57%
18/06/21 15:53:42 INFO mapreduce.Job: map 100% reduce 71%
18/06/21 15:53:43 INFO mapreduce.Job: map 100% reduce 86%
18/06/21 15:53:44 INFO mapreduce.Job: map 100% reduce 100%
18/06/21 15:53:44 INFO mapreduce.Job: Job job_1529566866753_0001 completed successfully
18/06/21 15:53:44 INFO mapreduce.Job: Counters: 66
    File System Counters
      FILE: Number of bytes read=412
      FILE: Number of bytes written=1024771
      FILE: Number of read operations=0
      FILE: Number of large read operations=0
      FILE: Number of write operations=0
      HDFS: Number of bytes read=635
      HDFS: Number of bytes written=0
      HDFS: Number of read operations=2
      HDFS: Number of large read operations=0
      HDFS: Number of write operations=0

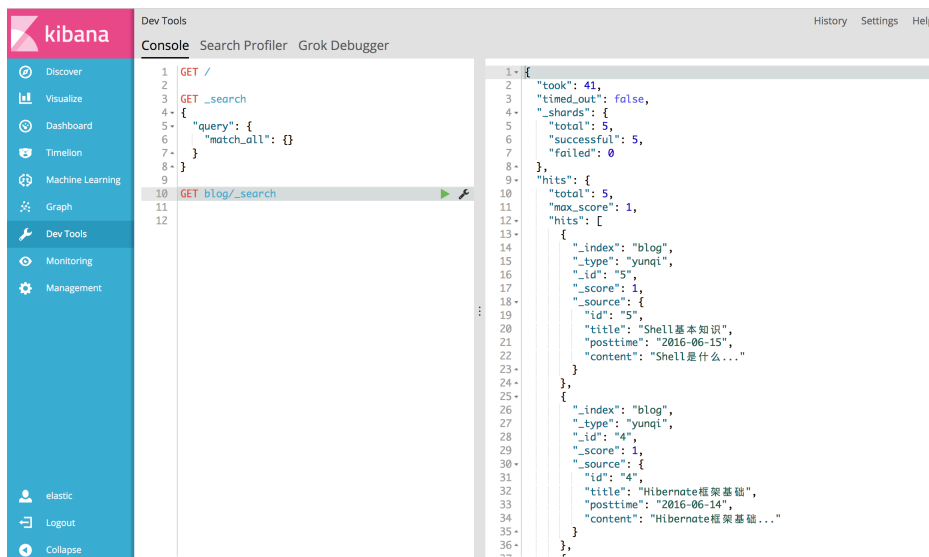
```

Verify the result

Run the following command to verify that the data is successfully written to Elasticsearch:

```
curl -u elastic -XGET es-cn-v0h0jdp990001rta9.elasticsearch.aliyuncs.com:9200/blog/_search? pretty
```

You can also view the result on Kibana:



API analysis

During the Map process, data is read and written by line. The type of input key is object. The type of input value is text. The type of output key is NullWritable, which is a special type of Writable with zero-length serialization. No bytes are written to or read from the stream. It is used as a placeholder.

For example, in MapReduce, a key or value can be declared as NullWritable when you do not need to use the key or value. This example sets the output key to NullWritable. If the output value is set to BytesWritable, serialize the JSON strings.

The Reduce process is not required because only data writing is performed.

Parameter descriptions

```
conf.set( "es.net.http.auth.user" , "X-PACK username" )
```

This parameter specifies the X-PACK username.

```
conf.set( "es.net.http.auth.pass" , "X-PACK password" )
```

This parameter specifies the X-PACK password.

```
conf.setBoolean( "mapred.map.tasks.speculative.execution" , false)
```

This parameter disables speculative execution for the mappers.

```
conf.setBoolean( "mapred.reduce.tasks.speculative.execution" , false)
```

This parameter disables speculative execution for the reducers.

```
conf.set( "es.nodes" , "The internal network address of your Elasticsearch" )
```

This parameter specifies the IP address and port for logging on to the Elasticsearch instance.

```
conf.set( "es.resource" , "blog/yunqi" )
```

This parameter specifies the index names and types that are used to index the data written to the Elasticsearch instance.

```
conf.set( "es.mapping.id" , "id" )
```

This parameter specifies the document IDs. "id" indicates the ID column in the document.

```
conf.set( "es.input.json" , "yes" )
```

This parameter specifies the format of the input files as JSON.

```
job.setInputFormatClass(TextInputFormat.class)
```

This parameter specifies the format of the input stream as text.

```
job.setOutputFormatClass(EsOutputFormat.class)
```

This parameter specifies the output format as EsOutputFormat.

```
job.setMapOutputKeyClass(NullWritable.class)
```

This parameter specifies the the output key format of Map as NullWritable.

```
job.setMapOutputValueClass(BytesWritable.class)
```

This parameter specifies the output value format of Map as BytesWritable.

```
FileInputFormat.setInputPaths(job, new Path(otherArgs[0]))
```

This parameter specifies the path of the files that you need to upload to HDFS.

Logstash deployment

Prepare the environment

Buy Alibaba Cloud ES instances and ECS instances that can access self-built clusters and Alibaba Cloud ES. If you already have ECS instances that meet the requirements, there is no need to purchase additional ECS instances. Prepare the JDK of version 1.8 or later.

The ECS instance on a classic network can be used as long as the ECS instance can access the Alibaba Cloud ES service within VPC through Classiclink.

Download Logstash v5.5.3.

Download the Logstash of the version matching Elasticsearch on the Elastic website (v5.5.3 is recommended).

Decompress the downloaded Logstash package.

```
tar -xzf logstash-5.5.3.tar.gz
# A stringent configuration file checking feature is added to Elasticsearch later than version 5.x.
```

Test cases

Create the user name and password for data access

Create a role

```
curl -XPOST -H "Content-Type: application/json" -u elastic:es-password
http://***instanceId***.elasticsearch.aliyuncs.com:9200/_xpack/security/role/***role-name*** -d '{"cluster":
["manage_index_templates", "monitor"], "indices": [{"names": [ "logstash-*" ],
"privileges":["write", "delete", "create_index"]}]}'
# es-password is the Kibana logon password.
# ***instanceId*** is the ES instance ID.
# ***role-name*** is the role name.
# The default index name of Logstash is in the format of logstash-current date. Therefore, the read and write
permissions on the Logstash-* index must be assigned when you add a user role.
```

Create a user

```
curl -XPOST -H "Content-Type: application/json" -u elastic:es-password
http://***instanceId***.elasticsearch.aliyuncs.com:9200/_xpack/security/user/***user-name*** -d '{"password":
"***logstash-password***", "roles": [ "***role-name***", "full_name": "***your full name***"]}'
```

```
# es-password is the Kibana logon password.  
# ***instanceId*** is the ES instance ID.  
# ***user-name*** is the user name for data access.  
# ***logstash-password*** is the password for data access.  
# ***role-name*** is the role name you created earlier.  
# ***your full name*** is the full name of the current user.
```

Note:

The role and user can also be created on the Kibana page.

Add a role

The screenshot shows the Kibana Management interface. The left sidebar has the 'Management' tab selected. The main content area is titled 'logstash-writer-role'. Under 'Cluster Privileges', the following are checked: 'monitor', 'manage_index_templates', and 'manage_index_templates'. Under 'Run As Privileges', there is a field 'Add a user...'. Under 'Index Privileges', the 'Indices' field is set to 'logstash-*'. The 'Privileges' field contains 'read', 'create', 'write', 'delete', and 'create_index'. The 'Granted Documents Query' and 'Granted Fields' fields are empty. There are 'Save' and 'Cancel' buttons at the bottom.

Add a user

The screenshot shows the Kibana Management interface. The left sidebar has the 'Management' tab selected. The main content area is titled 'New User'. The 'Username' field is set to 'aliyun-logstash-write'. The 'Password' and 'Password Again, Please' fields are empty. The 'Full Name' field is set to 'helloworld'. The 'Email' field is set to 'helloworld@aliyun'. The 'Roles' field is set to 'logstash-writer-role'. There are 'Save' and 'Cancel' buttons at the bottom.

Prepare the conf file.

For more information, see Configuration file structure.

Example

Create the test.conf file on the ECS instance and add the following configurations.

```
input {
  file {
    path => "/your/file/path/xxx"
  }
}
filter {
}
output {
  elasticsearch {
    hosts => ["http://***instanceId***.elasticsearch.aliyuncs.com:9200"]
    user => "***user-name***"
    password => "***logstash-password***"
  }
}
```

instanceId is the ES instance ID.
user-name is the user name for data access.
logstash-password is the password for data access.
Place the user name and password in quotation marks to prevent errors in Logstash startup caused by special characters.

Run

Run Logstash according to the conf file.

```
bin/logstash -f path/to/your/test.conf
```

Logstash provides many input, filter, and output plugins. Only simple configurations are required for data transfer. This example shows how to obtain file changes through Logstash and submit the changed data to the Elasticsearch cluster. All the new contents in the monitored file can be automatically indexed to the Elasticsearch cluster by Logstash.

FAQ

How to configure the index automatically created by the cluster?

To ensure security during users' data operations, Alibaba Cloud Elasticsearch does not allow automatic creation of indexes by default.

Logstash creates indexes by submitting data in data upload, instead of using the create index API. Therefore, before using Logstash to upload data, allow the automatic creation of indexes.

Note:

After the setting is changed and confirmed, the Alibaba ES cluster restarts.

No permissions to create indexes?

Check whether the role you created for data access has the write, delete, and create_index permissions.

Insufficient memory?

By default, Logstash has a 1 GB memory. If your requested ECS memory becomes insufficient, reduce the memory usage of Logstash by changing the memory settings in config/jvm.options.

No quotation marks added to the user name and password in test.conf configuration?

If the user name or password containing special characters in the test.conf file are not added to quotation marks, the previous error message is displayed.

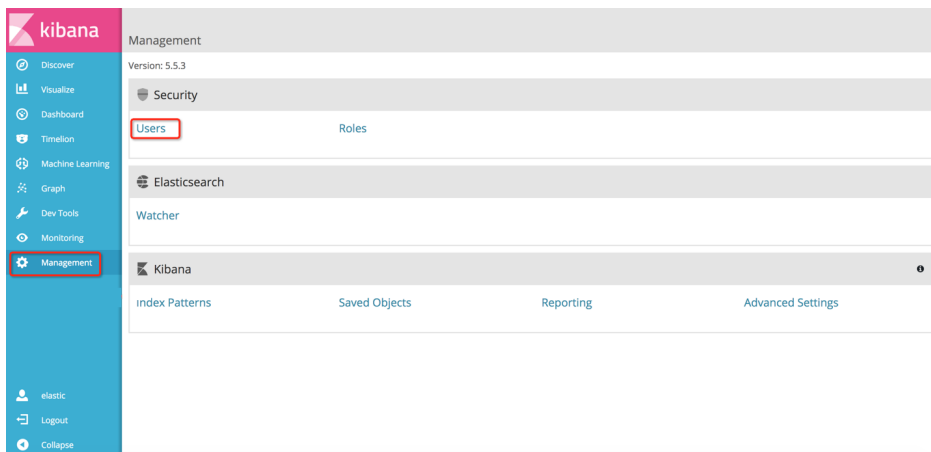
Additional information

To monitor the Logstash node and collect logs:

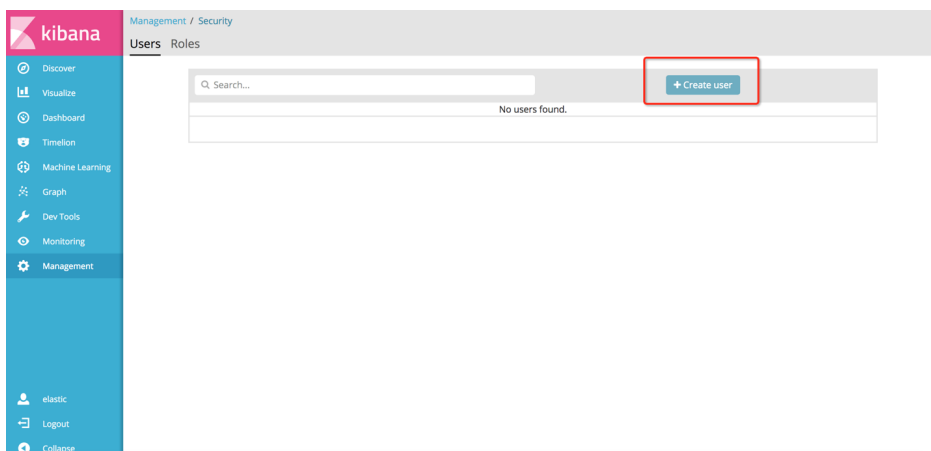
- Install the X-Pack plugin for Logstash. For more information, see [download link](#).
- Deploy the X-Pack after download.
- bin/logstash-plugin install file:///path/to/file/x-pack-5.5.3.zip.
- Add a Logstash monitor user. Alibaba Cloud Elasticsearch cluster disables the logstash_system user by default. You need to create a user with the role name logstash_system. The user name cannot be logstash_system. The user name can be changed. In this example, the user name is logstash_system_monitor. The following two methods are recommended for creating users:

Create a monitor user through the Kibana module

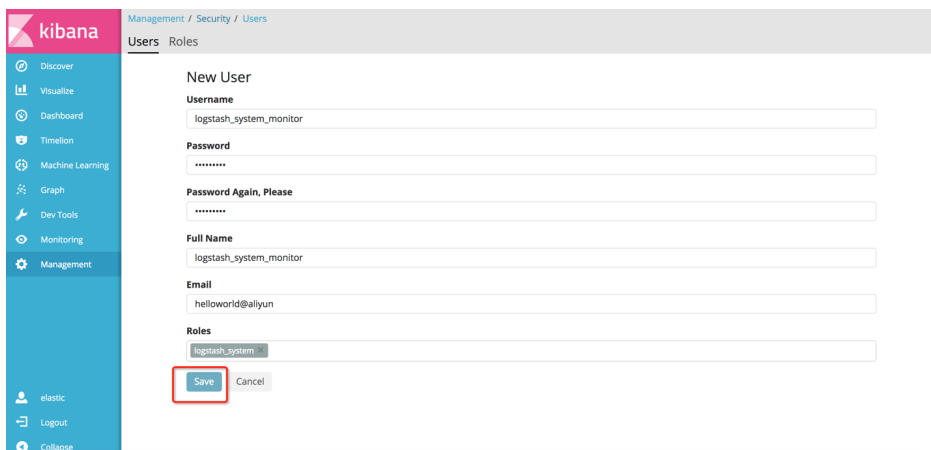
Log on to the Kibana management page, and perform the operations according to the following figure:



Click the **Create User** button.



Enter the required information. Save and submit the information.



Add a user through commands

```
curl -u elastic:es-password -XPOST
```

```
http://***instanceId***.elasticsearch.aliyuncs.com:9200/_xpack/security/user/logstash_system_monitor -d
'{"password" : "***logstash-monitor-password***", "roles" : ["logstash_system"], "full_name" : "your full name"}'

# es-password is the Kibana logon password.
# ***instanceId*** is the ES instance ID.
# ***logstash-monitor-password*** is the password of logstash_system_monitor.
```

ES self-built migration

Prerequisites

You must meet the following requirements to migrate data from a user-created Elasticsearch instance to an Alibaba Cloud Elasticsearch instance.

The ECS instance that hosts the user-created Elasticsearch instance must be connected to a VPC network. **ECS instances connected to a VPC network through a ClassicLink are not supported. The ECS instance and your Alibaba Cloud Elasticsearch instance must be connected to the same VPC network.

You can use an ECS instance to run the `reindex.sh` script. To perform this task, you must make sure that the ECS instance can access port 9200 on the user-created and Alibaba Cloud Elasticsearch instances.

The VPC security group must allow all IP addresses in the IP whitelist to access the ECS instance and port 9200 must be open.

The VPC security group must allow the IP addresses of all Elasticsearch instance nodes to access the ECS instance. You can view these IP addresses in the Kibana console.

To check whether the ECS instance that runs the script can access port 9200 on the source and target Elasticsearch instances, run the `curl -XGET http://<host>:9200` command on the ECS instance.

Procedure

1. Create indexes.
2. Migrate data.

Create indexes

You must create indexes on the target Elasticsearch instance based on the indexes on the source cluster. You can also choose to enable dynamic index creation and dynamic mapping (not recommended) to create indexes on the target cluster. You must enable auto index creation before you enable dynamic index creation.

The following section provides a Python script (indiceCreate.py). You can copy all the indexes from the source cluster to the target cluster. Only the number of shards and zero replica are configured. You need to configure the remaining settings.

Note:

If the following error occurs when you run the cURL command, add the -H "Content-Type: application/json" parameter to the command and run the command again.

```
{"error": "Content-Type header [application/x-www-form-urlencoded] is not supported", "status": 406}
```

```
// Obtain all the indexes on the source cluster. If you do not have the required permissions, remove the "-u user:pass" parameter. Make sure that you have replaced oldClusterHost with the name of the ECS instance that hosts the source cluster.
curl -u user:pass -XGET http://oldClusterHost/_cat/indices | awk '{print $3}'

// Based on the returned indexes, obtain the setting and mapping of the index that you need to migrate for the specified user. Make sure that you have replaced indexName with the index name that you need to query.
curl -u user:pass -XGET http://oldClusterHost/indexName/_settings,_mapping?pretty=true

// Create a new index in the target cluster according to the _settings and _mapping settings that you have obtained from the preceding step. You can set the number of index replicas to zero to accelerate the data synchronization process, and change the number to one after the migration has completed.
// ewClusterHost indicates the ECS instance that hosts the target cluster, testindex indicates the name of the index that you have created, and testtype indicates the type of the index.
curl -u user:pass -XPUT http://<newClusterHost>/<testindex> -d '{
"testindex" : {
"settings" : {
"number_of_shards" : "5", //Set the number of shards for the corresponding index on the source cluster, for example, 5
"number_of_replicas" : "0" //Set the number of index replicas to zero
}
},
"mappings" : { //Set the mapping for the index on the source cluster. For example, you can set the mapping as follows
"testtype" : {
"properties" : {
"uid" : {
"type" : "long"
}
},
"name" : {
```

```
"type" : "text"
},
"create_time" : {
  "type" : "long"
}
}
}
}
}
```

Accelerate the synchronization process

Note:

If the index is too large, you can set the number of replicas to 0 and the refresh interval to -1 before migration. After the data has been migrated, set the replicas and refresh settings to the previous values. This accelerates the synchronization process.

```
// You can set the number of index replicas to zero and disable refresh, to accelerate the migration process.
curl -u user:password -XPUT 'http://<host:port>/indexName/_settings' -d' {
  "number_of_replicas" : 0,
  "refresh_interval" : "-1"
}'

// After the data has been migrated, set the number of index replicas to `1` and the refresh interval to `1` (default
value, which means 1 second).
curl -u user:password -XPUT 'http://<host:port>/indexName/_settings' -d' {
  "number_of_replicas" : 1,
  "refresh_interval" : "1s"
}'
```

Data migration

To ensure data consistency after the migration, you must **stop the write operation on the source cluster**. You do not need to stop the read operation. After the migration process has been completed, switch the read and write operations to the target cluster. Data inconsistency may occur if you do not stop the write operation on the source cluster.

Note:

When using the following method to migrate data, if you access the source cluster using an IP address and a port, you must configure a reindex whitelist in the YAML file of the target cluster, and add the IP address of the source cluster to the whitelist: `reindex.remote.whitelist: 1.1.1.1:9200,1.2. *. *.*`

If you access the source cluster using a domain name, do not use the `http://host:port/path` format. The domain name must not contain the path.

1. Migrate small amounts of data

Run the `reindex.sh` script.

```
#!/bin/bash
# file:reindex.sh

indexName="The name of the index"
Newclusteruser = "The username that is used to log on to the target cluster"
Newclusterpass = "The password that is used to log on to the target cluster"
Newclusterhost = "The ECS instance that hosts the target cluster"
Oldclusteruser = "The username that is used to log on to the source cluster"
Oldclusterpass = "The password that is used to log on to the source cluster"
# Set oldClusterHost in the format of [scheme]://[host]:[port]. Example: http://10.37.1.1:9200.
Oldclusterhost = "The ECS instance that hosts the source cluster"

curl -u ${newClusterUser}:${newClusterPass} -XPOST "http://${newClusterHost}/_reindex? pretty" -H "Content-Type: application/json" -d'{
  "source": {
    "remote": {
      "host": "${oldClusterHost}",
      "username": "${oldClusterUser}",
      "password": "${oldClusterPass}"
    },
    "index": "${indexName}",
    "query": {
      "match_all": {}
    }
  },
  "dest": {
    "index": "${indexName}"
  }
}'
```

2. Migrate large amounts of data without delete operations and with update time

If the amount of data is large without deletion operations, you can use rolling migration to minimize the time period during which your write operation is suspended. Rolling migration requires that your data schema has a time-series attribute that indicates the update time. You can stop the write operation after the data has been migrated, then migrate the incremental data. Switch the read and write operations to the target cluster.

```

#!/bin/bash
# file: circleReindex.sh
# CONTROLLING STARTUP:
# This script is used to remotely rebuild the index using the reindex operation. Requirements:
#1. You have created the index on the target cluster, or the target cluster supports automatic index creation and
dynamic mapping.
# 2. You must configure an IP whitelist in the YML file of the target cluster: reindex.remote.whitelist: 172.16.123.
*:9200
#3. You need to specify the ECS instance address in the following format: [scheme]://[host]:[port].

USAGE="Usage: sh circleReindex.sh <count>
count: The number of executions. A negative number indicates loop execution. You can set this parameter to
perform the reindex operation only once or multiple times.
Example:
sh circleReindex.sh 1
sh circleReindex.sh 5
sh circleReindex.sh -1"

indexName="The name of the index"
newClusterUser="The username that is used to log on to the target cluster"
newClusterPass="The password that is used to log on to the target cluster"
oldClusterUser="The username that is used to log on to the source cluster"
oldClusterPass="The password that is used to log on to the source cluster"
## http://myescluster.com
newClusterHost="The host of the target cluster"
# You need to address of the ECS instance that hosts the source cluster in the following format:
[scheme]://[host]:[port]. Example: http://10.37.1.1:9200
oldClusterHost="The ECS instance that hosts the source cluster"
timeField="The field that specifies the time window during which the incremental data is migrated"

reindexTimes=0
lastTimestamp=0
curTimestamp=`date +%s`
hasError=false

function reIndexOP() {
reindexTimes=$((reindexTimes + 1)
curTimestamp=`date +%s`

ret=`curl -u ${newClusterUser}:${newClusterPass} -XPOST "${newClusterHost}/_reindex? pretty" -H "Content-Type:
application/json" -d '{
"source": {
"remote": {
"host": "${oldClusterHost}",
"username": "${oldClusterUser}",
"password": "${oldClusterPass}"
},
"index": "${indexName}",
"query": {
"range" : {
"${timeField}" : {
"gte" : "${lastTimestamp}",
"lt" : "${curTimestamp}"
}
}
}
}'

```

```

}
},
"dest": {
  "index": "${indexName}"
}
}`
lastTimestamp=${curTimestamp}
echo "${reindexTimes} reindex operations have been performed. The last reindex operation is completed at
${lastTimestamp} Result:${ret}"
if [[ ${ret} == *error* ]]; then
  hasError=true
  echo "An unknown error occurred while performing this operation. All subsequent operations have been
  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 index ${indexName}"
start $1
echo "You have performed ${reindexTimes} reindex operations"

```

3. Migrate large amounts of data without deletion operations or update time

When you need to migrate large amounts of data and no update time field is defined in the mapping, you must add a update time field to the code that is used to access the source cluster. After the field has been added, you can migrate the existing data, and then use rolling migration described in the preceding data migration plan to migrate the incremental data.

The following script shows how to migrate the existing data without the update time field.

```

#!/bin/bash
# file:miss.sh

indexName="The name of the index"
newClusterUser="The username that is used to log on to the target cluster"
newClusterPass="The password that is used to log on to the target cluster"
newClusterHost="The ECS instance that hosts the target cluster"
oldClusterUser="The username that is used to log on to the source cluster"
oldClusterPass="The password that is used to log on to the source cluster"
# The address of the ECS instance that hosts the source cluster must be in this format: [scheme]://[host]:[port].
Example: http://10.37.1.1:9200.
oldClusterHost="The ECS instance that hosts the source cluster"
timeField="update_time"

curl -u ${newClusterUser}:${newClusterPass} -XPOST "http://${newClusterHost}/_reindex? pretty" -H "Content-Type: application/json" -d '{
  "source": {
    "remote": {
      "host": "${oldClusterHost}",
      "username": "${oldClusterUser}",
      "password": "${oldClusterPass}"
    },
    "index": "${indexName}",
    "query": {
      "bool": {
        "must_not": {
          "exists": {
            "field": "${timeField}"
          }
        }
      }
    }
  },
  "dest": {
    "index": "${indexName}"
  }
}'

```

4. Migrate data without suspending the write operation

This feature will soon be available.

Use the batch creation operation to replicate indexes from the source cluster

The following Python script shows how to replicate indexes from the source cluster to the target cluster. The default number of newly created index replicas is 0.

```

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

```

```

import sys
import base64
import time
import httplib
import json

## The ECS instance that hosts the source cluster (ip+port)
oldClusterHost = "old-cluster.com"
# The username that is used to log on to the source cluster. The username field can be left empty
oldClusterUserName = "old-username"
## The password that is used to log on to the source cluster. The password field can be left empty
oldClusterPassword = "old-password"
## The ECS instance that hosts the target cluster (ip+port)
newClusterHost = "new-cluster.com"
## The username that is used to log on to the target cluster. The username field can be left empty
newClusterUser = "new-username"
## The password that is used to log on to the target cluster. The password field can be left empty
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:
        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 + " The original settings: \n" + indexSettings
    settingsDict = json.loads(indexSettings)
    ## The number of shards equals the number of indexes on the source cluster by default
    number_of_shards = settingsDict[index]["settings"]["index"]["number_of_shards"]
    ## The default number of replicas 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 + "The 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, oldClusterPassword)
    mappingStr = getMapping(oldIndexName, oldClusterHost, oldClusterUserName, oldClusterPassword)
    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 + "settings and mappings: \n" + createstatement
    endpoint = "/" + newIndexName
    createResult = httpPut(newClusterHost, endpoint, createstatement, newClusterUser, newClusterPassword)
    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 that will not be recreated. Create the index based on your needs."

```

Note:

You can use Logstash to migrate data. For more information, see [Logstash deployment](#).