Alibaba Cloud Elasticsearch

Best Practices

Issue: 20190605

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 due to product version upgrades , adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall 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 the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults " and "as available" basis. 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 bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, 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 content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, 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 the Alibaba Cloud website, product programs, or content 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 contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information 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.
	This warning information 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 restore business.
	This indicates warning informatio n, supplementary instructions, and other content that the user must understand.	Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus , page names, and other UI elements.	Click OK.
`Courier font`	It is used for commands.	Run the `cd / d C :/ windows` command to enter the Windows system folder.
Italics	It is used for parameters and variables.	`bae log list -- instanceid` *Instance_ID*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *[-all|-t]*

Style	Description	Example
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich` *{stand | slave}*

Contents

1 Build a visualized O&M system with Beats

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 objects, such as Logstash or Elasticsearch.

As an agent of Beats and a lightweight shipper, Metricbeat is designed to collect metrics from your systems and services, and then send the metrics to target objects, such as Elasticsearch. Metricbeat is a lightweight method to send system and service statistics from CPUs to memory, Redis to NGINX, and much more.

This topic describes how to use Metricbeat to collect metrics from a MacBook, send the metrics to an Alibaba Cloud Elasticsearch instance, and generate a corresponding dashboard in Kibana.

> Note:
> The procedures to collect metrics from a computer that runs a Linux or Windows system and to send the metrics to an Alibaba Cloud Elasticserach instance are similar.

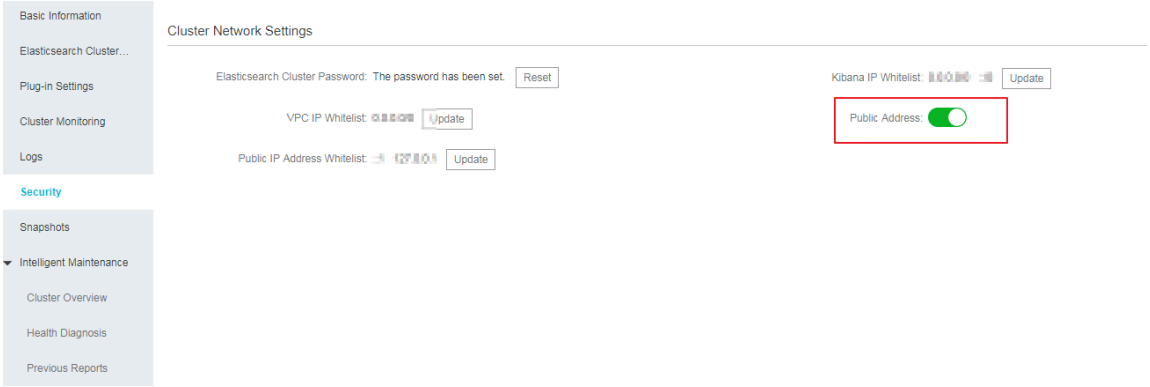
1. Purchase and configure an Alibaba Cloud Elasticsearch instance

   If you do not have an Alibaba Cloud Elasticsearch instance, you must activate Alibaba Cloud Elasticsearch and create an instance ####. You can then send the data collected from the MacBook to the Alibaba Cloud Elasticsearch instance through the internal or public IP address of the instance.

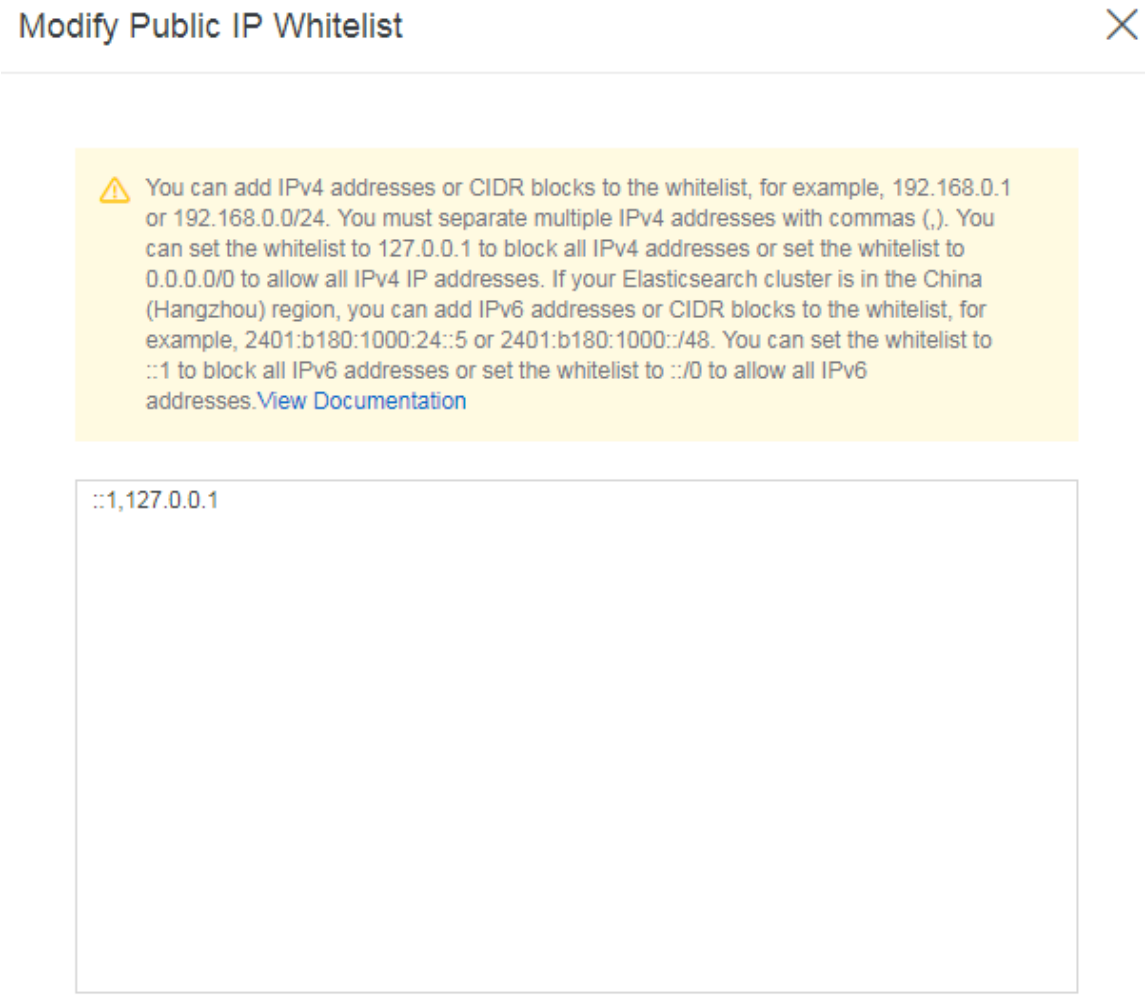
   > Note:
   > - If you access the Alibaba Cloud Elasticsearch instance through its public IP address, you must switch on Public Address and configure a public IP address whitelist on the Security page.
   > - If you access the Alibaba Cloud Elasticsearch instance through its internal IP address, you must create an Alibaba Cloud Elastic Compute Service (ECS)

instance in the same VPC and region as the Alibaba Cloud Elasticsearch instance to manage access to the Elasticsearch.

a. Log on to the Alibaba Cloud Elasticsearch console, click the instance name or ID, and then click Security in the left-side navigation pane. On the Security page, switch on Public Address.

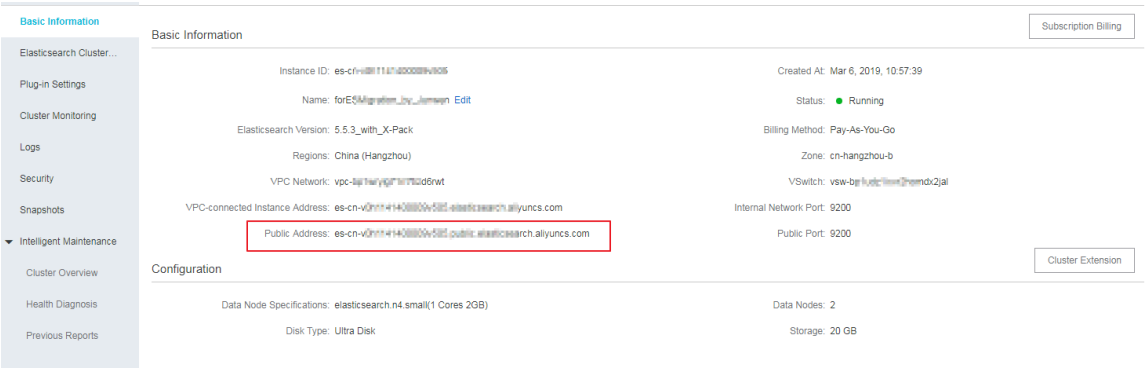


b. Add the public IP address of the MacBook to the whitelist.

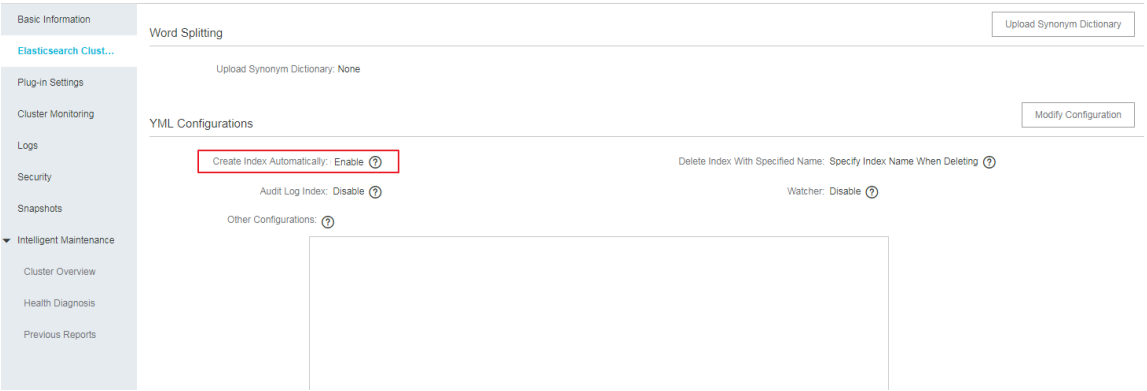


> **Notice:**
> If you use a public network, add the IP address of the jump server that controls outbound network traffic of the public network to the whitelist. If you cannot obtain the IP address of the jump server, add `0.0.0.0/1`, `128.0.0.0/1` to the whitelist to allow a certain part of IP addresses. This setting exposes the Alibaba Cloud Elasticsearch instance to the public network. Evaluate the risks and proceed with caution.

c. After you complete the configuration, click Basic Information in the left-side navigation pane and copy the public IP address of the Alibaba Cloud Elaticsearch instance.



d. Modify the YML configuration. On the YML Configurations page, enable Create Index Automatically. By default, this feature is disabled. This operation restarts the Elasticsearch instance and takes some time to take effect.



2. Download and configure Metricbeat

- *Metricbeat installation package for Mac operating systems*.
- *Metricbeat installation package for 32-bit Linux operating systems*.
- *Metricbeat installation package for 64-bit Linux operating systems*.
- *Metricbeat installation package for 32-bit Windows operating systems*.
- *Metricbeat installation package for 64-bit Windows operating systems*.

a. Download, unzip, and open the Metricbeat file.

```
zhaohongyangdeMacBook-Pro:Desktop zhaohongyang$ cd metricbeat-6.3.1-darwin-x86_64
zhaohongyangdeMacBook-Pro:metricbeat-6.3.1-darwin-x86_64 zhaohongyang$ ls
LICENSE.txt                 data                        logs                        metricbeat.yml
NOTICE.txt                  fields.yml                  metricbeat                  modules.d
README.md                   kibana                      metricbeat.reference.yml
zhaohongyangdeMacBook-Pro:metricbeat-6.3.1-darwin-x86_64 zhaohongyang$
```

b. Open and edit the Elasticsearch output section of the metricbeat.yml file. You need to uncomment the corresponding 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-[redacted].public.elasticsearch.aliyuncs.com:9200"]

  # Optional protocol and basic auth credentials.
  protocol: "http"
  username: "[redacted]:"
  password: "[redacted]"
```

Note:

Alibaba Cloud Elasticsearch provides the following access control information:

- hosts: the public or internal IP address of the Alibaba Cloud Elasticsearch instance. This example uses the public IP address.
- protocol: set to `http` .
- username: the default username is elastic.
- password: the password that is used to log on to Alibaba Cloud Elasticsearch.

3. Activate Metricbeat

   Run the following command to activate and use Metricbeat to send data to the Alibaba Cloud Elasticsearch instance.

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

```
zhaohongyangdeMacBook-Pro:metricbeat-6.3.1-darwin-x86_64 zhaohongyang$ ./metricbeat -e -c metricbeat.yml
```

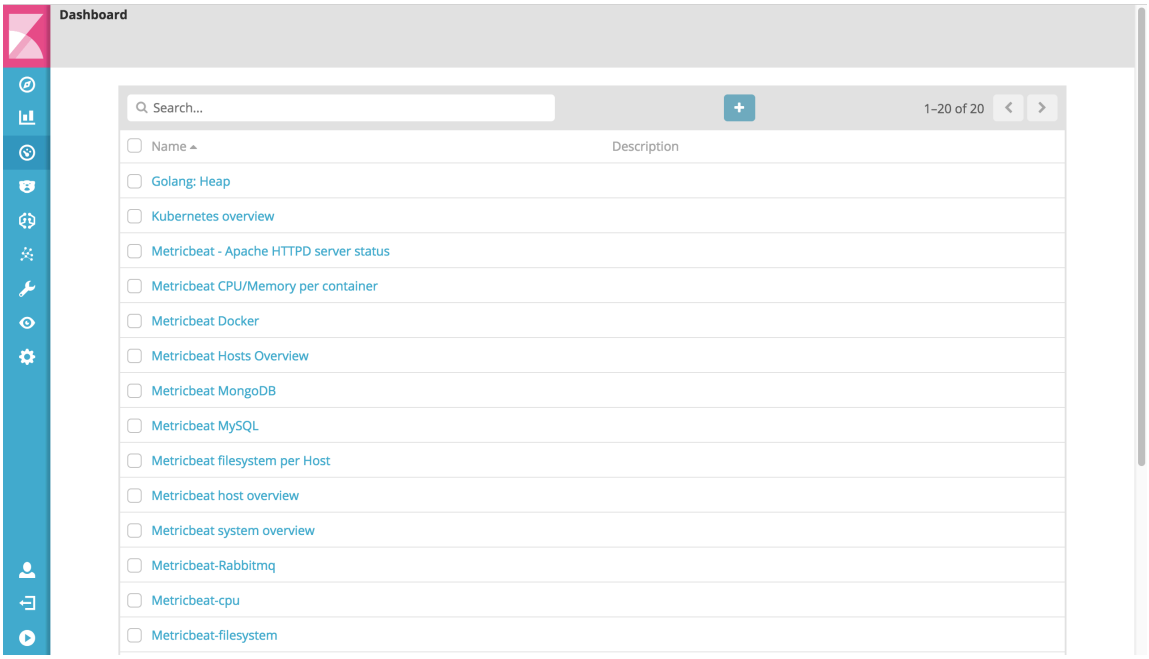
4. View the dashboard in Kibana

   Click Kibana Console in the upper-right corner in the Alibaba Cloud Elasticsearch console. You will be directed to the Dashboard page, as shown in the following figure:

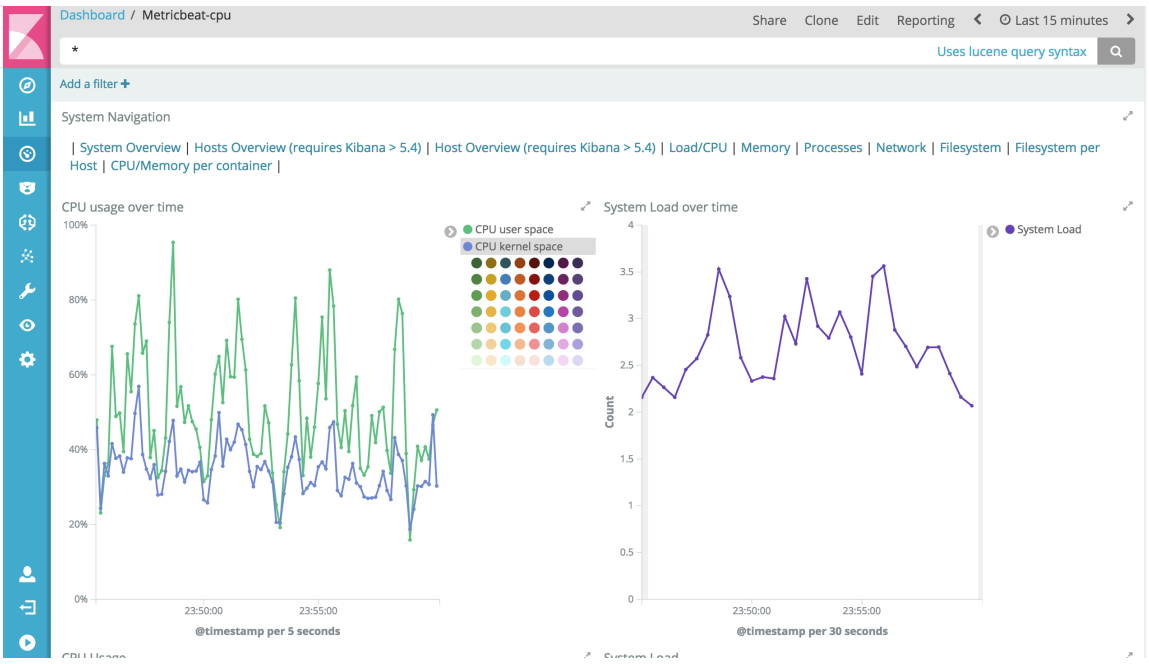
   Note:

If you have not created an index pattern in the Kibana console, the corresponding information may not be displayed on the dashboard. To resolve this issue, create an index pattern and view the information on the Dashboard page again.

a. List of metrics.



b. CPU metrics.



Note:

You can schedule the system to refresh data every five seconds and generate reports, and configure a webhook to send alerts when an exception occurs.

2 Use Curator

Install Elasticsearch Curator

1. Purchase an Alibaba Cloud ECS instance in the same VPC network as your Alibaba Cloud Elasticsearch instance. This example uses an ECS instance that runs CentOS 7.3 64-bit.
2. Run the following command:

   a. Install Elasticsearch Curator:

   ```
   pip install elasticsearch-curator
   ```

   Note:

   - We recommend that you install Elasticsearch Curator 5.6.0. This version supports Alibaba Cloud Elasticsearch 5.5.3 and 6.3.2.
   - *Curator and Elasticsearch version compatibility*.

   b. View the version of the Curator:

   ```
   curator --version
   ```

   Returned version information:

   ```
   curator, version 5.6.0
   ```

Singleton command line interface

- You can use `curator_cli` to perform an action.
- *Singleton command line interface*.

Note:

- You can perform only one action each time.
- Not all of the actions can be performed by using Curator, for example, Alias and Restore.

Schedule tasks using Crontab

You can use the crontab and curator commands to schedule a task to perform multiple actions.

Curator command:

```
curator [ OPTIONS ] ACTION_FIL E
Options :
 -- config PATH Path to configurat ion 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 file (official reference)*.
- When you run the curator command, you must specify the *action.yml file (official reference)*.

Hot-warm architecture practice

Use Curator to migrate indexes from hot nodes to warm nodes (official reference).

Migrate indexes from hot nodes to warm nodes

1. Create the config.yml file in the `/ usr / curator /` path as follows:

> **Notice:**
>
> - `hosts` : Replace hosts with the address of the Alibaba Cloud Elasticsearch instance that you need to access. In this example, the private address of the Elasticsearch instance is used.
> - `http_auth` : Replace http_auth with the username and password that are used to log on to the Alibaba Cloud Elasticsearch instance.

```
client :
  hosts :
   - http :// es - cn - 0pp0z9p2v0 0031234 . elasticsea rch .
aliyuncs . com
  port : 9200
  url_prefix :
  use_ssl : False
  certificat e :
  client_cer t :
  client_key :
  ssl_no_val idate : False
  http_auth : user : password
  timeout : 30
  master_onl y : False
logging :
  loglevel : INFO
  logfile :
  logformat : default
```

```
    blacklist : [' elasticsea  rch ', ' urllib3 ']
```

2. Create the action.yml file in the `/ usr / curator /` path as follows:

> **Note:**
>
> - The following content migrates indexes created 30 minutes ago and starting with `logstash -` from hot nodes to warm nodes.
> - You can customize the following content based on your business needs.

```
actions :
  1 :
    action :  allocation
    descriptio  n : " Apply   shard   allocation   filtering
rules   to   the   specified   indices "
    options :
      key :  box_type
      value :  warm
      allocation  _type :  require
      wait_for_c  ompletion :  true
      timeout_ov  erride :
      continue_i  f_exceptio  n :  false
      disable_ac  tion :  false
    filters :
   -  filtertype :  pattern
      kind :  prefix
      value :  logstash -
   -  filtertype :  age
      source :  creation_d  ate
      direction :  older
      timestring : '% Y -% m -% dT % H :% M :% S '
      unit :  minutes
      unit_count :  30
```

3. Check whether the curator command can run normally:

```
curator  -- config  / usr / curator / config . yml  / usr / curator
/ action . yml
```

The following information is returned when the command runs normally:

```
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   allocation   filtering
rules   to   the   specified   indices
2019 - 02 - 12   20 : 11 : 30 , 693   INFO        Updating   index
  setting  {' index . routing . allocation . require . 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. **Run the crontab command to run the curator command at an interval of 15 minutes:**

```
*/ 15  * * * *  curator  -- config  / usr / curator / config . yml
 / usr / curator / action . yml
```

3 Data synchronization and migration

3.1 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 and *Data Integration*, which allows you to synchronize offline data to Elasticsearch every five 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

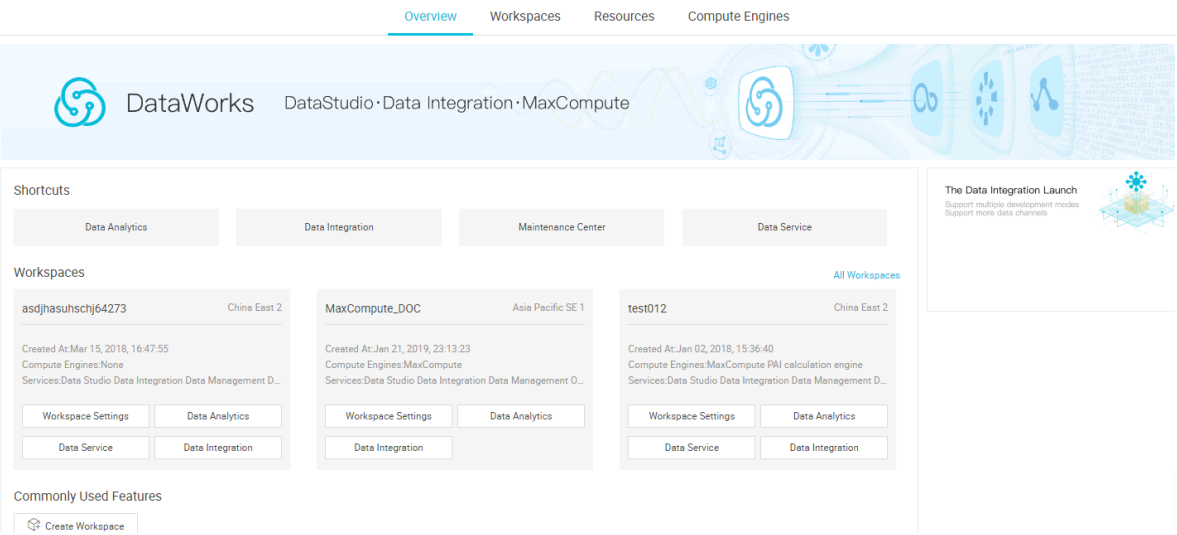
1. Buy an ECS instance that is in the same VPC as the Elasticsearch service. Allocate a public IP address to the ECS instance or enable the elastic IP address for the ECS

instance. To lower costs, you can use an existing ECS instance. For how to buy an ECS instance, see *Step 2. Create an 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.

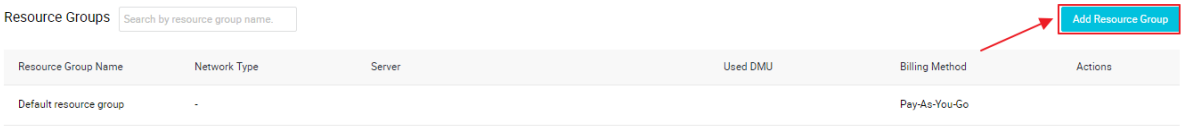
2. Log on to the *Data Integration console* to open the workbench.

   If Data Integration or DataWorks has been enabled, you can see:



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

3. 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 *Add task resources*.



4. Configure the data synchronization script in the Data Integration service. For the configuration procedure, see *Script mode configuration*. For the instructions on configuring Elasticsearch, see *Configure Elasticsearch 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.

5. 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 pop-up window.
   - A periodic job is executed at 00:00 on the next day according to the rule you have configured.

6. 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.

3.2 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. For more information, see *Data interconnection between ES-Hadoop and Elasticsearch* and *Use ES-Hadoop on E-MapReduce*.

Prerequisites

1. 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 *Step 3 : Create a cluster*.

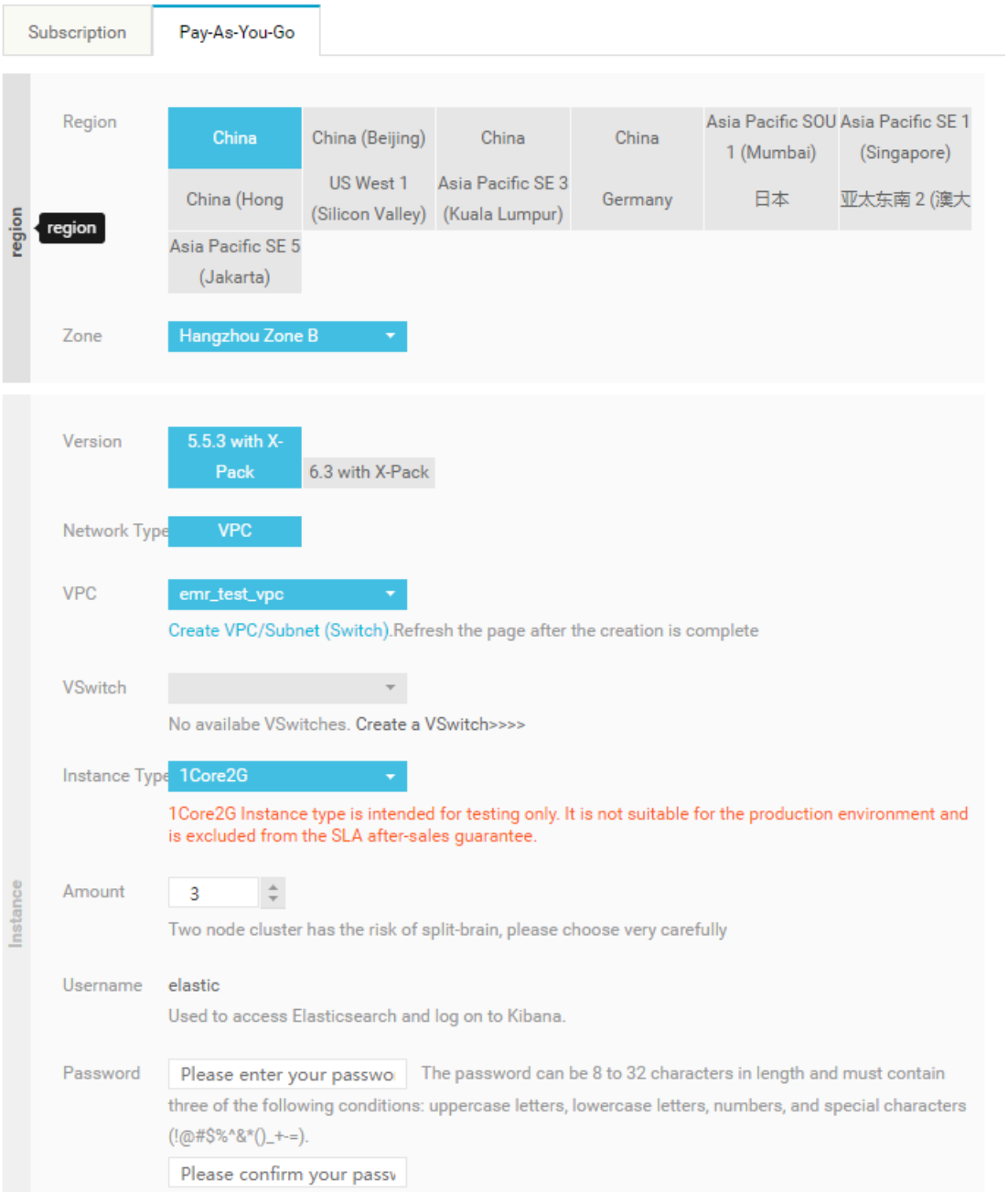
   Specifically, the following EMR Hadoop version information is used:

   - 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 public and private IPs for the ECS master nodes, and selects non-high availability (non-HA) mode。

2. 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*.



3. DataWorks

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

Prepare data

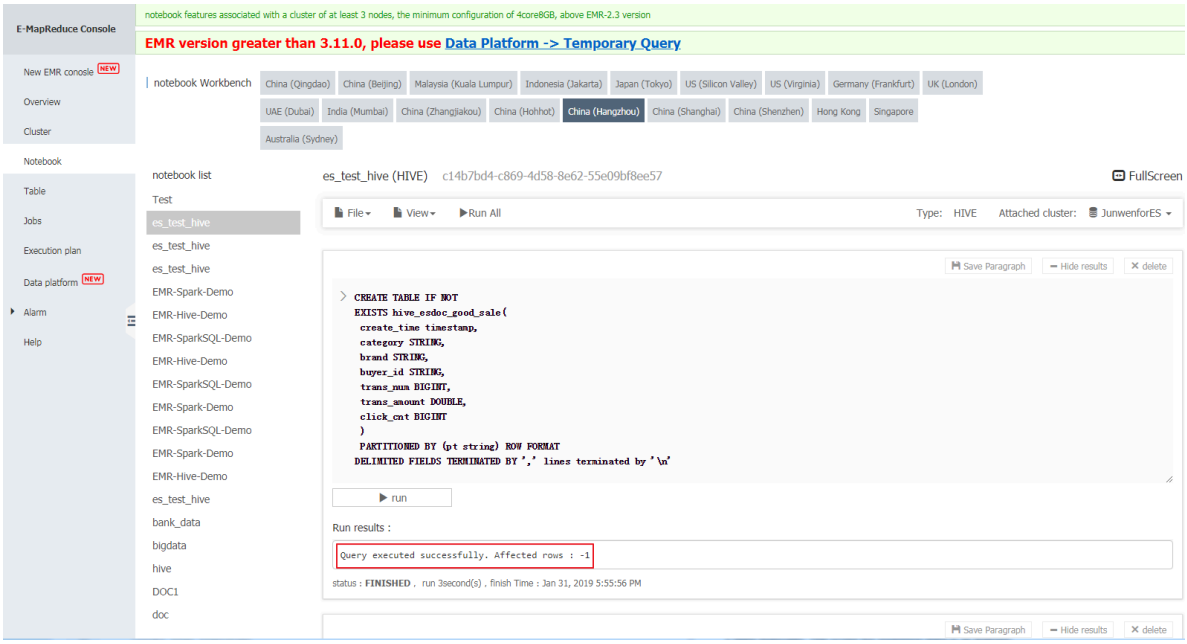
To create test data in the Hadoop cluster, follow these steps:

1. Log on to the *EMR* console, go to Old EMR Scheduling, and in the left-side navigation pane, click Notebook.
2. 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.
3. Enter the syntax for creating a Hive table:

```
CREATE   TABLE   IF   NOT

EXISTS   hive_esdoc  _good_sale (
 create_tim  e   timestamp ,
 category   STRING ,
 brand   STRING ,
 buyer_id   STRING ,
 trans_num   BIGINT ,
 trans_amou  nt   DOUBLE ,
 click_cnt   BIGINT
)
 PARTITIONE  D   BY  ( pt   string )  ROW   FORMAT
DELIMITED   FIELDS   TERMINATED   BY  ','  lines   terminated   by
'\ n '
```

4. 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.



5. 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 ) ;
```

6. 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

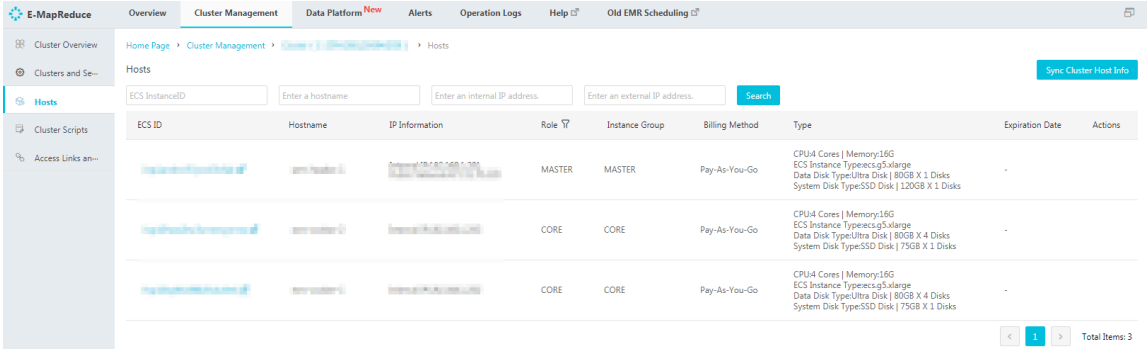
> **Note:**
> Because the network environment of the DataWorks project is generally not connected to that of the Hadoop cluster core nodes, you can customize your resource groups to run the synchronization task of DataWorks on Hadoop cluster master nodes (this is because Hadoop cluster master and core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

1. View core nodes of the EMR Hadoop cluster.

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

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

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



> **Note:**
> 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.

   d. 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. 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

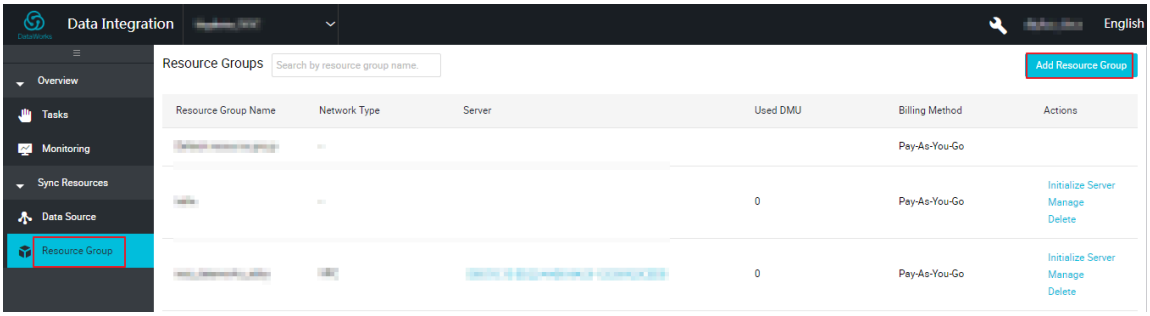
-------------------------------------------------
Live datanodes (2):

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

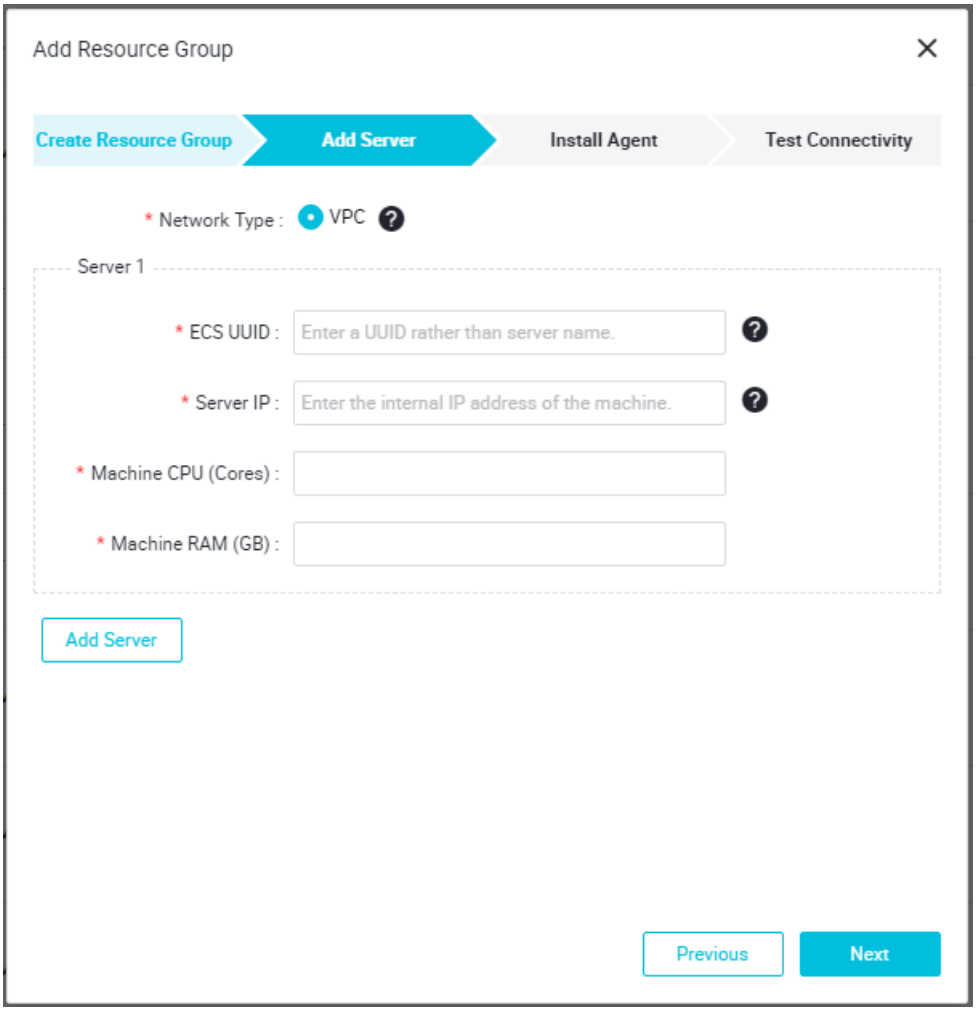

Name: 192.168.1.205:50010 (emr-worker-1.cluster-77026)
Hostname: emr-worker-1.cluster-77026
Decommission Status : Normal
Configured Capacity: 333373341696 (310.48 GB)
DFS Used: 104890368 (100.03 MB)
Non DFS Used: 302723072 (288.70 MB)
DFS Remaining: 332965728256 (310.10 GB)
DFS Used%: 0.03%
DFS Remaining%: 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
```

2. Create a custom resource group

   a. 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 task resources*.



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



   - Network type is a proprietary network (VPC).

> **Note:**
> 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.

c. 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*. 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.

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

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

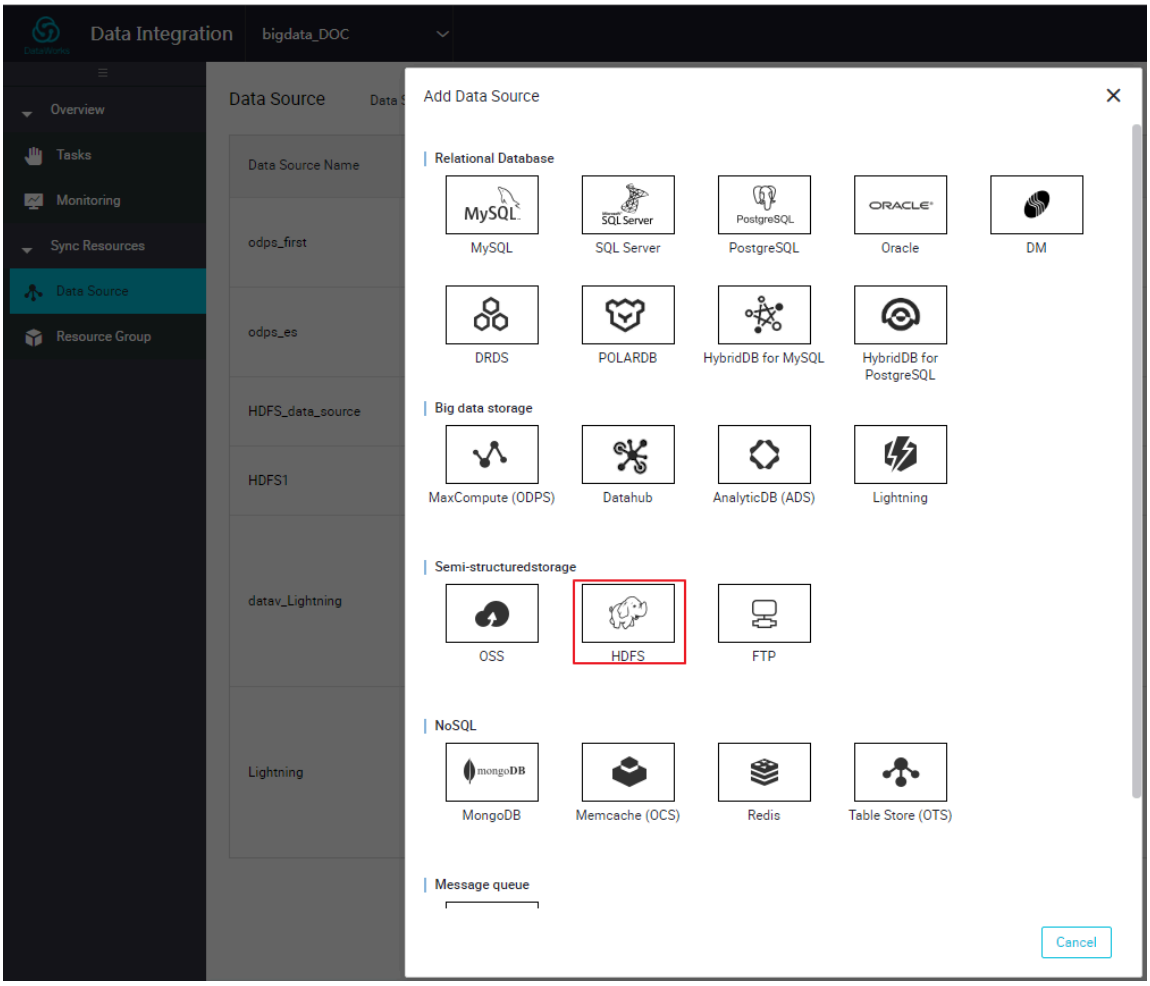
If the status is unavailable, log on to the master node and run the `tail -f /home/admin/alisataskn ode/logs/heartbeat.log` command

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/alisatasknode/logs/heartbeat.log
2018-09-06 21:47:34,440 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
^C
```

3. Create a data source

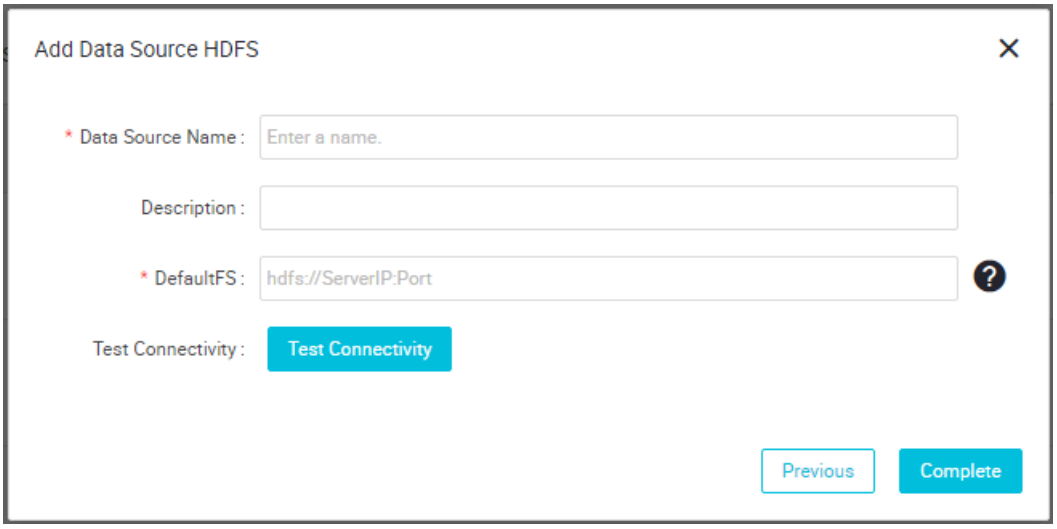
   a. In the Data Integration page of DataWorks, click Data Sources > New source, and select HDFS.



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

> **Note:**
> For an EMR Hadoop cluster, if it is a non-HA cluster, the address is set to `hdfs://IP of emr-header-1:9000`. If it is an HA cluster, the address

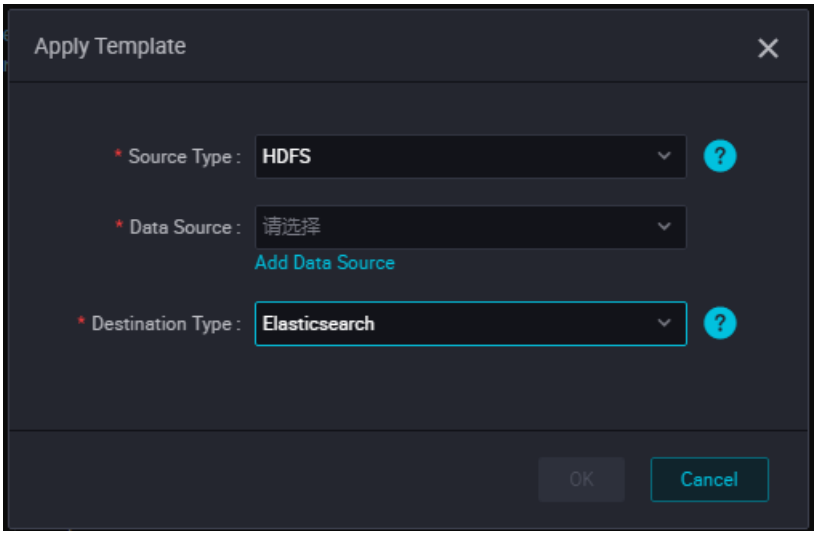
is set to `hdfs :// IP of emr - header - 1 : 8020` . In this example, emr-header-1 and DataWorks are connected through a VPC network, so an intranet IP is set, and the test connectivity is unavailable.



4. Configure a data synchronization task

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

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



   c. 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*.

Hdfs Reader

```
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",
```

- 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 configurat ion of your ES instance; and setting, which refers to synchronization configurat ions such as packet loss rate and maximum concurrency.
- 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_g ood_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.
- 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 publick network access whitelist (including the server IP addresss 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.

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

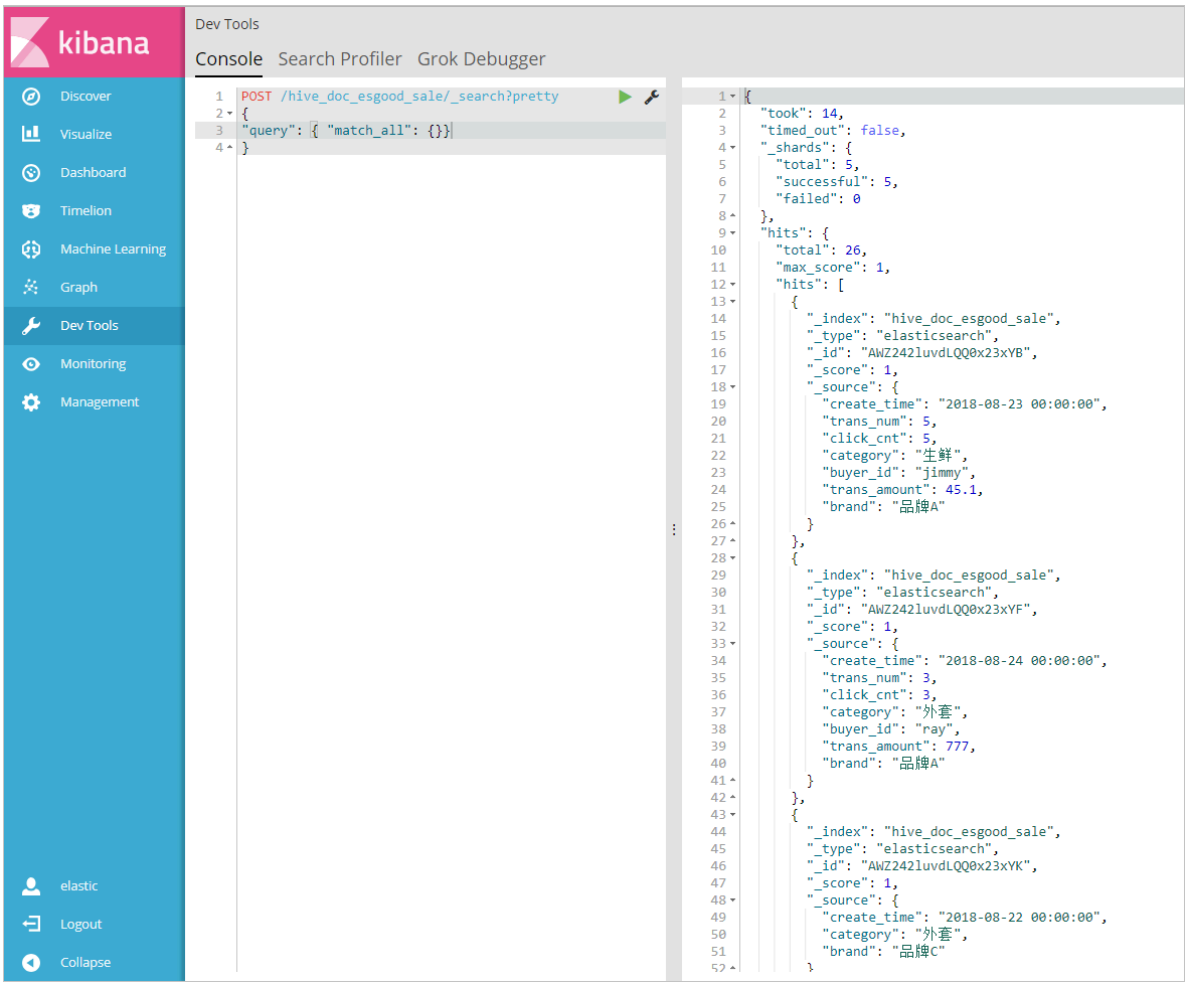
   If the prompt Task run successfully is displayed, it indicates that the task is synchronized successfully. If the task fails to run, copy the error logs for troubleshooting.

Verify the synchronization result

1. Go to the Elasticsearch console, click Kibana console in the upper right corner and then select Dev Tools.
2. Run the following command to view the synchronized data.

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

`hive_doc_e sgood_sale` is the value of the `index` field when the data is synchronized.

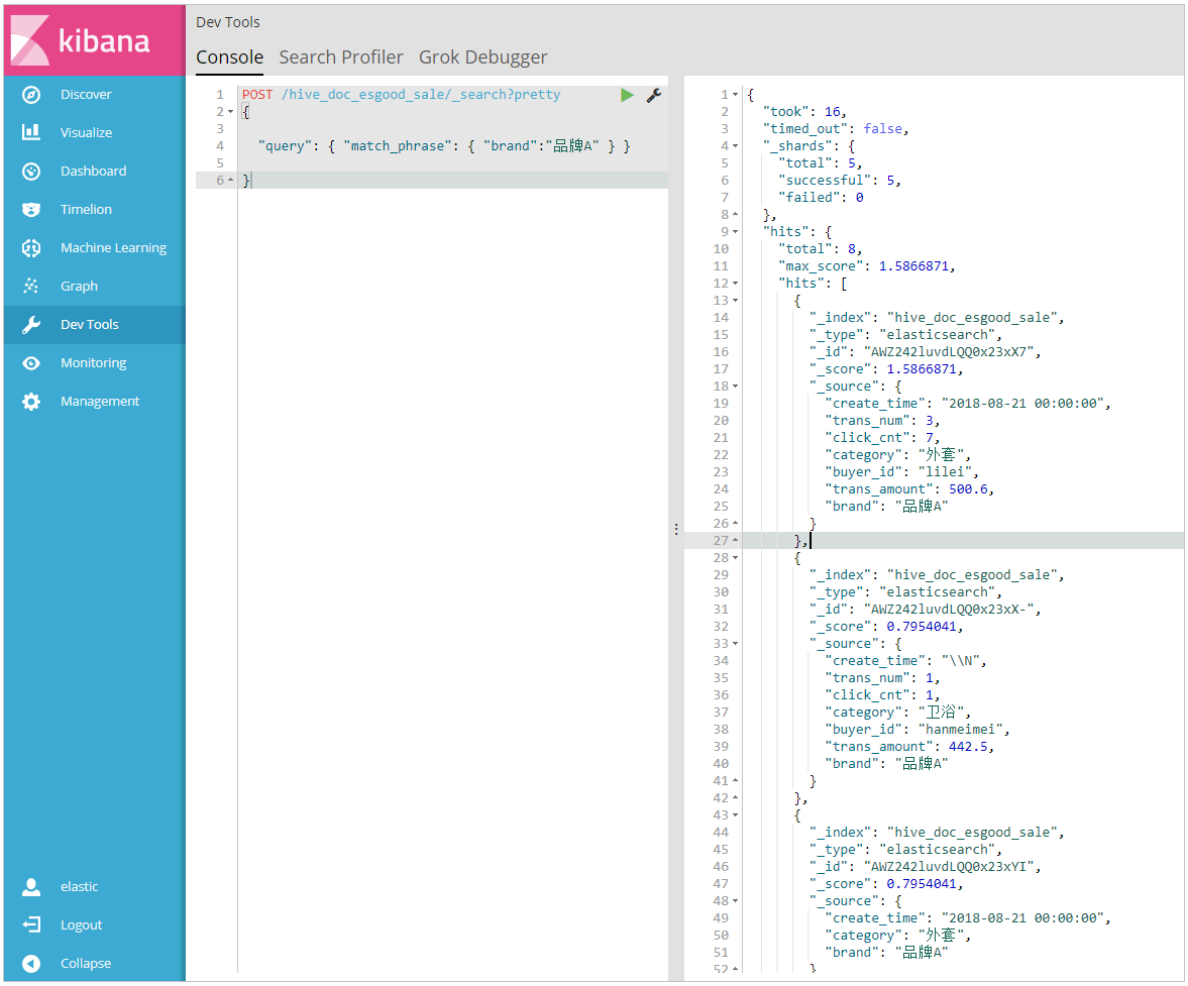


Data query and analysis

1. The following example returns all the documents of Brand A.

```
POST / hive_doc_e sgood_sale / _search ? pretty
{
" query ": { " match_phra se ": { " brand ":" Brand A " } }
```

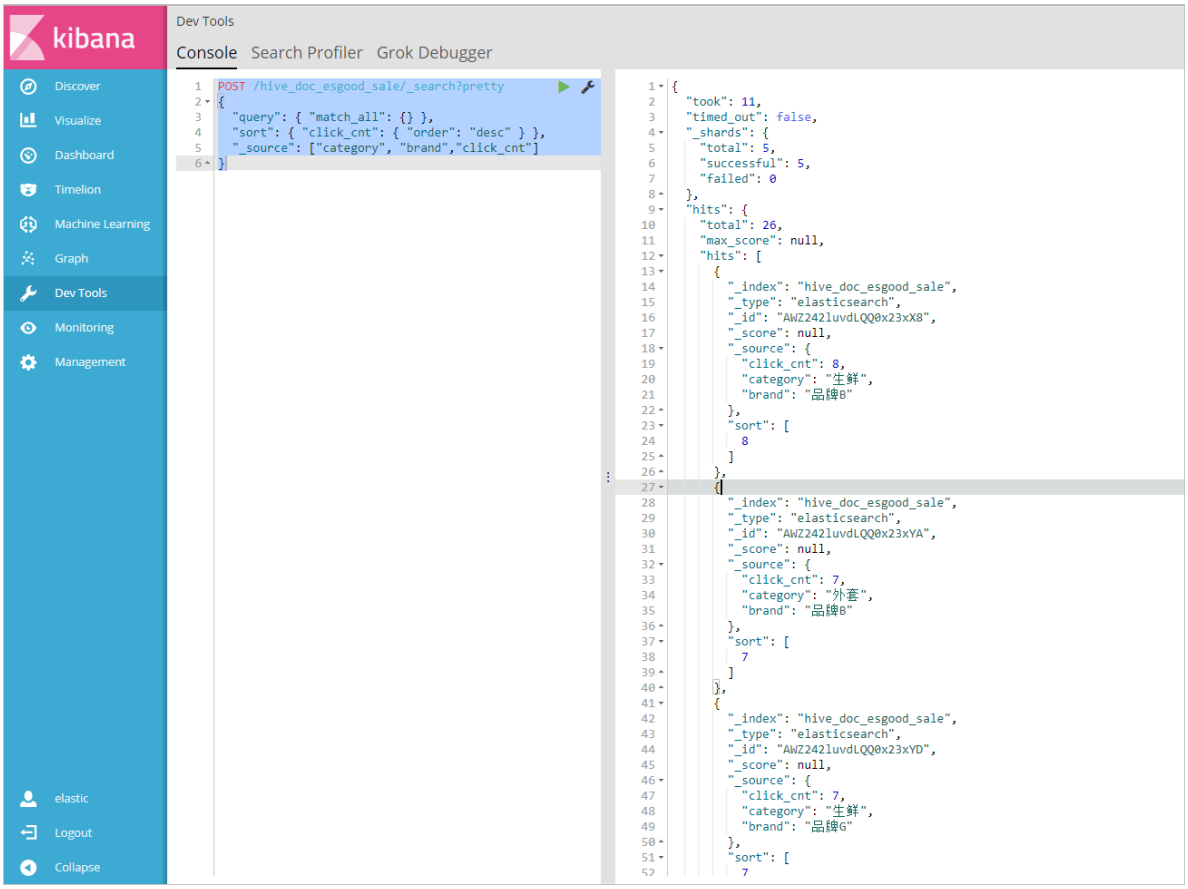
```
}
```



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

```
 POST  / hive_doc_e  sgood_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*.

3.3 Synchronize data from an ApsaraDB RDS for MySQL database to an Alibaba Cloud Elasticsearch instance, and query and analyze data

Alibaba Cloud provides you with a wide range of cloud storage and database services. If you want to analyze and search data stored in these services, use Data Integration to replicate the data to Alibaba Cloud Elasticsearch, and then query or analyze the data. Data Integration allows you to replicate data at a minimum interval of five minutes.

> **Note:**
> Data replication generates public network traffic and may incur fees.

Prerequisites

Perform the following tasks before you analyze or query the on-premises data:

- Create a database. You can use an ApsaraDB RDS for MySQL database, or create a database on your local server. This example uses an ApsaraDB RDS for MySQL database. The following figure shows the dataset stored in the database:

create_time	category	brand	buyer_id	trans_num	trans_amount	click_cnt
2018-08-21 00:00:00	Outside	B	l	3	500.6	7
2018-08-22 00:00:00	Raw	B	l	1	303	8
2018-08-22 00:00:00	Outside	B	h	2	510	2
1970-01-01 08:00:00	Guard	B	h	1	442.5	1
2018-08-22 00:00:00	Raw	B	h	2	234	3
2018-08-23 00:00:00	Outside	B	j	9	2000	7
2018-08-23 00:00:00	Raw	B	j	5	45.1	5
2018-08-23 00:00:00	Outside	B	j	5	100.2	4
2018-08-24 00:00:00	Raw	B	p	10	5560	7
2018-08-24 00:00:00	Guard	B	p	1	445.6	2
2018-08-24 00:00:00	Outside	B	r	3	777	3
2018-08-24 00:00:00	Guard	B	r	3	122	3
2018-08-24 00:00:00	Outside	B	r	1	62	7

- Purchase an Alibaba Cloud Elastic Compute Service (ECS) instance that is connected to the same VPC network as your Alibaba Cloud Elasticsearch instance. This ECS instance is used to retrieve data from data sources and run tasks to write the data to the Alibaba Cloud Elasticsearch instance. The tasks are dispatched by Data Integration.
- Activate Data Integration, and add the ECS instance to Data Integration as a resource to run synchronization tasks.
- Configure a data synchronization script and run the script periodically.
- Create an Alibaba Cloud Elasticsearch instance to store the data synchronized by Data Integration.

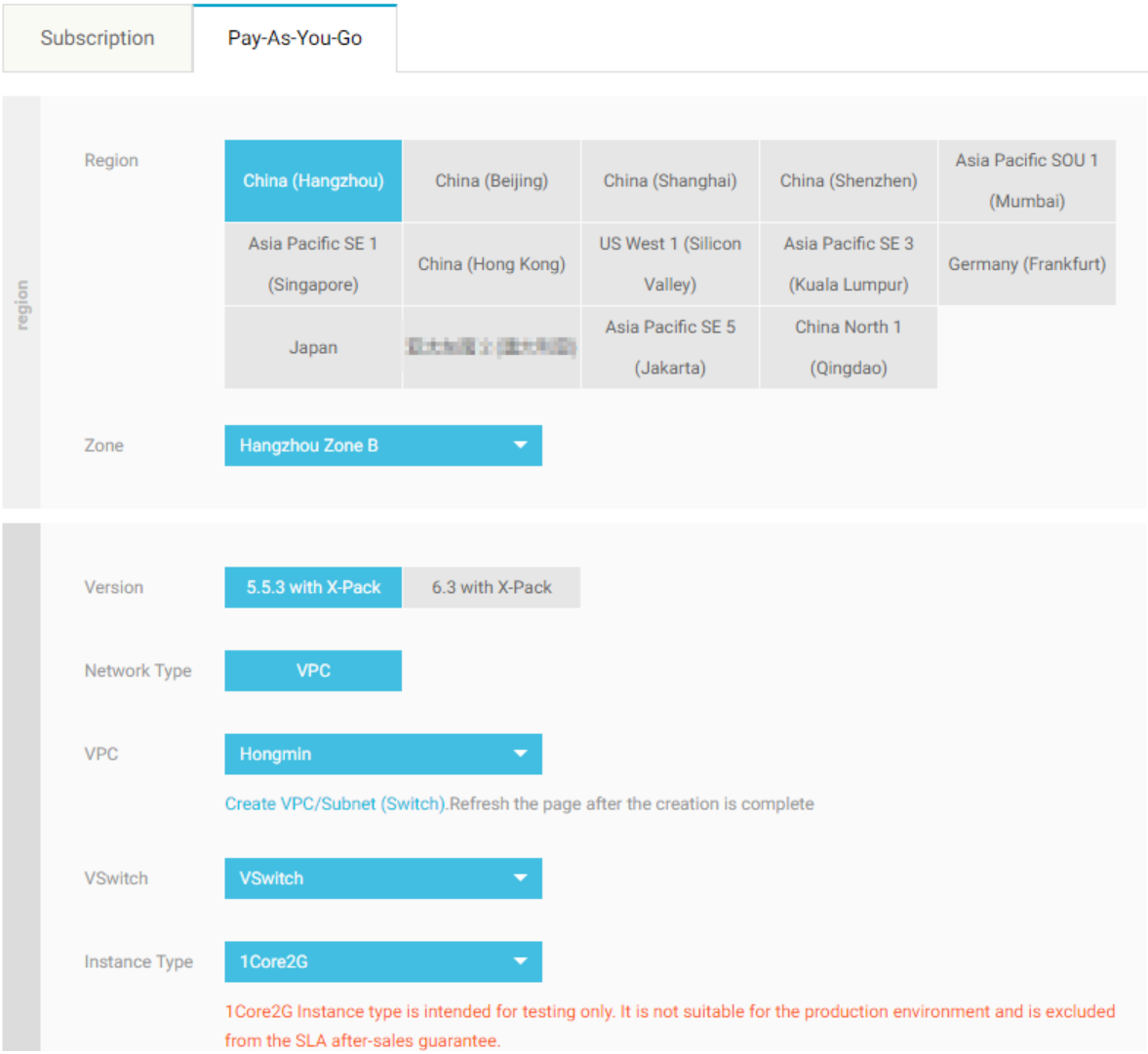
Procedure

Synchronize data

1. *Create a VPC*.
2. Log on to the *Alibaba Cloud Elasticsearch console* and click Create to create an Alibaba Cloud Elasticsearch instance.

Note:

The region, VPC network, and the VSwitch that you specify for the Alibaba Cloud Elaticsearch instance must be the same as those of the VPC network that you have created in the step 1.



3. Purchase an ECS instance that is connected to the same VPC network as the Alibaba Cloud Elasticsearch instance, and assign a public IP address or activate Elastic IP Address (EIP) to the ECS instance. To save costs, we recommend that you use an existing ECS instance that meets the requirements.

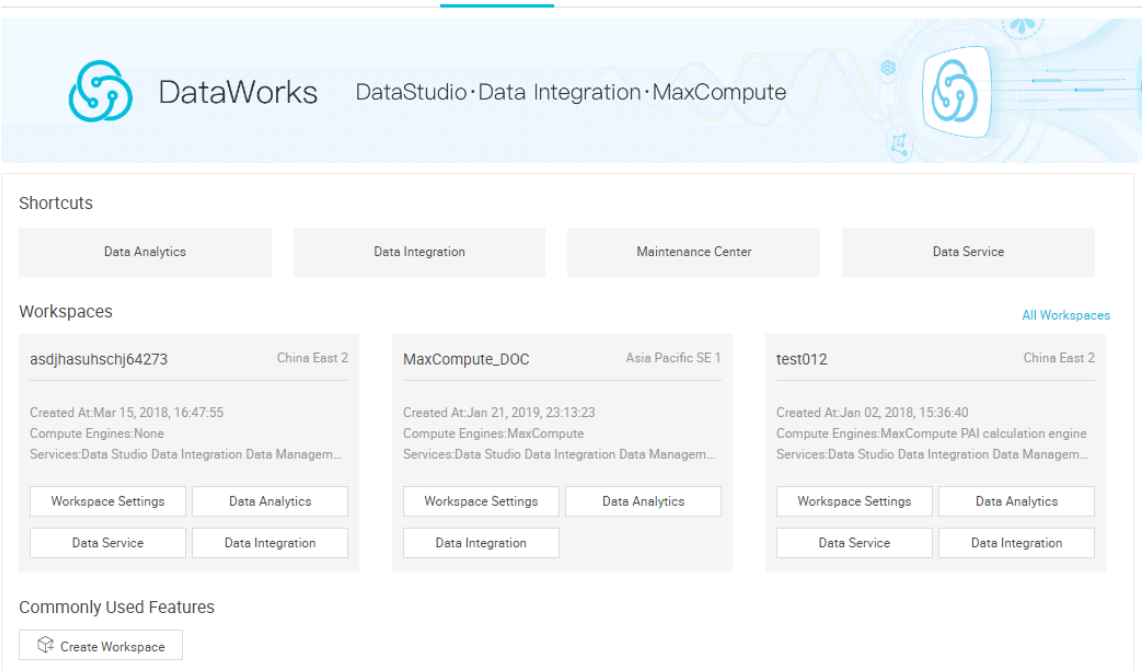
Note:

- We recommend that you use CentOS6, CentOS7, or AliyunOS.
- If the ECS instance needs to run MaxCompute or data synchronization tasks, you must verify that the current Python version of the ECS instance is 2.6 or 2.7 . The Python version of CentOS 5 is 2.4 while that of other CentOS versions is 2.6 or later.

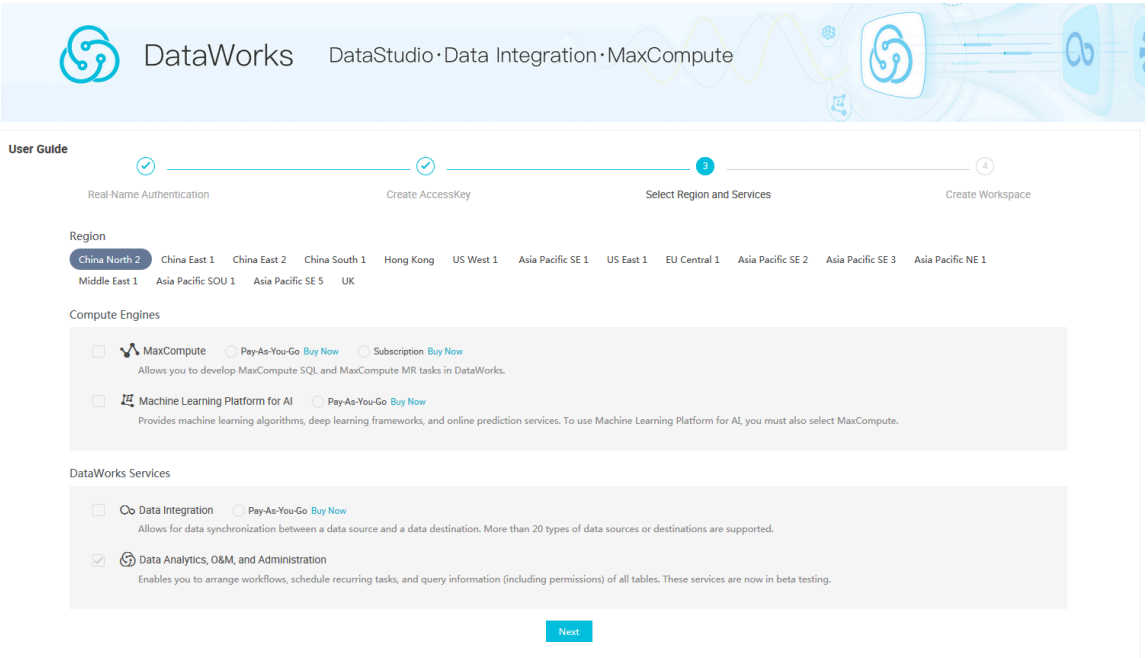
- Make sure that the ECS instance has a public IP address.

4. Log on to the *DataWorks console*.

   - The following page is displayed if you have already activated Data Integration:

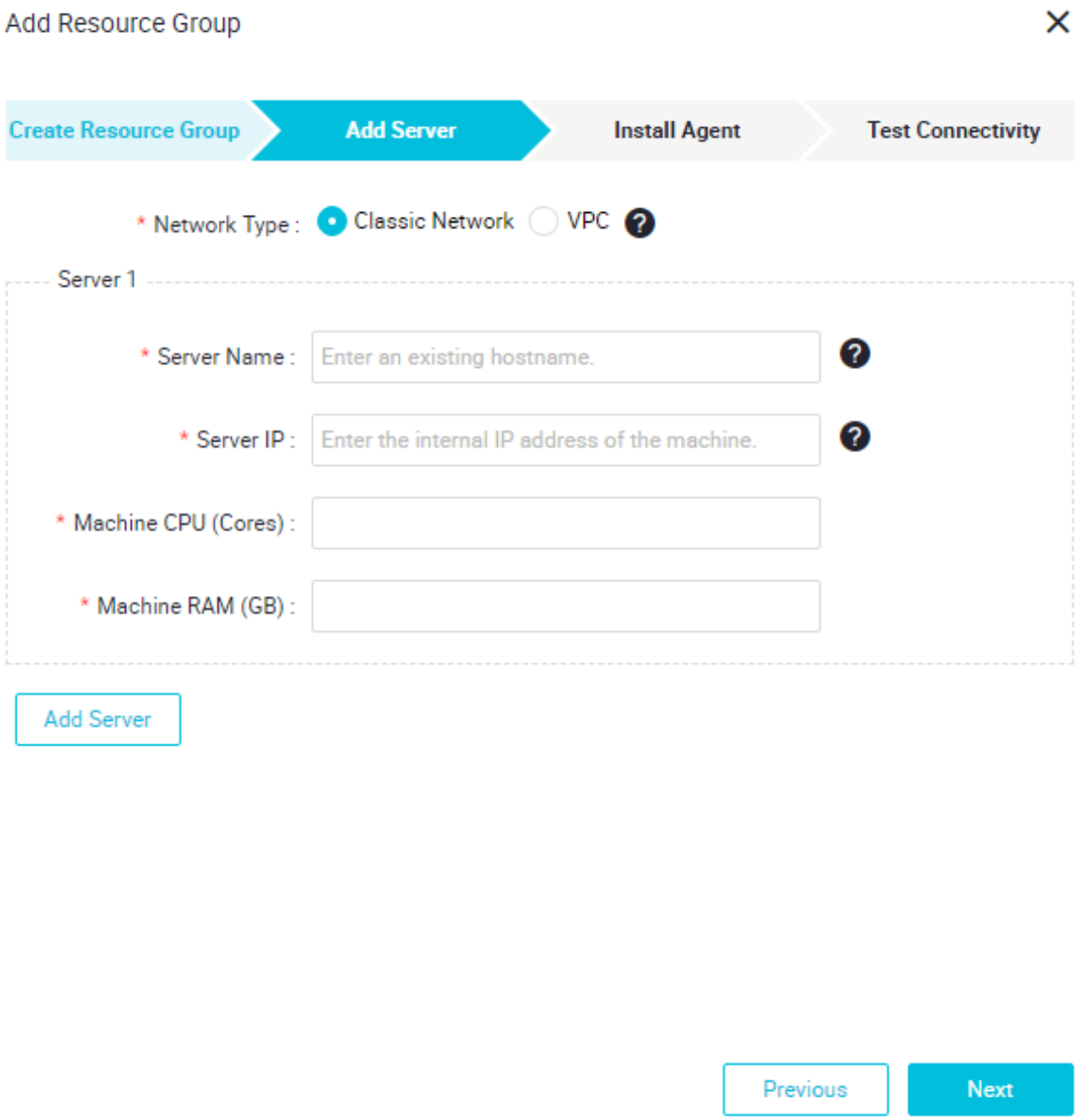


   - The following page is displayed if you have not activated Data Integration : Perform the following steps to activate Data Integration. Activating Data Integration incurs service fees. You can estimate the costs based on the billing items.



5. Click Data Integration.

6. On the Data Integration page, click Resource Group in the left-side navigation pane, and then click Add Resource Group in the upper-right corner.
7. Enter the resource group name and server information as required. The server you add on this page refers to the ECS instance that you have purchased. Enter the following information:



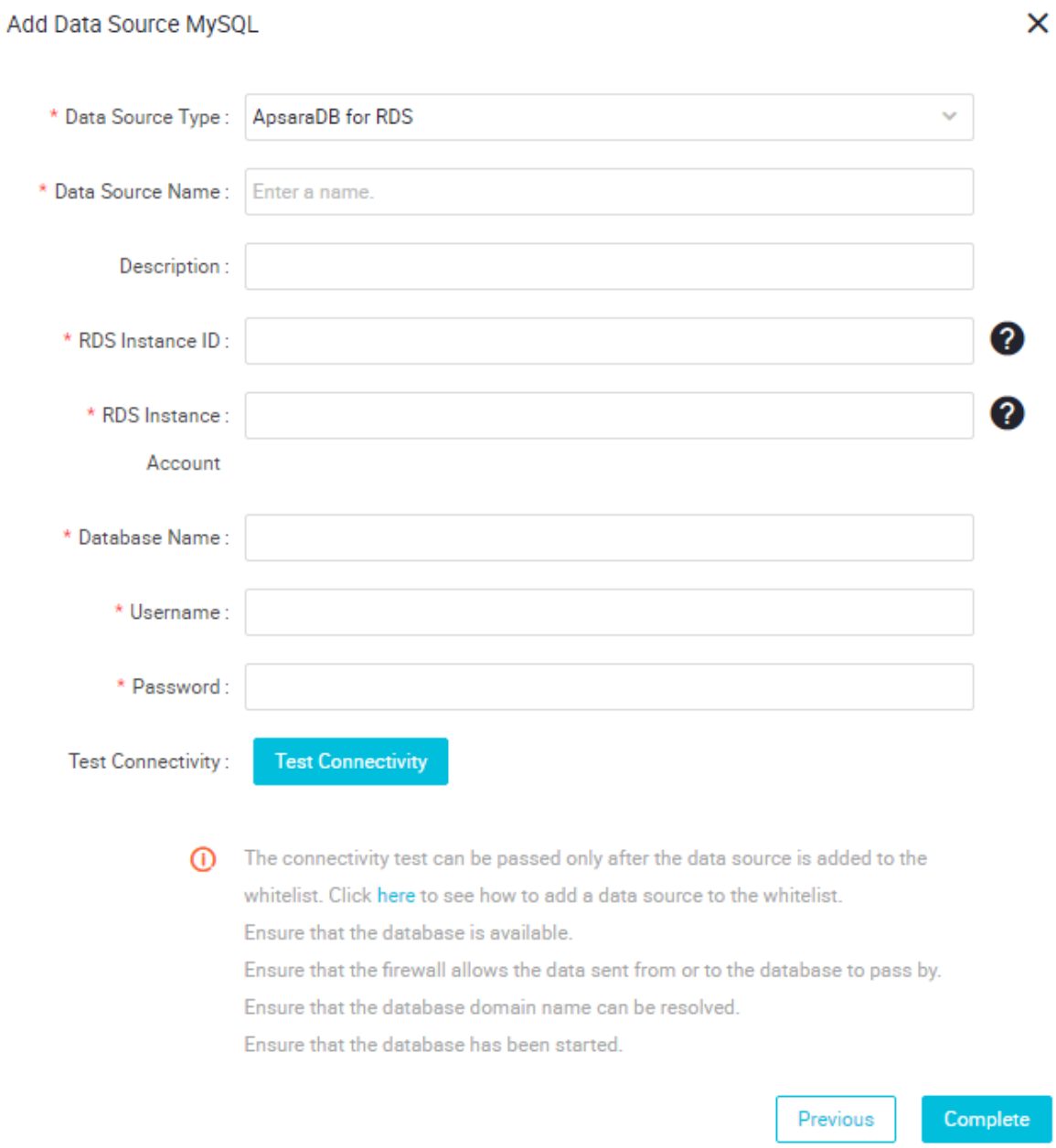
- ECS UUID: enter the UUID of the ECS instance. Log on to the ECS instance and run the `dmidecode | grep UUID` command to obtain the UUID. For more information, see *Step 3: Connect to an instance*.
- Server IP, Machine CPU (Cores), and Machine RAM (GB): enter the public IP address of the ECS instance, the CPU size, and the memory size. To obtain the

information, log on to the ECS console and click the ECS instance name. The information is listed in the Configuration Information area.

- Follow the instructions on the page to install an agent. Step 5 opens port 8000 of the ECS instance. You can use the default settings and skip this step.

8. Configure the database whitelist. Add the IP address of the resource group and the IP address of the ECS instance to the whitelist. For more information about whitelist configuration, see *Add whitelist*.
9. After you create the resource group, click Data Source in the left-side navigation pane, and then click Add Data Source in the upper-right corner.

10. Select MySQL. On the Add Data Source MySQL page, enter the required information, as shown in the following figure:



Data Source Type: this example uses an ApsaraDB RDS for MySQL database. You can select Public IP Address Available or Public IP Address Unavailable. For more information about the parameters, see *Configure MySQL data source*.

11. In the left-side navigation pane, click Sync Resources and then click Create Task. Select Script Mode.

12. In the Apply Template dialog box, choose Source Type > MySQL. Enter the name of the data source that you have added in step 10 in the Data Source field and select

Elasticsearch from the Destination Type drop-down list. Confirm the information and click OK.

13. Configure a data synchronization script For more information about the configuration, see *Script mode configuration*. For more information about Alibaba Cloud Elasticsearch instance configuration rules, see *Configure Elasticsearch Writer*.

```
1  {
2    "configuration": {
3      "reader": {
4        "plugin": "mysql",
5        "parameter": {
6          "datasource": "es_test_rdsmysql",
7          "column": [
8            "create_time",
9            "category",
10           "brand",
11           "buyer_id",
12           "trans_num",
13           "trans_amount",
14           "click_cnt"
15         ],
16         "where": "",
17         "splitPk": "",
18         "table": "good_sale"
19       }
20     },
21     "writer": {
22       "plugin": "elasticsearch",
23       "parameter": {
24         "accessId": "elastic",
25         "endpoint": "http://es-cn-[redacted] aliyuncs.com:9200",
26         "indexType": "elasticsearch",
27         "accessKey": "[redacted]",
28         "cleanup": false,
29         "discovery": false,
30         "column": [
31           {
32             "name": "create_time",
33             "type": "date"
34           },
35           {
36             "name": "category",
37             "type": "string"
38           },
39           {
40             "name": "brand",
41             "type": "string"
42           },
43           {
44             "name": "buyer_id",
45             "type": "string"
46           },
47           {
48             "name": "trans_num",
49             "type": "long"
50           },
51           {
52             "name": "trans_amount",
53             "type": "double"
54           },
55           {
56             "name": "click_cnt",
57             "type": "long"
58           }
59         ],
60         "index": "testrds",
61         "batchSize": 1000,
62         "splitter": ","
63       }
64     },
65     "setting": {
66       "errorLimit": {
67         "record": "0"
68       },
69       "speed": {
70         "throttle": false,
71         "concurrent": 1,
72         "mbps": "1",
73         "dmu": 1
74       }
75     }
```

Note:

- A data synchronization script includes three sections: the reader, writer, and settings. The reader sections contain the configuration of the data source (cloud resource) that stores the data to be synchronized. The writer section contains the configuration of the Alibaba Cloud Elasticsearch instance. The settings section contains data synchronization settings, such as the packet loss threshold and maximum concurrency.
- You can set the endpoint to the internal or public IP address of the Alibaba Cloud Elasticsearch instance. If you use the internal IP address, you must configure a system whitelist for the Alibaba Cloud Elasticsearch instance on the Elasticsearch Cluster Configuration page. If you use the public IP address, you must configure a whitelist on the Security page for the Alibaba Cloud Elasticsearch instance to allow visits from public IP addresses. The whitelist must include the *IP address of the ECS instance added to DataWorks* and the IP address of the resource group that you use.
- Set the accessId and accessKey parameters in the writer section to the username and password of the Alibaba Cloud Elasticsearch instance, respectively.
- Set the index parameter in the writer section to the index that of the Alibaba Cloud Elasticsearch instance. This index is used to access the data stored on the Alibaba Cloud Elasticsearch instance.

14. After you have configured the synchronization script, click Configure Resource Group on the right side of the page and select the resource group that you have created in step 7. Confirm and click Run to replicate data from the MySQL database to the Elasticsearch instance.

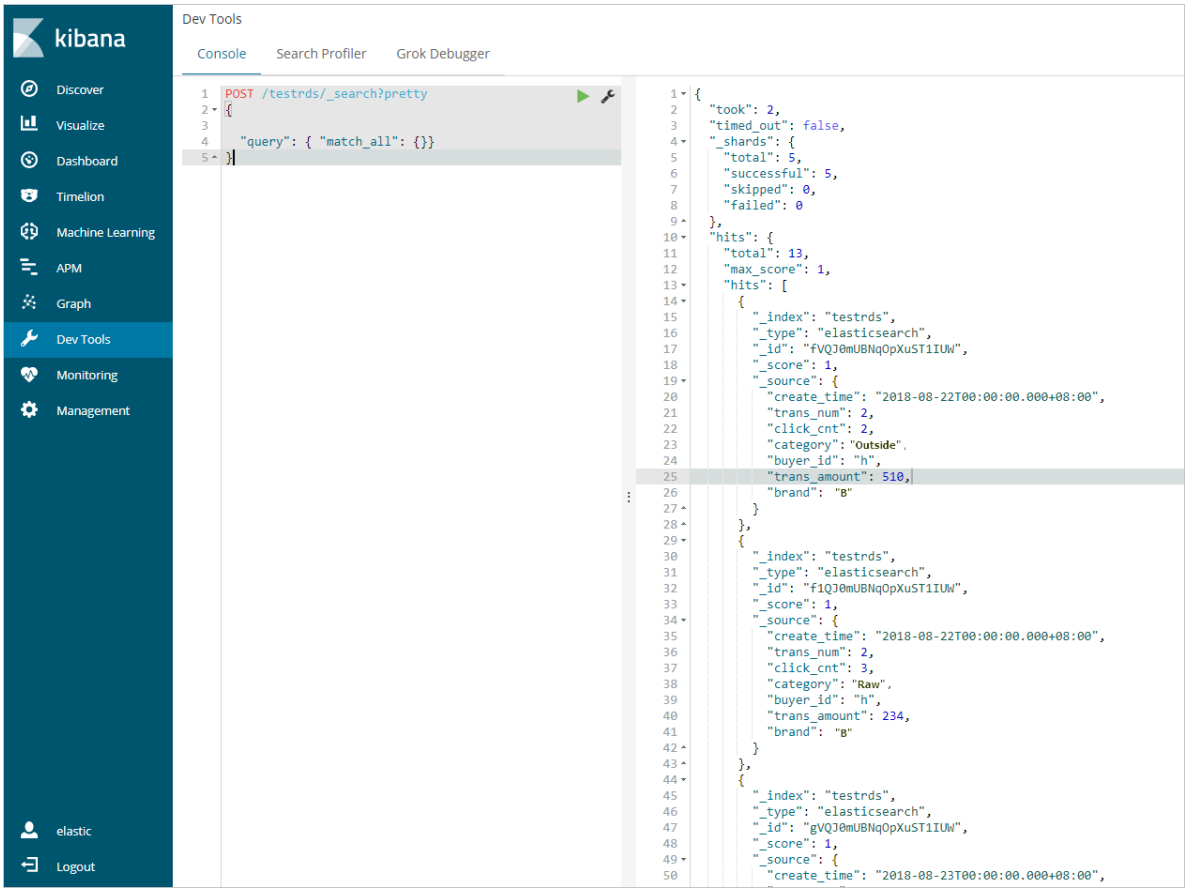
Query and analyze data

1. Log on to the Elasticsearch console, click Kibana Console in the upper-right corner, and then click Dev Tools.
2. Run the following command to view the synchronized data:

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

```
}
```

`testrds` is the value specified in the `index` parameter in the data synchronization script.



3. Run the following command to sort the data based on the `trans_num` column:

```
POST / testrds / _search ? pretty
{
" query ": { " match_all ": {} },
" sort ": { " trans_num ": { " order ": " desc " } }
}
```

4. Run the following command to query the `category` and `brand` columns in the data:

```
POST / testrds / _search ? pretty
{
" query ": { " match_all ": {} },
" _source ": [" category ", " brand "]
}
```

5. Run the following command to query data entries where the `category` column is set to Raw:

```
POST / testrds / _search ? pretty
{
" query ": { " match ": {" category ":" Raw "} }
```

```
}
```


```
{
  "took": 10,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 4,
    "max_score": 0.6931472,
    "hits": [
      {
        "_index": "testrds",
        "_type": "elasticsearch",
        "_id": "f1QJ0mUBNqOpXuST1IUW",
        "_score": 0.6931472,
        "_source": {
          "create_time": "2018-08-22T00:00:00.000+08:00",
          "trans_num": 2,
          "click_cnt": 3,
          "category": "Raw",
          "buyer_id": "h",
          "trans_amount": 234,
          "brand": "B"
        }
      },
      {
        "_index": "testrds",
        "_type": "elasticsearch",
        "_id": "gVQJ0mUBNqOpXuST1IUW",
        "_score": 0.6931472,
        "_source": {
          "create_time": "2018-08-23T00:00:00.000+08:00",
          "trans_num": 5,
          "click_cnt": 5,
          "category": "Raw",
          "buyer_id": "j",
          "trans_amount": 45.1,
          "brand": "B"
        }
      },
      {
        "_index": "testrds",
        "_type": "elasticsearch",
        "_id": "g1QJ0mUBNqOpXuST1IUW",
        "_score": 0.6931472,
        "_source": {
          "create_time": "2018-08-24T00:00:00.000+08:00",
          "trans_num": 10,
```

For more information about how to access Elasticsearch, see *Elasticsearch access test* and *Elastic documentation*.

FAQ

- An error occurred while accessing the database.

  Solution: Add the internal and public IP addresses of the ECS instance that in the resource group to the DataWorks database whitelist.

- An error occurred while accessing the Alibaba Cloud Elasticsearch instance.

  Solution: perform the following steps:

  1. Check whether you have selected the resource group created in the preceding step from Configure Resource Group.
     - Go to the next step if you have selected the correct resource group.
     - If you have not selected the correct resource group, click Configure Resource Group to select the correct one. Confirm and click Run.
  2. Check whether you have added the *IP address of the ECS instance* and the IP address of the resource group to the whitelist of the Elasticsearch instance.
     - Go to the next step if you have added these IP addresses to the whitelist.
     - If you have not added these IP addresses to the whitelist, add the *IP address of the ECS instance* and the IP address of the resource group to the whitelist of the Elasticsearch instance.

     > Note:
     > If you use the internal IP address, configure a system whitelist for the Elasticsearch instance on the Security page. If you use the public IP address, configure a whitelist for the Elasticsearch instance on the Security page to allow visits from public IP addresses. The whitelist must include the *IP address of the ECS instance* and the IP address of the resource group.

  3. Check whether the configuration of the script is correct Check the endpoint, accessId, and accessKey. The endpoint must be set to the internal or public IP address of the Elasticsearch instance. The accessId must be set to the username of the Elasticsearch instance. The default name is elastic. The accessKey must be set to the password of the Elasticsearch instance.

3.4 Real-time data synchronization from RDS for MySQL to ES

This section explains how to use *Data Transmission Service (DTS)* to quickly create a real-time data synchronization task from an RDS for MySQL instance to an Alibaba Cloud Elasticsearch (ES) instance. DTS uses this synchronization feature to synchronize RDS for MySQL data to ES instances and query data in real time.

Real-time synchronization type

DTS instances under the same Alibaba Cloud account from RDS for MySQL to ES.

SQL operation types

The main SQL operation types supported are as follows:

- Insert
- Delete
- Update

> **Note:**
> DTS does not support using DDL statements to synchronize data. DDL operations are ignored when data is synchronized.

If a table using DDL is encountered in an RDS for MySQL instance, the DML operations for the corresponding table may fail. To resolve this problem, complete the following steps:

1. Delete the object from the synchronization list. For more information, see *Delete synchronization objects*.
2. Delete the index corresponding to this table in the ES instance.
3. Re-add the table to the synchronization list and re-initialize it. For more information, see *Add a synchronization object*.

If the DDL is used to add a column or modify a table, the order of DDL operations is as follows:

1. Manually modify the corresponding mapping and new column in your ES instance.
2. Modify the table schema and add a new schema in the source RDS for MySQL instance.
3. Stop synchronizing instances in DTS, and restart DTS synchronization instances to reload the mapping relationship that was modified in ES.

Configure data synchronization

To synchronize data from an RDS for MySQL instance to an ES instance, complete these steps:

1. Purchase a DTS synchronization instance

   Log on to the *Data Transmission Service console* and go to the Data Synchronization pane. In the upper-right corner, click Create Synchronization Task to purchase a synchronization instance. You can then configure the synchronization instance.

   Note:

You must purchase a synchronization instance before you can configure it. Two billing modes are supported: Subscription and Pay-As-You-Go.

Purchase page parameters

- Function

  Select Data Synchronization.
- Source Instance

  Select MySQL.
- Source Region
  - Because this example uses the RDS for MySQL instance, you need to select the region where the RDS for MySQL instance is located.
- Target Instance

  Select Elasticsearch.
- Target Region

  Select the region where your Elasticsearch instance is located. Note that after the synchronization instance has been purchased, you cannot change its region. Target Instance
- Specification

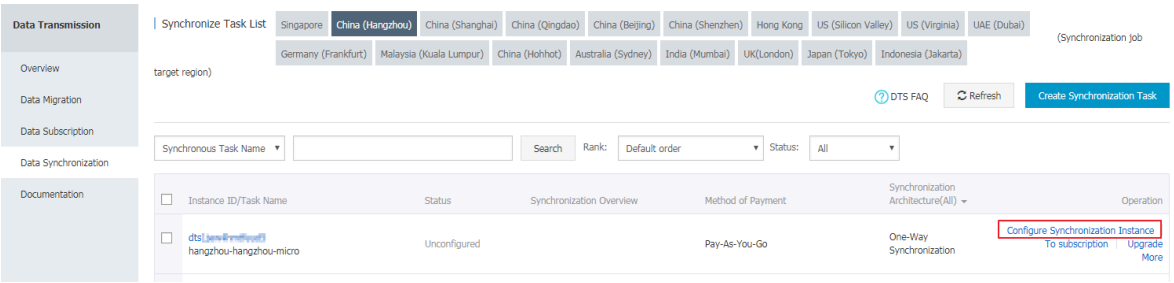
  Each instance specification corresponds to the performance of a synchronization instance. For more information, see *Data synchronization specifications*.
- Order Time
  - If the synchronization instance is prepaid, the order time is one month by default.
- Quantity

  By default, the quantity is 1.

  Note:
  The region of your DTS synchronization instance is the target region that you selected. For example, if the synchronization instance is from the Hangzhou-region RDS for MySQL to the Hangzhou-region Elasticsearch, the region of the DTS synchronization instance is Hangzhou. To configure your synchronization instance, go to the instance list in that region in DTS, search

for the synchronization instance you just purchased, and click Configure Synchronization Instance in the upper-right corner.

2. Configure your synchronization instance

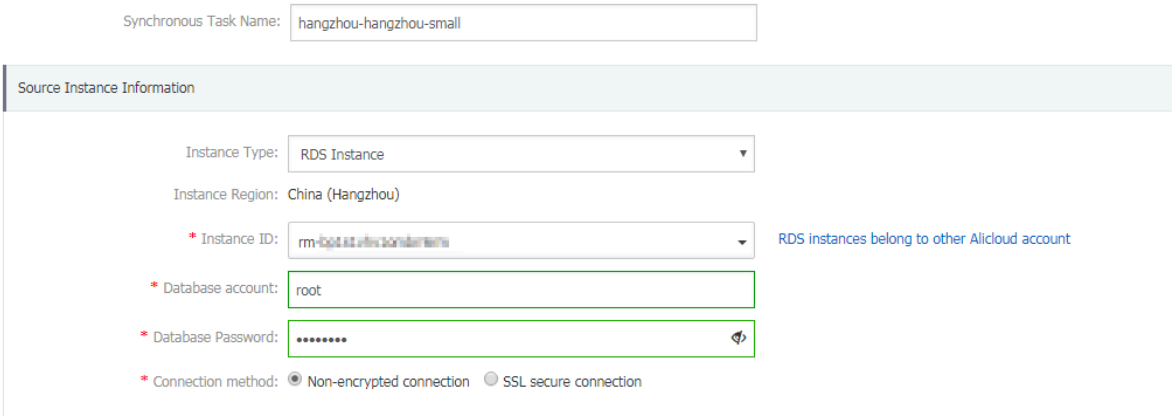


Synchronization task name

There are no requirements for the name of a synchronization instance.

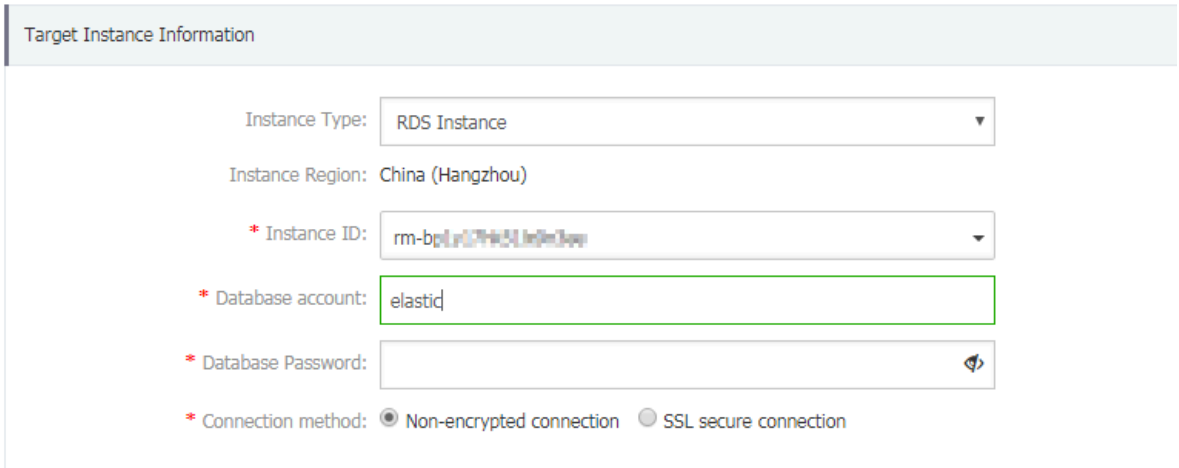
Source instance

This example uses RDS for MySQL as the data source. You need to set the instance type, region and ID, and database account and password.



Target instance

You need to configure the ID, account, and password for the ES instance.



Once you complete these configurations, click Authorize Whitelist and Enter Next Step to add IPs to RDS for MySQL and ES instance whitelists.

3. Authorize instance whitelists

> Note:
> If the source instance is RDS for MySQL, DTS automatically adds IPs to a whitelist or adds a security group.

If the source instance is RDS for MySQL, DTS adds the instance IP to the security group of an RDS instance' s whitelist. This means that, when creating synchronization tasks, you can avoid failures caused by a disconnection between the DTS instance and the RDS database. To ensure the stability of the synchronization task, do not delete the instance IP from the RDS instance.

After the whitelist is authorized, click Next to create a synchronization account.

4. Select the synchronization object

To configure synchronization objects and naming rules for indexes, complete these steps:

a. Select a naming rule for indexes: table name or database name_table name.

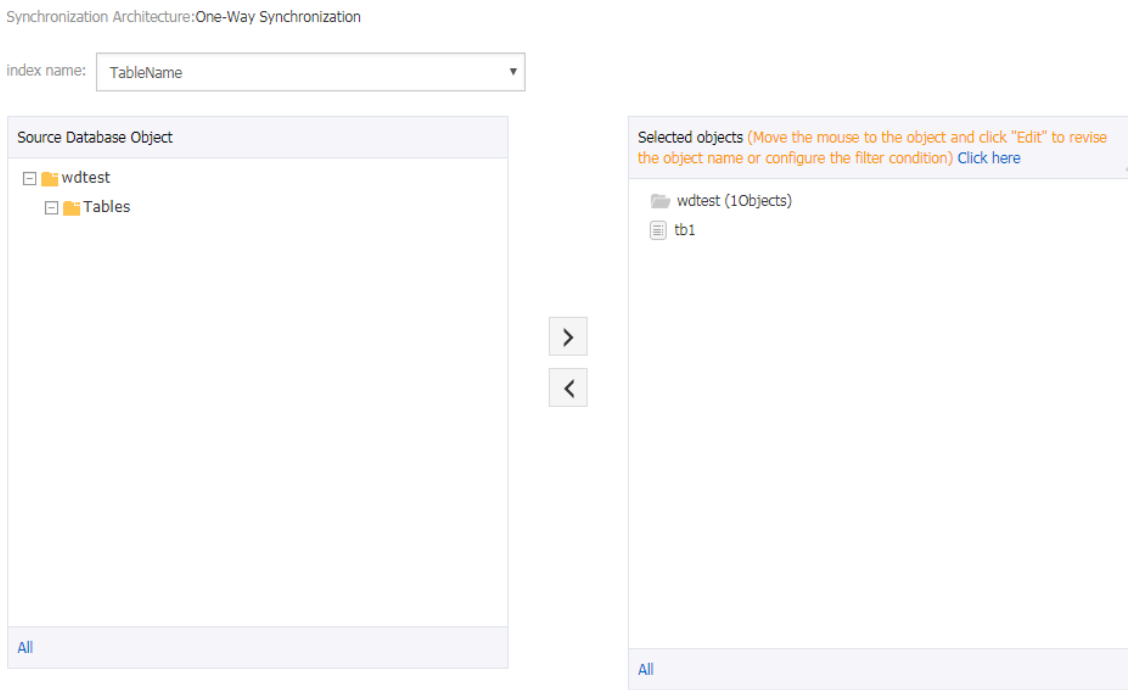
- If you select a table name, the name of the index is the name of the table.
- If you select a database name_table name, the naming rule for the index is database name_table name. For example, if a database is named dbtest and a

table is named sbtest1, after the table is synchronized to your ES instance, the index name would be dbtest_sbtest1.

- If two tables in different databases have the same name, we recommend that the index name be set to database name_table name.

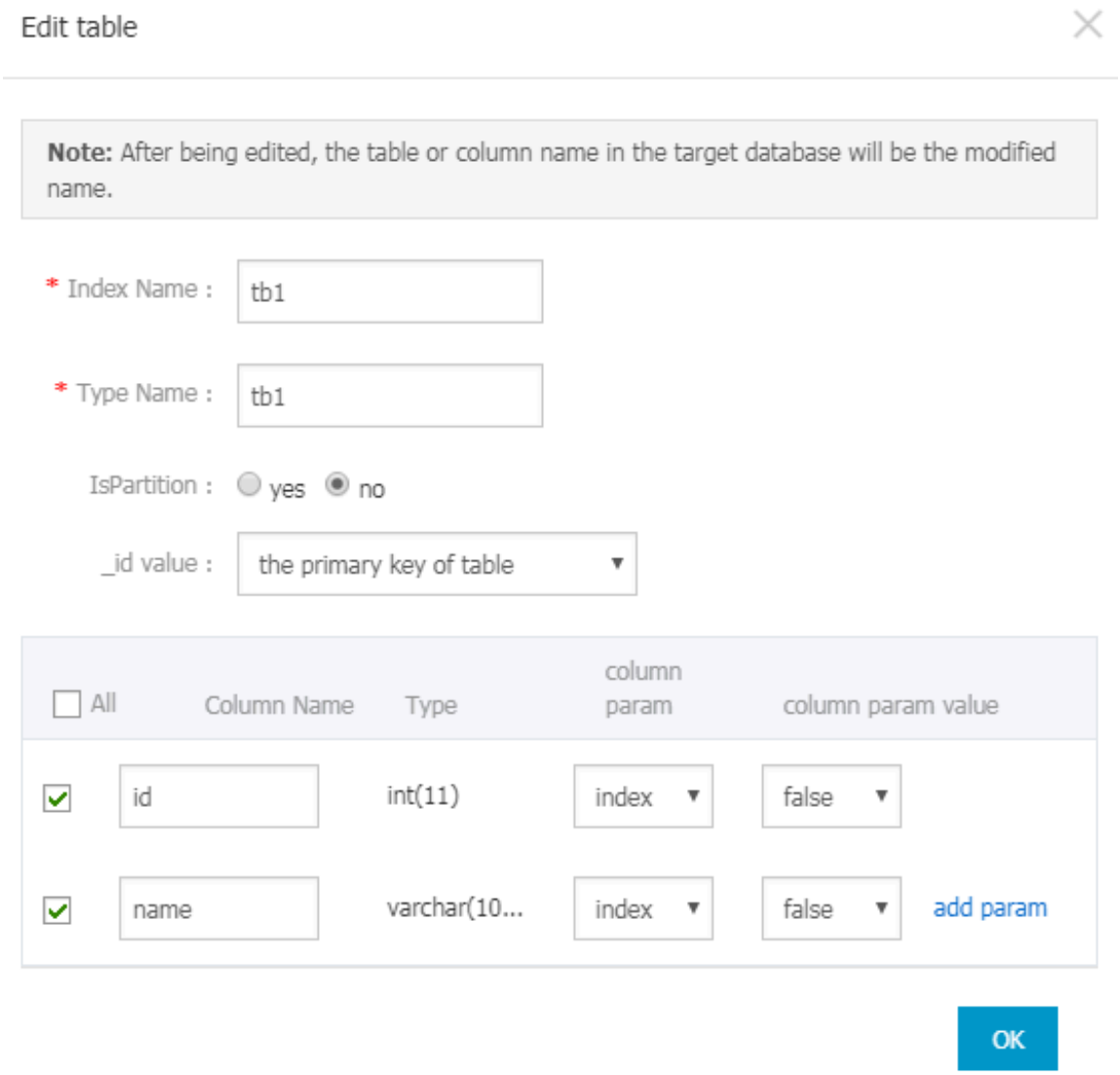
b. Select a specific database, table, and column. The selectable granularity of the synchronization objects supports table-level operations. This means that you can synchronize several databases and tables.

The selectable granularity of the synchronization objects supports table-level operations. This means that you can synchronize several databases and tables.



c. By default, the docid of all tables is the primary key. If some tables do not have the primary key, configure their docid corresponding to the columns in the

source tables. In the box of selected objects on the right, move the pointer over the corresponding table, and click Edit to enter the advanced settings pane.



d. In advanced settings, you can configure the index name, type name, partition column and quantity, and _id value column. If the value of _id is set to the business primary key, you need to select the corresponding business primary key column.

e. After synchronization objects are configured, proceed to the advanced setup.

5. Advanced setup

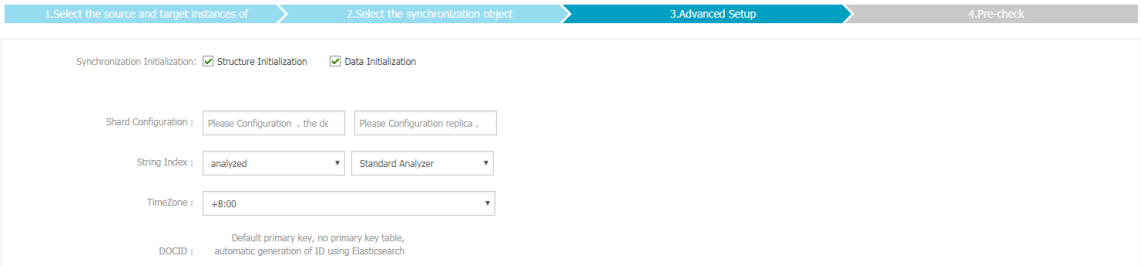
   Main configurations

   a. Synchronization Initialization: We recommend that you select Structure Initialization and Data Initialization, which allows DTS to automatically create indexes and initialize data. If you do not select Schema Initialization, you need

to define the mapping for indexes in ES manually before synchronizing. If you do not select Full Data Initialization, the starting time for incremental DTS data synchronization is the time at which synchronization starts.

b. Shard Configuration: There are 5 partitions and 1 replica by default. Once the configuration is adjusted, all indexes define partitions according to this configuration.

c. String Index is an analyzer that can select strings. By default, it is Standard Analyzer. Other values include: Simple Analyzer, Whitespace Analyzer, Stop Analyzer, Keyword Analyzer, English Analyzer, and Fingerprint Analyzer. The string fields of all indexes define Analyzer according to this configuration.



d. Time Zone is where time fields synchronized to your ES instance are stored. The default time zone in China is UTC (UTC +8).

6. Pre-check

   After synchronization task configurations are complete, DTS performs a pre-check. If the pre-check is verified, click Start to start the synchronization task.

   After the synchronization task starts, go to the synchronization job list and verify whether the task’s status is Sync initialization. The time it takes to initialize depends on the amount of data that the synchronization object has in the source instance. After completing the initialization, the synchronization instance’s status is Synchronizing. The synchronization link between the source and target instances is established.

7. Validate data

   After completing all of the preceding steps, log on to the ES console to check the corresponding indexes created in your ES instances and the synchronized data.

3.5 Synchronize data between MaxCompute and Elasticsearch with DataWorks

Alibaba Cloud provides you with a wide range of cloud storage and database services. If you want to analyze and search data in these services, use Data Integration to replicate your on-premises data to Alibaba Cloud Elasticsearch, and then search or analyze the data. Data Integration allows you to replicate data at a minimum interval of five minutes.

> **Note:**
> Data replication generates public network traffic and may incur fees.

Prerequisites

Follow these steps to analyze and search on-premises data:

- *Create and view a table*, and *import data*. You can *migrate data from Hadoop to MaxCompute*, and then synchronize the data. This example uses the following table schemes and data:

Column Name	Type
create_time	STRING
category	STRING
brand	STRING
buyer_id	STRING
trans_num	BIGINT
trans_amount	DOUBLE
click_cnt	BIGINT
pt	STRING

create_time	category	brand	buyer_id	trans_num	trans_amount	click_cnt	pt
2018-08-21 00:00:00	[illegible]	[illegible]A	null	null	null	null	1
2018-08-22 00:00:00	[illegible]	[illegible]B	null	null	null	null	1
2018-08-22 00:00:00	[illegible]	[illegible]C	null	null	null	null	1
	[illegible]	[illegible]A	null	null	null	null	1
2018-08-22 00:00:00	[illegible]	[illegible]D	null	null	null	null	1
2018-08-23 00:00:00	[illegible]	[illegible]B	null	null	null	null	1
2018-08-23 00:00:00	[illegible]	[illegible]A	null	null	null	null	1
2018-08-23 00:00:00	[illegible]	[illegible]E	null	null	null	null	1
2018-08-24 00:00:00	[illegible]	[illegible]G	null	null	null	null	1
2018-08-24 00:00:00	[illegible]	[illegible]F	null	null	null	null	1
2018-08-24 00:00:00	[illegible]	[illegible]A	null	null	null	null	1
2018-08-24 00:00:00	[illegible]	[illegible]G	null	null	null	null	1
2018-08-24 00:00:00	[illegible]	[illegible]C	null	null	null	null	1

- Create an Alibaba Cloud Elasticsearch instance to store the data that is successfully replicated by Data Integration.
- Purchase an Alibaba Cloud ECS instance that shares the same VPC with Alibaba Cloud Elasticsearch. This ECS instance will obtain data and execute Elasticsearch tasks (these tasks will be sent by Data Integration).
- Activate Data Integration, and register the ECS instance with Data Integration as a resource that can execute tasks.
- Configure a data synchronization script and periodically run the script.

Procedure

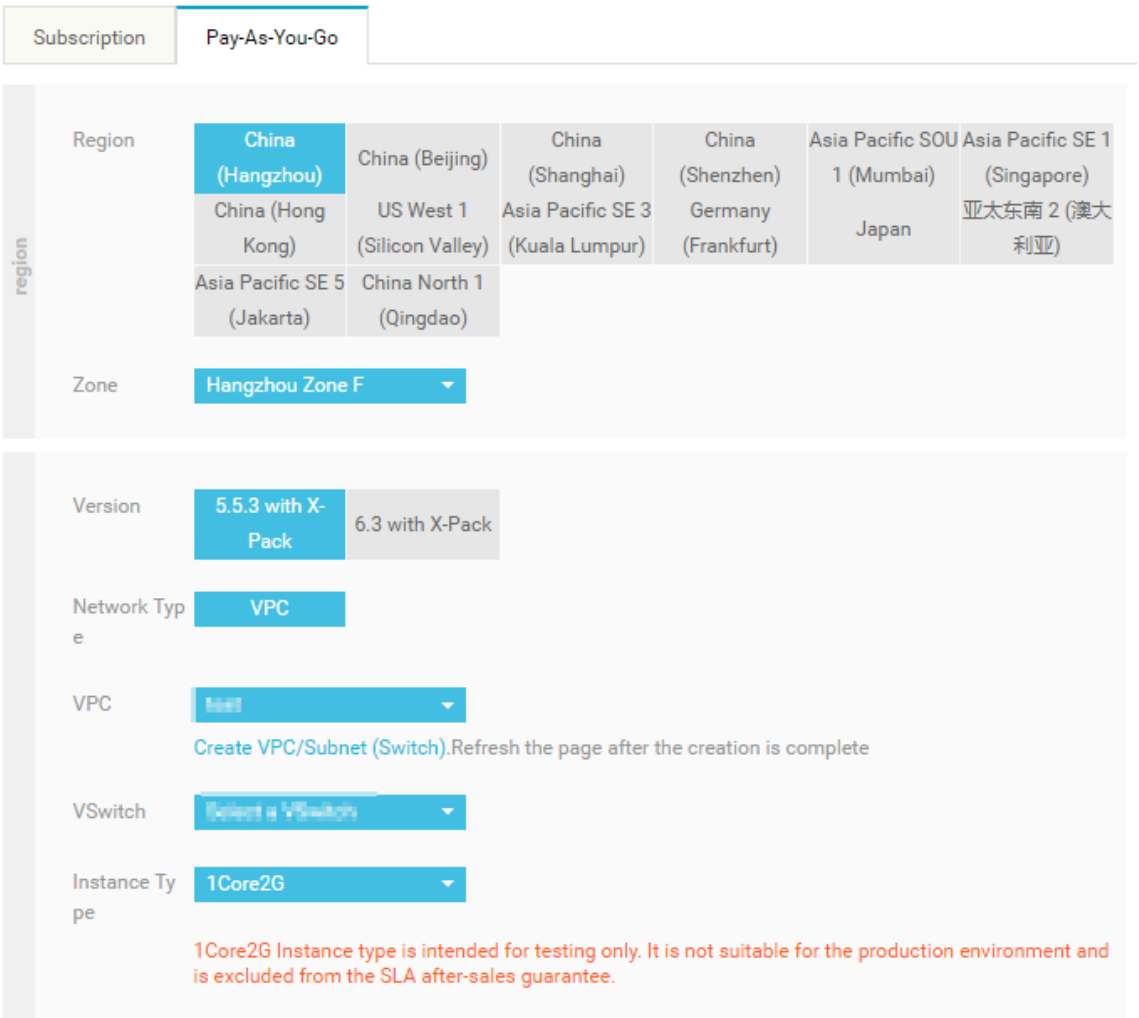
1. Create Alibaba Cloud Elasticsearch and ECS instances

   a. *Create a VPC*. This example creates a VPC in the China (Hangzhou) region. The instance name is es_test_vpc, and the corresponding VSwitch name is es_test_switch.

   b. Log on to the *Alibaba Cloud Elasticsearch console*, and create an Alibaba Cloud Elasticsearch instance.

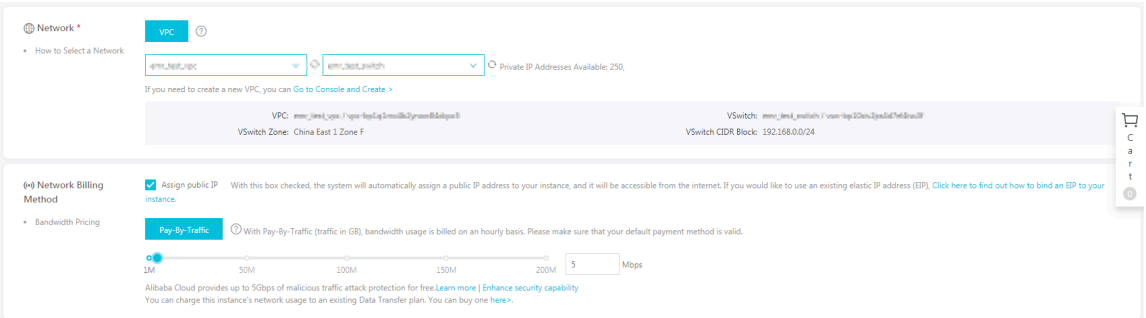
   Note:

Make sure that you select the same region, VPC, and VSwitch with the VPC that you have created in the preceding step.



c. Purchase an ECS instance that is in the same VPC as your Elasticsearch instance , and assign a public IP address or activate EIP. To save costs, we recommend that you use an existing ECS instance that meets the requirements.

This example creates an ECS instance in Zone F of China (Hangzhou). Select 64-bit CentOS 7.4 and Assign Public IP to configure network settings, as shown in the following figure:



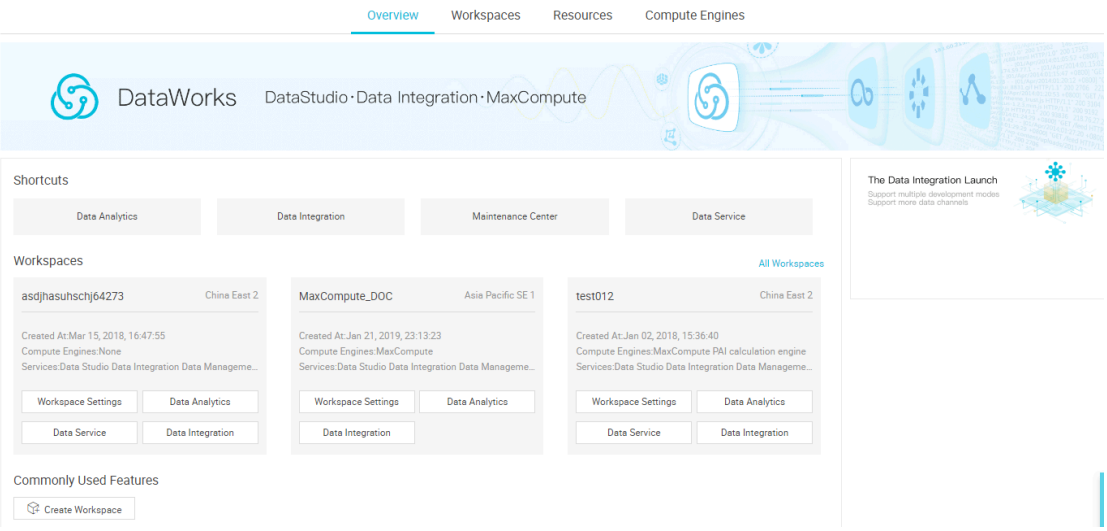
Note:

- We recommend that you use CentOS 6, CentOS 7, or Aliyun Linux.
- If the ECS instance that you have created needs to execute MaxCompute tasks or data synchronization tasks, you must verify that the version of Python running on the instance is either Python 2.6 or 2.7. When you install CentOS 5, Python 2.4 is also installed. Other versions of CentOS include Python 2.6 and later.
- Make sure that your ECS instance is assigned a public IP address.

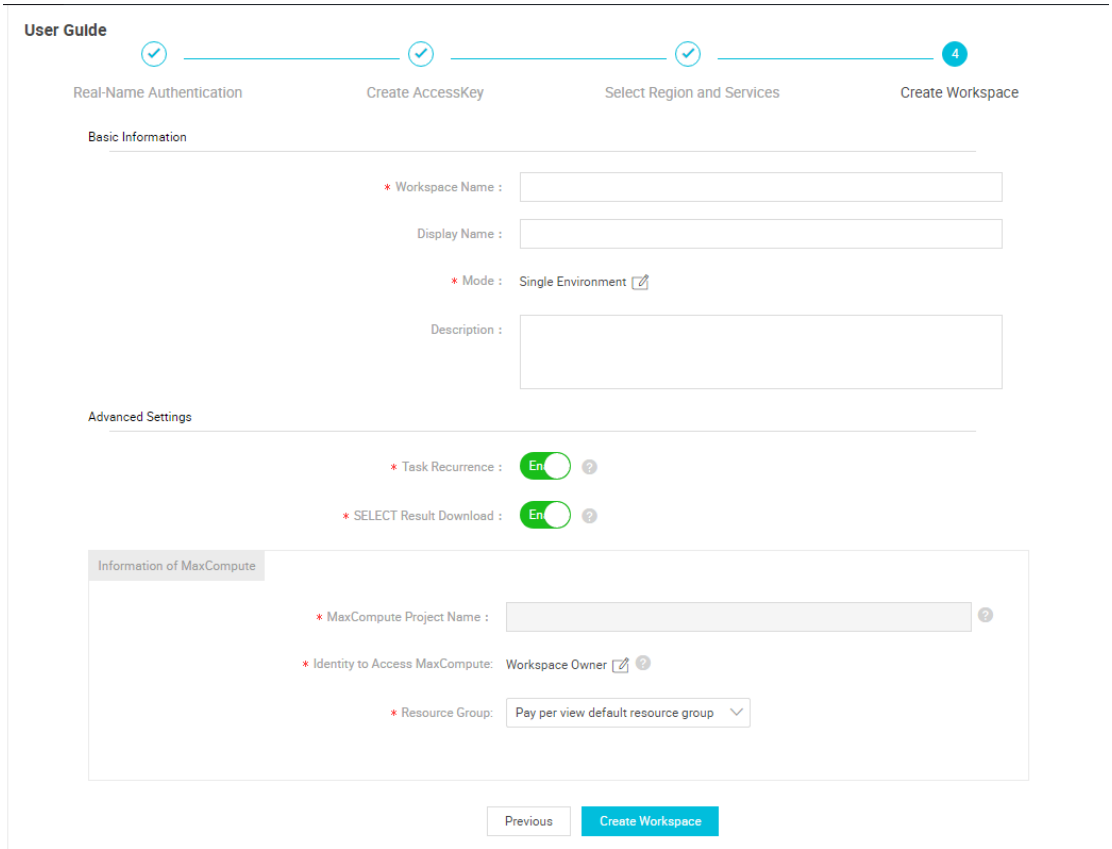
2. Configure data synchronization

   a. Log on to the *DataWorks console* to create a project. This example uses a DataWorks project named bigdata_DOC.

      - If you have already activated Data Integration, the following page is displayed:



      - If you have not activated Data Integration, the following page is displayed: You must follow these steps to activate Data Integration. Activating this service incurs fees, which you can estimate based on the pricing rules.



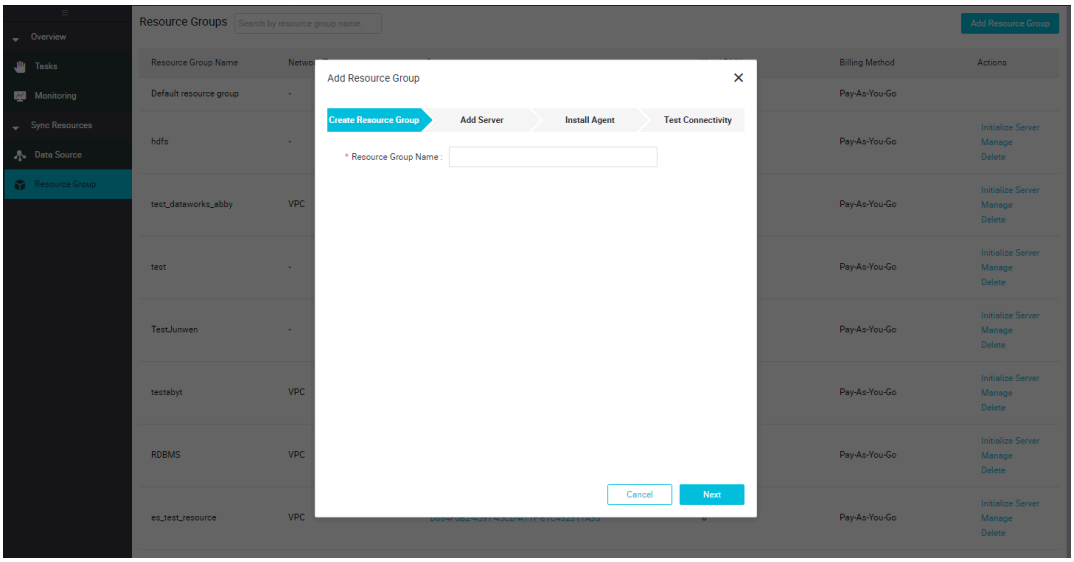
b. Click Data Integration under the DataWorks project.

c. Create resource group

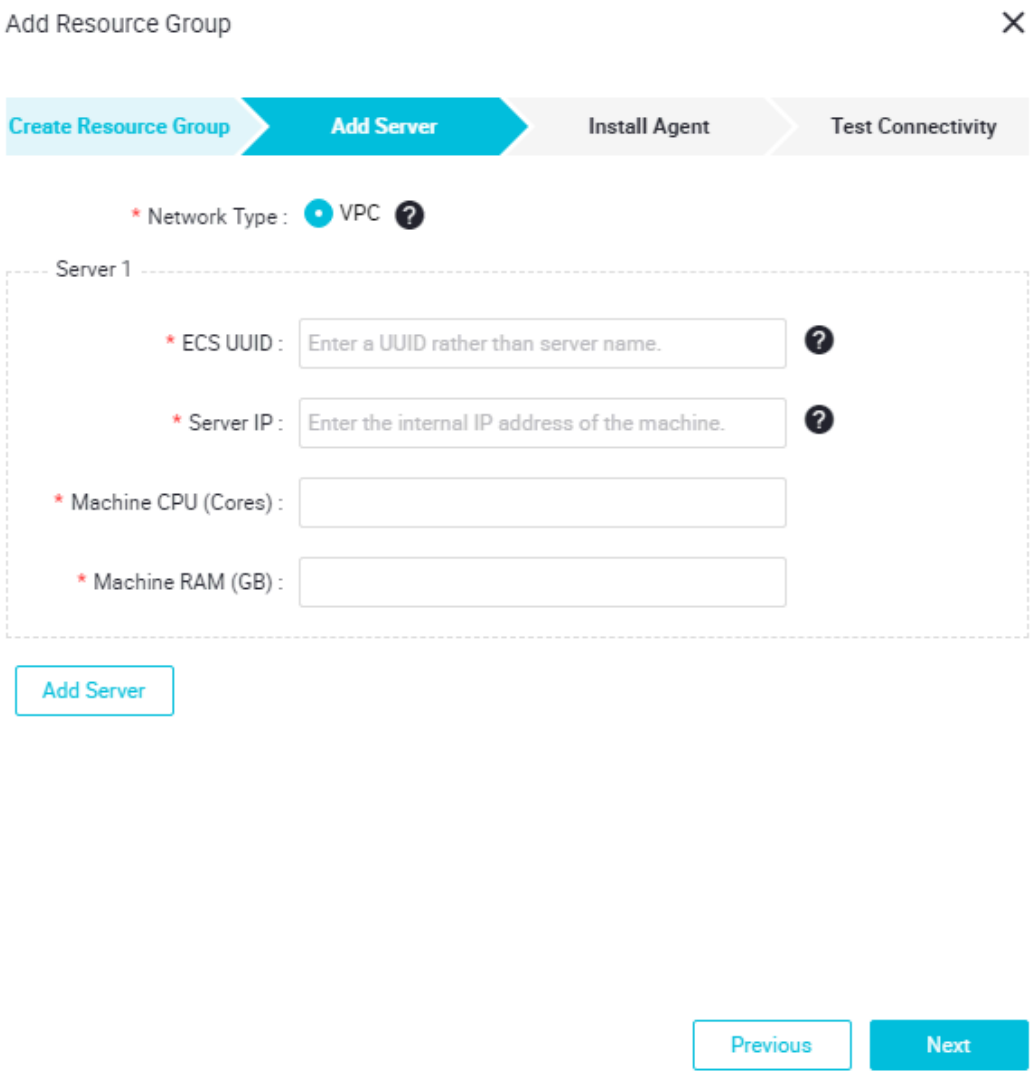
A. On the Data Integration page, select Resource Groups in the left-side navigation bar, and click Add Resource Group.

B. Follow these steps to add a resource group:

A. Create a resource group: Enter a resource group name. This example names the resource group as es_test_resource.



B. Add a server.



- ECS UUID: *Step 3: Connect to an instance*. Log on to the ECS instance, and run the `dmidecode | grep UUID` command to obtain a returned value].

- Machine IP/Machine CPUs (Cores)/Memory Size (GB): Specify the public IP address, CPU cores, and memory size of the ECS instance. Log on to

the ECS console, and click the name of the instance to view the relevant information in the Configuration Information module.

C. Install an agent: Complete the installation of Agent following these steps. This example uses a VPC. Therefore, you do not need to open port 8000 for the instance.

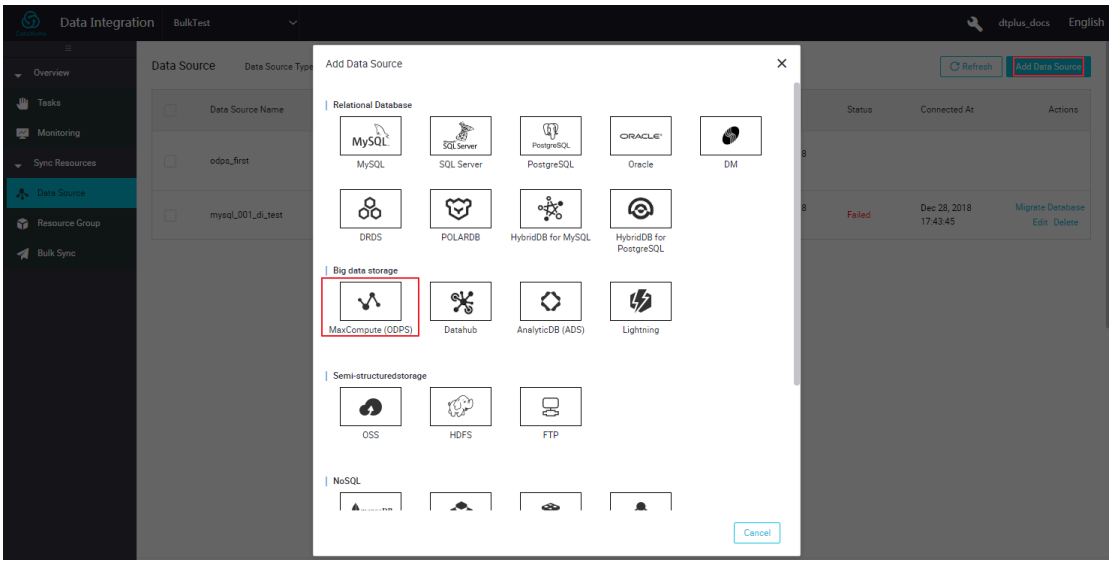
D. Verify the connectivity: After the connection is successfully established, the status is changed to Available. If the status is Unavailable, you must log on to the ECS instance, and run the `tail - f / home / admin / alisataskn ode / logs / heartbeat . log` command to check

whether the heartbeat message between DataWorks and the ECS instance is timed out.

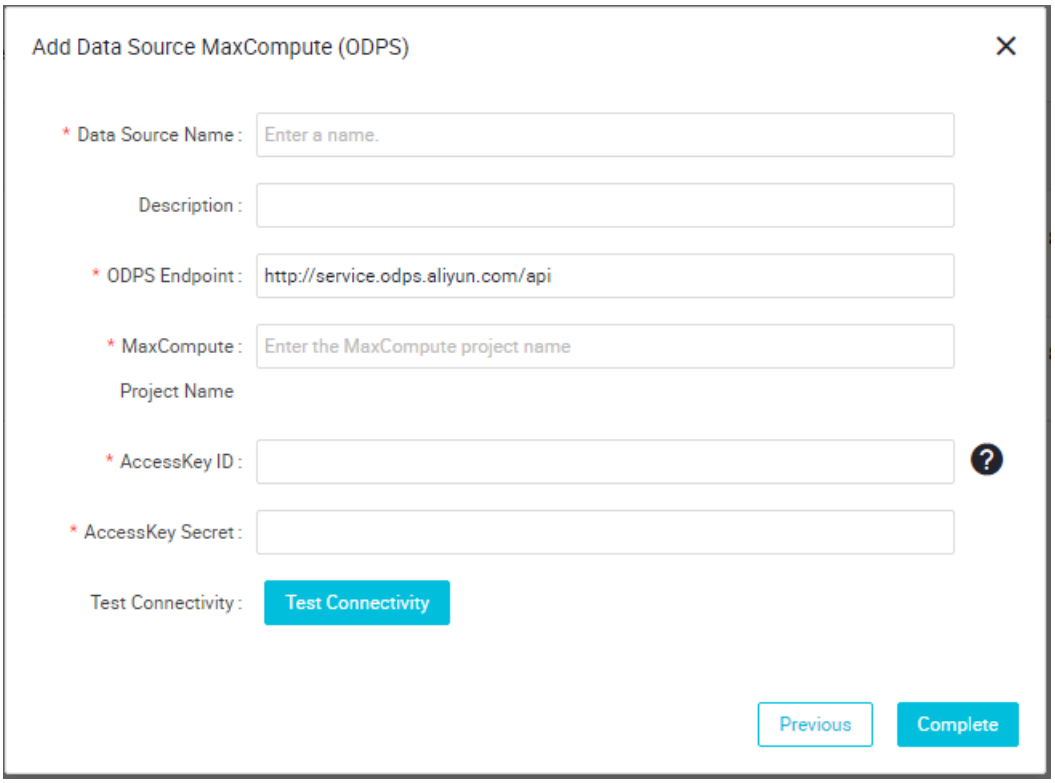
d. Add a data source.

A. On the Data Integration page, select Data Source in the left-side navigation bar, and click Add Data Source.

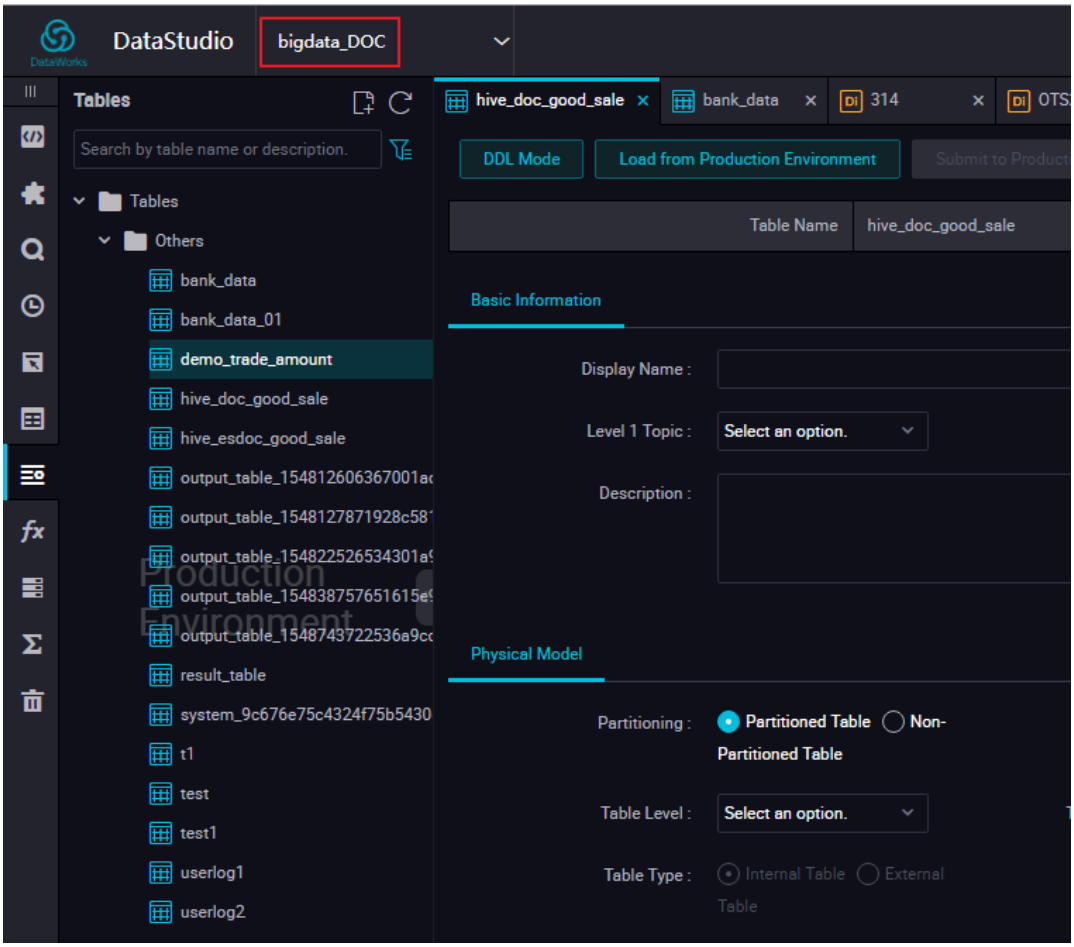
B. Select MaxCompute as the source type.



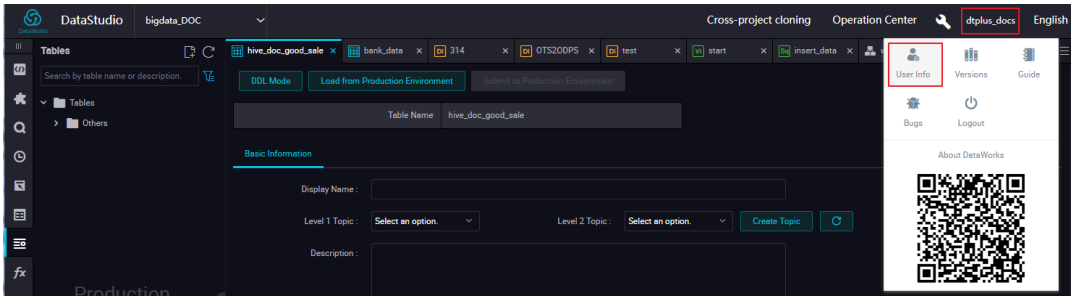
C. Enter information about the data source. This example creates a data source named odps_es, as shown in the following figure:



- ODPS workspace name: On the Data Analytics page of DataWorks, the corresponding workspace name of a table is displayed on the right of the icon in the upper left corner, as shown in the following figure:



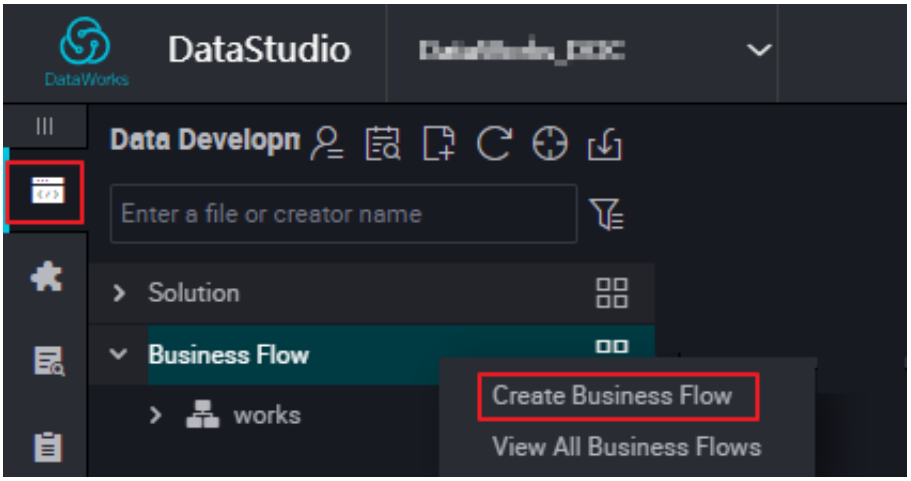
- AccessKeyId/AccessKeySecrete: Move the pointer over your username and select User Info, as shown in the following figure:



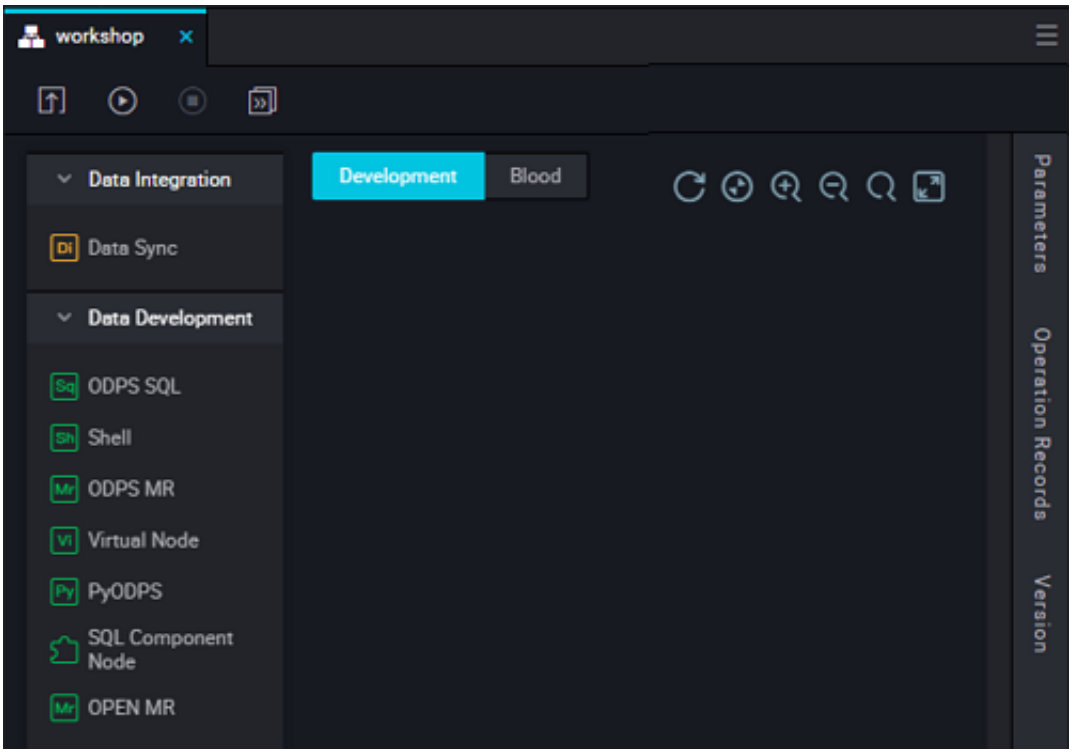
On the Personal Account page, move the pointer over your avatar, and click accesskeys as shown in the following figure:

e. Configure the synchronization task.

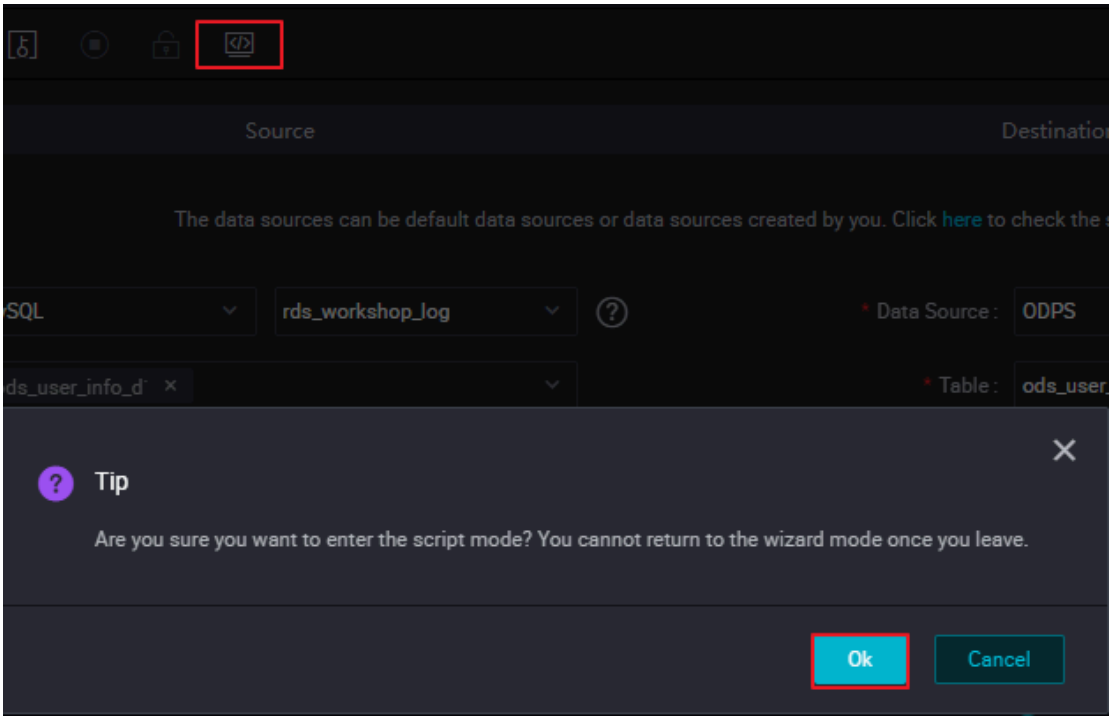
A. On Data Analytics page, click the Data Analytics icon in the left-side navigation pane, and click Business Flow.



B. Click the target business flow, select Data Integration, select Create Data Integration Node > Data Sync, and then enter the synchronization node name.

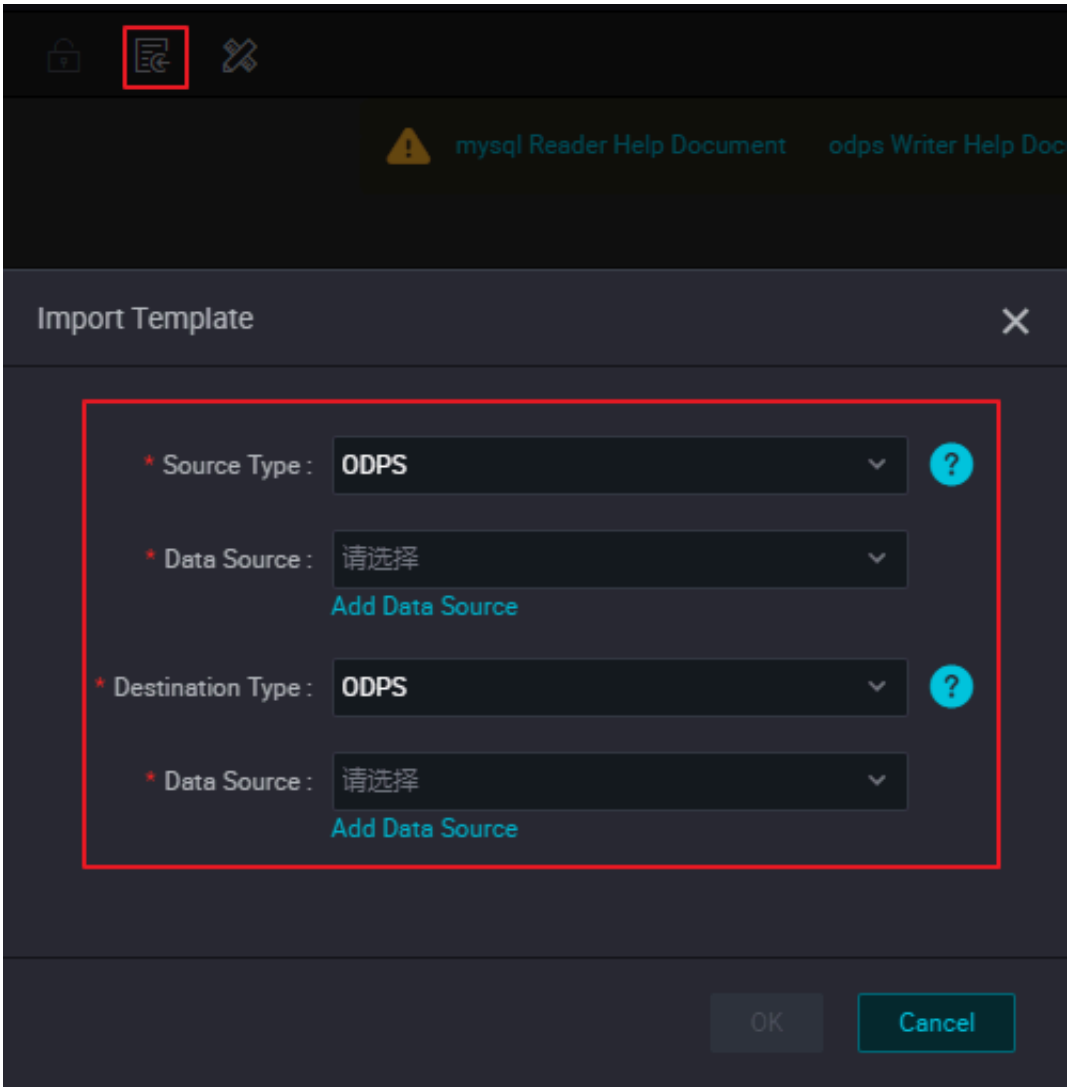


C. After successfully creating the synchronization node, click the Switch to Script Mode icon at the top of the new synchronization node page, and select Confirm.



D. At the top of on the Script Mode page, click the Apply Template icon. Enter the corresponding information for Source Type, Data Source, Destination

source type and data source options, and then click OK to generate an initial script.



E. *Configure the data synchronization script*. For more information about configuration rules of Elasticsearch, see *Configure writer plug-ins*.

Odps Reader 帮

```
"reader": {
  "plugin": "odps",
  "parameter": {
    "partition": "pt=1",
    "datasource": "odps_es",
    "column": [
      "create_time",
      "category",
      "brand",
      "buyer_id",
      "trans_num",
      "trans_amount",
      "click_cnt"
    ],
    "table": "hive_doc_good_sale"
  }
},
"writer": {
  "plugin": "elasticsearch",
  "parameter": {
    "accessId": "elastic",
    "endpoint": "http://es-cn-mp[redacted].elasticsearch.aliyuncs.com:9200",
    "indexType": "elasticsearch",
    "accessKey": "[redacted]",
    "cleanup": true,
    "discovery": false,
    "column": [
      {
        "name": "create_time",
        "type": "string"
      },
      {
        "name": "category",
        "type": "string"
      },
      {
        "name": "brand",
        "type": "string"
      },
      {
        "name": "buyer_id",
        "type": "string"
      },
      {
        "name": "trans_num",
        "type": "long"
      },
      {
        "name": "trans_amount",
        "type": "double"
      },
      {
        "name": "click_cnt",
        "type": "long"
      }
    ],
    "index": "es_index",
    "batchSize": 1000,
```

Note:

- The configuration of the synchronization script contains three parts: Reader, Writer, and Setting. `Reader` is used to configure the source cloud services whose data you want to synchronize. `Write` is used to configure the config file of Alibaba Cloud Elasticsearch. `Setting` is used to configure settings for packet loss and maximum concurrent tasks.

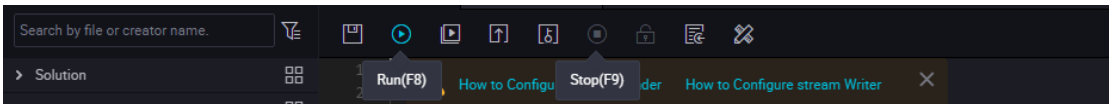
- `Endpoint` specifies the private or public IP address of the Alibaba Cloud Elasticsearch instance. This example uses a private IP address. Therefore, no whitelist is required. If you use an external IP address, you must configure a whitelist that contains public IP addresses that are allowed to access Elasticsearch on the Network and Snapshots page of Alibaba Cloud Elasticsearch. The whitelist must contain the *IP addresses of your DataWorks server* and the resource groups you use.
- You must configure the username and password that are used to log on to the Alibaba Cloud Elasticsearch instance in `accessId` and `accesskey` of Elasticsearch Writer.
- Enter the index name of the Elasticsearch instance in `index` . You need to use this index name to access the data on the Alibaba Cloud Elasticsearch instance. This example uses the `index` named `es_index` .
- If your MaxCompute table is a partitioned table, you must configure the partition information in the `partition` field. The partition information in this example is pt=1.

Sample configuration code:

```
{
" configurat  ion ": {
" reader ": {
" plugin ": " odps ",
" parameter ": {
  " partition ": " pt = 1 ",
  " datasource ": " odps_es ",
  " column ": [
    " create_tim  e ",
    " category ",
    " brand ",
    " buyer_id ",
    " trans_num ",
    " trans_amou  nt ",
    " click_cnt "
  ],
  " table ": " hive_doc_g  ood_sale "
}
},
" writer ": {
" plugin ": " elasticsea  rch ",
" parameter ": {
  " accessId ": " elastic ",
  " endpoint ": " http :// es - cn - mpXXXXXXX . elasticsea
 rch . aliyuncs . com : 9200 ",
  " indexType ": " elasticsea  rch ",
  " accessKey ": " XXXXXX ",
  " cleanup ":  true ,
```

```
  " discovery ":  false ,
  " column ": [
    {
      " name ": " create_tim  e ",
      " type ": " string "
    },
    {
      " name ": " category ",
      " type ": " string "
    },
    {
      " name ": " brand ",
      " type ": " string "
    },
    {
      " name ": " buyer_id ",
      " type ": " string "
    },
    {
      " name ": " trans_num ",
      " type ": " long "
    },
    {
      " name ": " trans_amou  nt ",
      " type ": " double "
    },
    {
      " name ": " click_cnt ",
      " type ": " long "
    }
  ],
  " index ": " es_index ",
  " batchSize ":  1000 ,
  " splitter ": ",",
}
},
" setting ": {
" errorLimit ": {
  " record ": " 0 "
},
" speed ": {
  " throttle ":  false ,
  " concurrent ":  1 ,
  " mbps ": " 1 ",
  " dmu ":  1
}
}
},
" Type ": " job  ",
" version ": " 1 . 0 "
}
```

F. After the script is synchronized, click Run to synchronize ODPS data to Alibaba Cloud Elasticsearch.



3. Verify the result

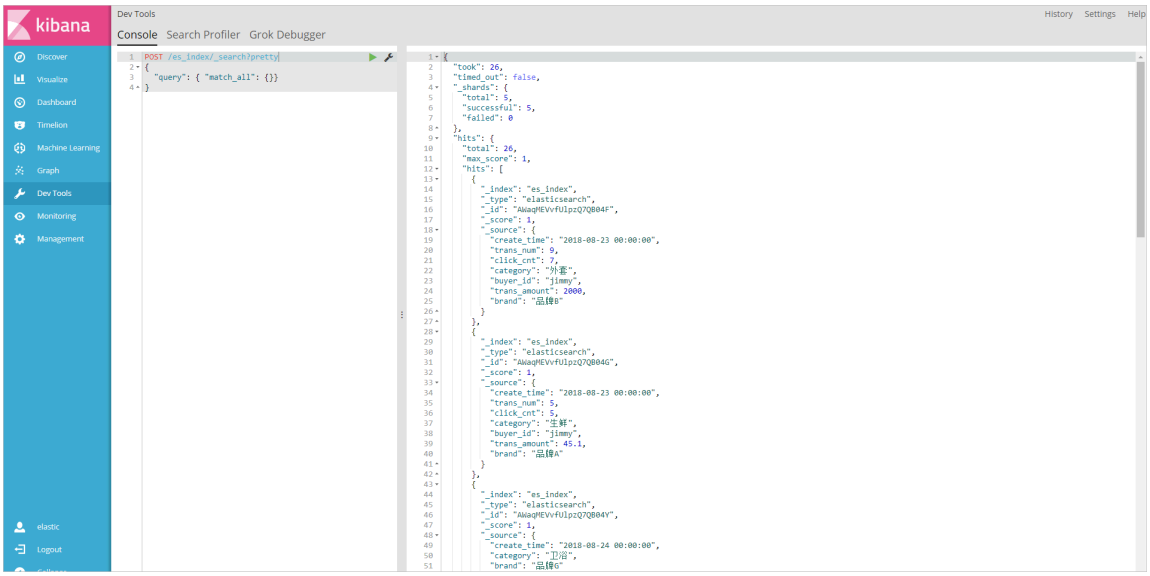
   a. Log on to the Alibaba Cloud Elasticsearch console, click Kibana console in the upper-right corner, and select Dev Tools.

   b. Run the following command to verify that data is successfully replicated to Elasticsearch.

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

`es_index` indicates the value of the `index` field during data synchronization.

If data is successfully synchronized, the following page is displayed:



   c. Run the following command to sort documents based on the `trans_num` field:

```
 POST  / es_index / _search ?  pretty
{
" query ": { " match_all ": {} },
" sort ": { " trans_num ": { " order ": " desc " } }
}
```

   d. Run the following command to search the `category` and `brand` fields in documents:

```
 POST  / es_index / _search ?  pretty
{
" query ": { " match_all ": {} },
" _source ": [" category ", " brand "]
```

```
}
```

e. Run the following command to query documents whose `category` is `fresh` :

```
POST / es_index / _search ? pretty
{
" query ": { " match ": {" category ":" fresh "} }
}
```

For more information, see *Elasticsearch access test* and *Elastic help center*.

FAQ

An error occurs when connecting to the Alibaba Cloud Elasticsearch instance

1. Before you execute the synchronization script, check whether you have selected the resource group that you have created in the preceding step on the right-side configuration tasks resources group menu.
   - If you have selected the resource group, go to the next step.
   - If you have not selected the resource group, click the right-side configuration tasks resources group menu, select the resource group that you have created, and click Run.
2. Check whether the configuration of the synchronization script is correct, including the endpoint, accessId, and accesskey. The endpoint specifies the private or public IP address of your Elasticsearch instance. Configure a whitelist if you use a public IP address. The accessId specifies the username that is used to access the Elasticsearch instance, which is elastic by default. The accesskey specifies the password that is used to access the Elasticsearch instance.

3.6 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.

Versions

Elasticsearch 5.5.3 with X-Pack is supported.

> **Note:**
> Elasticsearch 6.3.2 with X-Pack is not supported.

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:

1. Activate Alibaba Cloud VPC
   a. On the Alibaba Cloud website, choose Products > Networking > Virtual Private Cloud, and then click Activate Now.
   b. Log on to the VPC console, and click Create VPC to create a VPC.
   c. 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)* .
2. Activate Alibaba Cloud Object Storage Service
   a. Log on to the Alibaba Cloud console, choose Products > Storage & CDN > Object Storage Service, and click Buy Now.
   b. 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.
   c. Create a bucket according to the instructions displayed on the page.
3. Activate Alibaba Cloud Elasticsearch
   a. On the Alibaba Cloud website, choose Products > Analytics & Big Data > Elasticsearch, and then the product page is displayed.

   Note:

You can get a 30-day free trial.

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

4. Activate Alibaba Cloud E-MapReduce

a. On the Alibaba Cloud website, choose Products > Analytics & Big Data > E-MapReduce, and then the product page is displayed.

b. Click Buy Now, and complete the configuration.

c. 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.

- 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 HistorySer ver
 18981 WebAppProx yServer
 14023 Jps
 15949 gateway . jar
 16621 ZeppelinSe rver
 1133 EmrAgent
 15119 RunJar
 17519 ResourceMa nager
 1871 Applicatio n
 19316 JobHistory Server
 1077 WatchDog
 17237 SecondaryN ameNode
 16502 NameNode
 16988 ApacheDsTa nukiWrappe r
 18429 Applicatio nHistorySe rver
```

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:

1. Install Maven.

Make sure that your computer has *Maven* installed.

2. Generate an engineering framework.

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

```
mvn   archetype : generate  - DgroupId = com . aliyun . emrtoes
 - DartifactI  d = emrtoes  - Darchetype  ArtifactId = maven -
archetype - quickstart  - Dinteracti  veMode = 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.

3. Add Hadoop and ES-Hadoop dependencies.

   Start this project with any IED, then edit the `pom . xml` file. Add the following content to 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 . 0 . 3 </ version >
  </ dependency >
   < dependency >
       < groupId > org . elasticsea  rch </ groupId >
       < artifactId > elasticsea  rch - hadoop - mr </ artifactId >
       < version > 2 . 5 . 0 </ version >
   </ dependency >
```

4. 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 >
     < configurat  ion >
       < archive >
         < manifest >
           < mainClass > com . aliyun . emrtoes . EmrToES </
 mainClass >
         </ manifest >
       </ archive >
       < descriptor  Refs >
         < descriptor  Ref > jar - with - dependenci  es </
 descriptor  Ref >
       </ descriptor  Refs >
```

```
    </ configurat  ion >
    < 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 > 2 . 1 . 0 </ version >
    < executions >
      < execution >
        < phase > package </ phase >
        < goals >
          < goal > shade </ goal >
        </ goals >
        < configurat  ion >
          < transforme  rs >
            < transforme  r   implementa  tion =" org . apache
. maven . plugins . shade . resource . ApacheLice  nseResourc
eTransform  er ">
            </ transforme  r >
          </ transforme  rs >
        </ configurat  ion >
      </ execution >
    </ executions >
  </ plugin >
</ plugins >
```

5. **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 . Configurat  ion ;
 import   org . apache . hadoop . fs . Path ;
 import   org . apache . hadoop . io . NullWritab  le ;
 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 .
FileInputF  ormat ;
 import   org . apache . hadoop . mapreduce . lib . input .
TextInputF  ormat ;
 import   org . apache . hadoop . util . GenericOpt  ionsParser ;
 import   org . elasticsea  rch . hadoop . mr . EsOutputFo  rmat ;
 import   java . io . IOExceptio  n ;
 public   class   EmrToES  {
     public   static   class   MyMapper   extends   Mapper < Object
,  Text ,  NullWritab  le ,  Text > {
         private   Text   line  =  new   Text ();
        @ Override
         protected   void   map ( Object   key ,  Text   value ,
Context   context )
                throws   IOExceptio  n ,  Interrupte  dException
{
```

```
              if  ( value . getLength () >  0 ) {
                  line . set ( value );
                  context . write ( NullWritab  le . get (),  line
);
            }
        }
    }
     public   static   void   main ( String []  args )  throws
  IOExceptio  n ,  ClassNotFo  undExcepti  on ,  Interrupte
dException  {
         Configurat  ion   conf  =  new   Configurat  ion ();
         String []  otherArgs  =  new   GenericOpt  ionsParser (
conf ,  args ). getRemaini  ngArgs ();
        // Alibaba   Cloud   Elasticsea  rch   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 . speculativ  e
. execution ",  false );
         conf . setBoolean (" mapred . reduce . tasks . speculativ
e . execution ",  false );
         conf . set (" es . nodes ", " The   private   address   of
  your   Elasticsea  rch   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 . getInstanc  e ( conf , " EmrToES ");
         job . setJarByCl  ass ( EmrToES . class );
         job . setMapperC  lass ( EsMapper . class );
         job . setInputFo  rmatClass ( TextInputF  ormat . class );
         job . setOutputF  ormatClass ( EsOutputFo  rmat . class );
         job . setMapOutp  utKeyClass ( NullWritab  le . class );
         job . setMapOutp  utValueCla  ss ( Text . class );
         FileInputF  ormat . setInputPa  ths ( job ,  new   Path (
otherArgs [ 0 ]));
         System . exit ( job . waitForCom  pletion ( true ) ?  0
:  1 );
    }
```

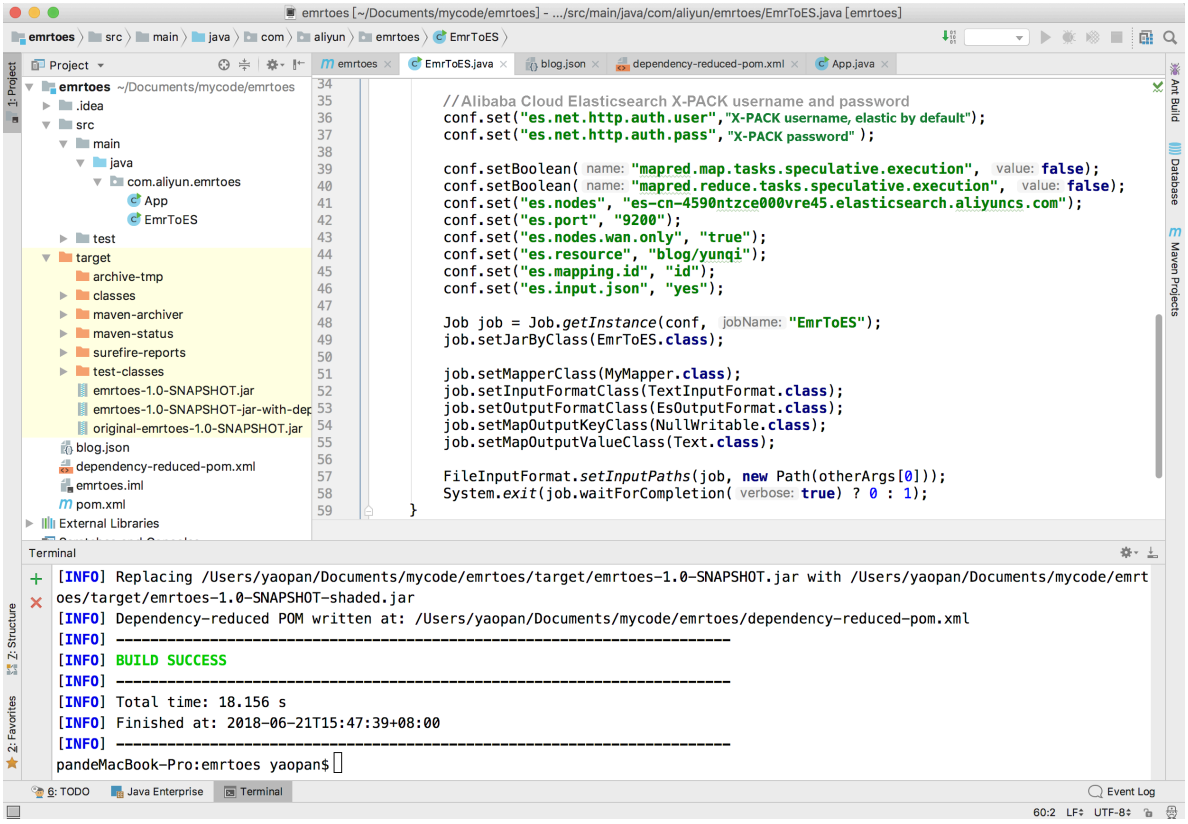
```
}
```

6. 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

1. Test the data

   a. Write the following data to blog.json:

```
{" id ":" 1 "," title ":" git introducti on "," posttime ":"
 2016 - 06 - 11 "," content ":" The main difference between
   svn and git ..."}
{" id ":" 2 "," title ":" Introducti on 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 ?..."}
```

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

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

c. **Upload blog.json to HDFS:**

```
hadoop   fs  - mkdir  / work
```

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

2. 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 -
dependenci es . jar root @ YourIP :/ root
```

3. Execute the MR job

   Run the following command:

```
hadoop jar emrtoes - 1 . 0 - SNAPSHOT - jar - with -
dependenci es . 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/had
oop/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. Instea
d, 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/app
lication_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
```

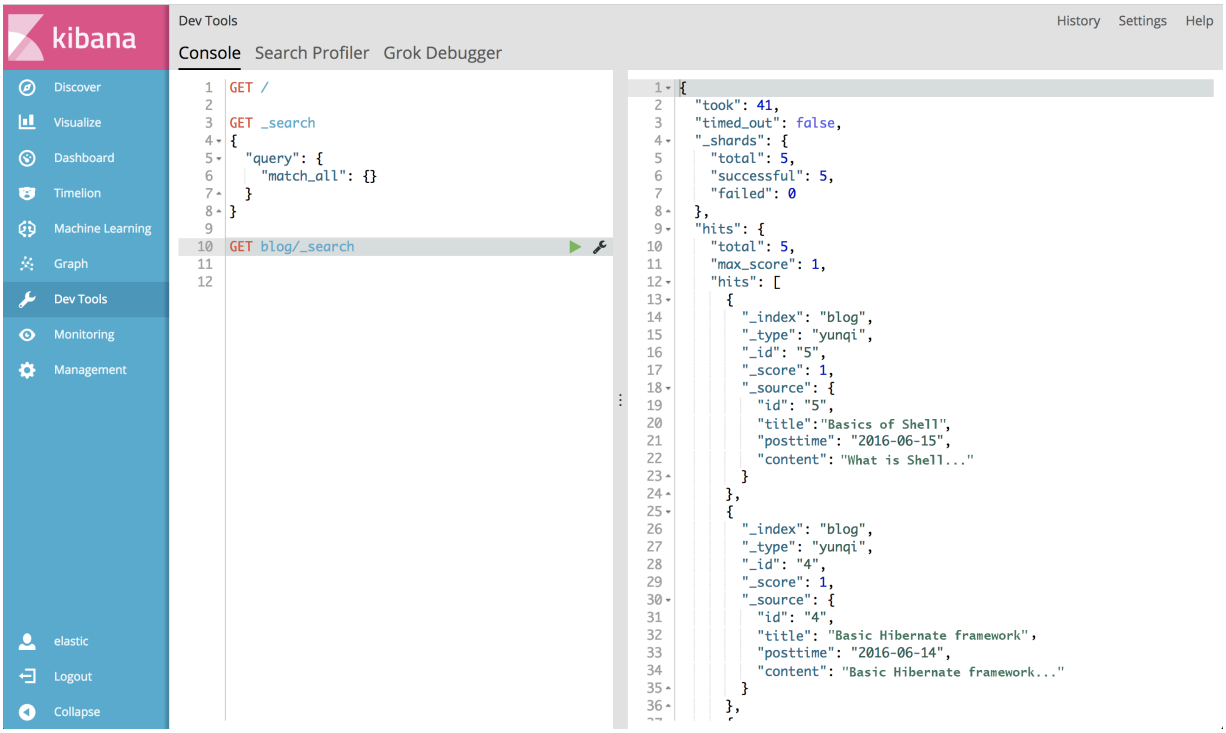
Verify results

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
```

```
1. root@emr-header-1:~ (ssh)
[root@emr-header-1 ~]# curl -u elastic -XGET es-cn-4590nukw4000xuig3.elasticsearch.aliyuncs.com:9200/blog
/_search?pretty
Enter host password for user 'elastic':
{
  "took" : 17,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "failed" : 0
  },
  "hits" : {
    "total" : 5,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "blog",
        "_type" : "yunqi",
        "_id" : "5",
        "_score" : 1.0,
        "_source" : {
          "id" : "5",
          "title" : "Basics of Shell",
          "posttime" : "2016-06-15",
          "content" : "What is Shell..."
        }
      },
      {
        "_index" : "blog",
        "_type" : "yunqi",
        "_id" : "4",
        "_score" : 1.0,
        "_source" : {
          "id" : "4",
          "title" : "Basic Hibernate framework",
          "posttime" : "2016-06-14",
          "content" : "Basic Hibernate framework..."
        }
      },
```

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 reducers.
- Conf.setBoolean(“mapred.reduce.tasks.speculative.exe cution ", false)

  This parameter disables speculative execution for the mappers.

- 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.

3.7 Logstash deployment

Prepare the environment

1. 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 *Classic network errors*.

2. Download Logstash v5.5.3.

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

3. Decompress the downloaded Logstash package.

```
tar - xzvf logstash - 5 . 5 . 3 . tar . gz
# A stringent configurat ion file checking feature is
  added to Elasticsea rch later than version 5 . x .
```

Test cases

1. Create the user name and password for data access.

   - Creates a role.

```
curl - XPOST - H " Content - Type : applicatio n / json
" - u elastic : es - password http ://*** instanceId ***.
elasticsea rch . aliyuncs . com : 9200 / _xpack / security /
role /*** role - name *** - d '{" cluster ": [" manage_ind
ex_templat es ", " monitor "]," indices ": [{" names ":
[ " logstash -*" ], " privileges ":[" write "," delete ","
create_ind ex "]}]}'
# 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 permission s on the Logstash -*
index must be assigned when you add a user role
.
```

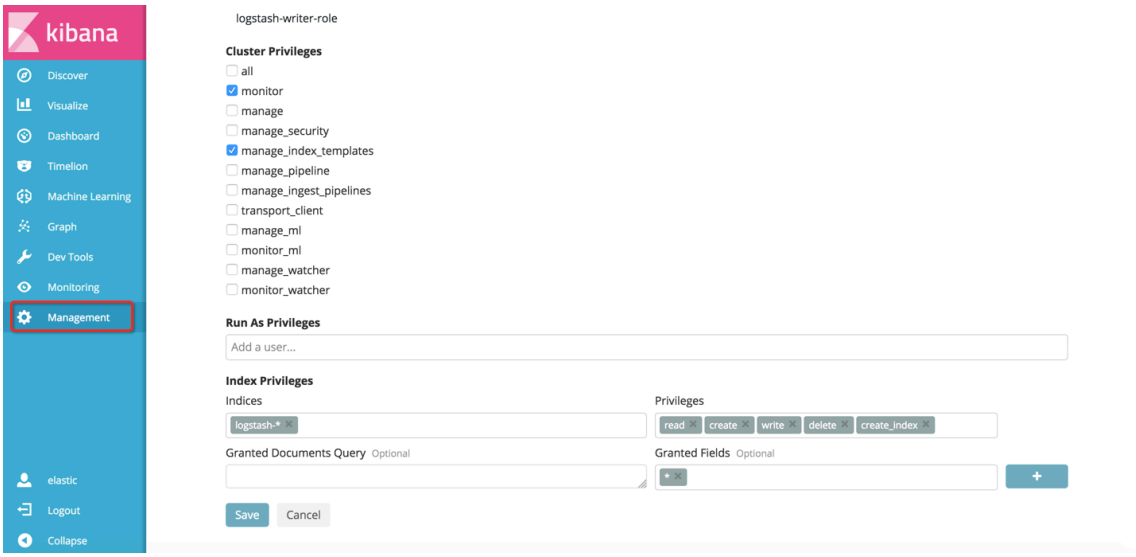
   - Create a user

```
curl - XPOST - H " Content - Type : applicatio n / json
" - u elastic : es - password http ://*** instanceId ***.
elasticsea rch . 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 .
```

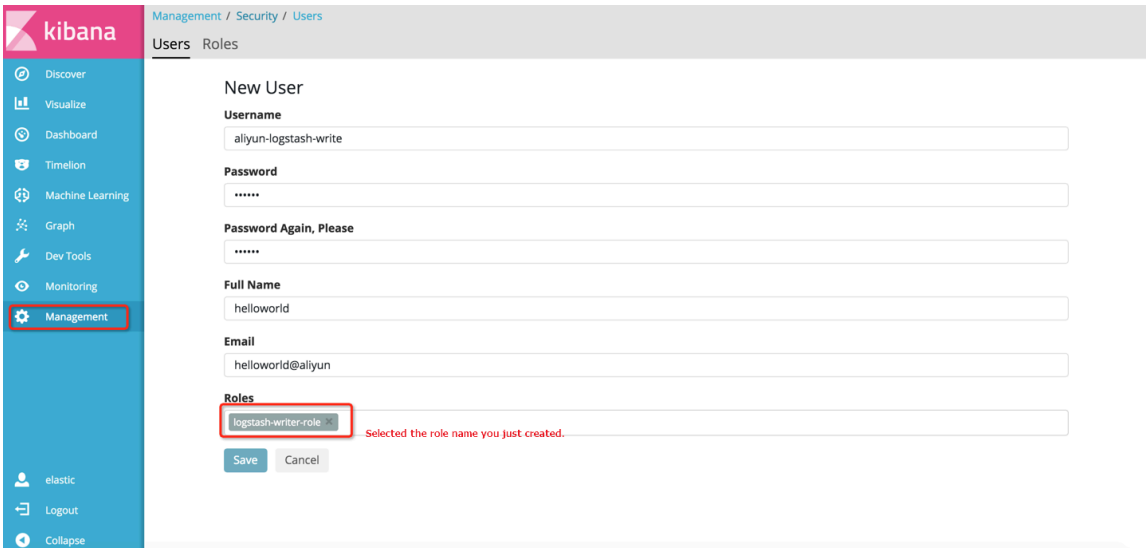
Note:

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

- Add a role



- Add a user



2. 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  {
   elasticsea  rch  {
     hosts  => [" http ://*** instanceId ***. elasticsea  rch .
 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 .
```

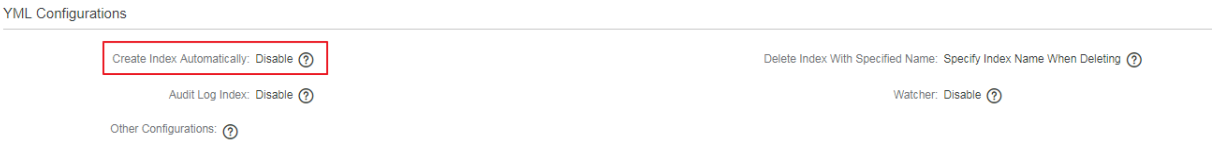
Run

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  configurat  ions  are  required  for
   data  transfer .
 This  example  shows  how  to  obtain  file  changes
 through  Logstash  and  submit  the  changed  data  to  the
   Elasticsea  rch  cluster .  All  the  new  contents  in
 the  monitored  file  can  be  automatica  lly  indexed  to
   the  Elasticsea  rch  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

```
[2017-12-01T15:01:11,523][INFO ][logstash.outputs.elasticsearch] Retrying individual bulk actions that failed or were rejected by the previous bulk request. {:count=>1}
[2017-12-01T15:01:13,534][INFO ][logstash.outputs.elasticsearch] retrying failed action with response code: 403 ({"type"=>"security_exception", "reason"=>"action [indices:admin/create] is unauthorized for
user [logstash-writer-user]"})
[2017-12-01T15:01:13,534][INFO ][logstash.outputs.elasticsearch] Retrying individual bulk actions that failed or were rejected by the previous bulk request. {:count=>1}
[2017-12-01T15:01:17,549][INFO ][logstash.outputs.elasticsearch] retrying failed action with response code: 403 ({"type"=>"security_exception", "reason"=>"action [indices:admin/create] is unauthorized for
user [logstash-writer-user]"})
[2017-12-01T15:01:17,549][INFO ][logstash.outputs.elasticsearch] Retrying individual bulk actions that failed or were rejected by the previous bulk request. {:count=>1}
[2017-12-01T15:01:25,567][INFO ][logstash.outputs.elasticsearch] retrying failed action with response code: 403 ({"type"=>"security_exception", "reason"=>"action [indices:admin/create] is unauthorized for
user [logstash-writer-user]"})
[2017-12-01T15:01:25,567][INFO ][logstash.outputs.elasticsearch] Retrying individual bulk actions that failed or were rejected by the previous bulk request. {:count=>1}
[2017-12-01T15:01:41,592][INFO ][logstash.outputs.elasticsearch] retrying failed action with response code: 403 ({"type"=>"security_exception", "reason"=>"action [indices:admin/create] is unauthorized for
user [logstash-writer-user]"})
```

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

Insufficient memory

```
Java HotSpot(TM) 64-Bit Server VM warning: INFO: os::commit_memory(0x00000000c5330000, 986513408, 0) failed; error='Cannot allocate memory' (errno=12)
#
# There is insufficient memory for the Java Runtime Environment to continue.
# Native memory allocation (mmap) failed to map 986513408 bytes for committing reserved memory.
# An error report file with more information is saved as:
```

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

```
[root@iZbp1droc0yv90nh95ae6zZ logstash-5.5.3]# bin/logstash -f task/test.conf
ERROR StatusLogger No log4j2 configuration file found. Using default configuration: logging only errors to the console.
Sending Logstash's logs to /root/tmp/logstash-5.5.3/logs which is now configured via log4j2.properties
[2017-12-01T15:18:02,034][ERROR][logstash.agent           ] Cannot create pipeline {:reason=>"Expected one of #, {, } at line 12, column 22 (byte 261) after output {\n  elasticsearch {\n    hosts => [\"ht
tp://[redacted].elasticsearch.aliyuncs.com:9200\"]\n    user => \"logstash[redacted]\"\n    password => Ac[redacted]"}
[2017-12-01T15:18:02,465][INFO ][logstash.outputs.elasticsearch] Elasticsearch pool URLs updated {:changes=>{:removed=>[], :added=>[http://logstash_system_monitor:xxxxxx@es-cn-mp90cbsyl002ejbtn.elasticsea
rch.aliyuncs.com:9200/]}}
[2017-12-01T15:18:02,475][INFO ][logstash.outputs.elasticsearch] Running health check to see if an Elasticsearch connection is working {:healthcheck_url=>http://logstash_system_monitor:xxxxxx@es-cn-mp90cb
```

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.

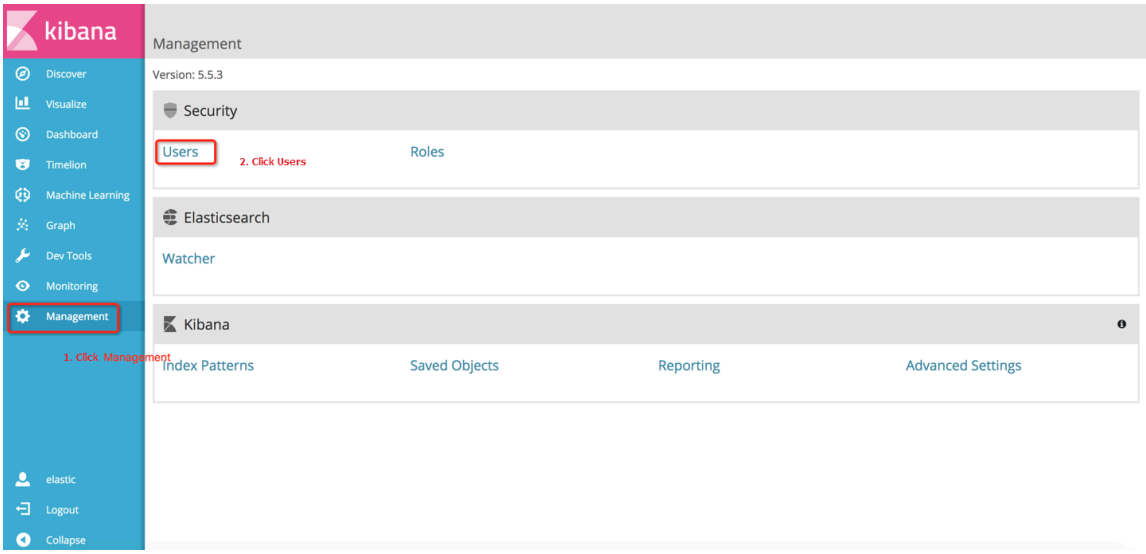
Additional instructions

To monitor the Logstash node and collect logs:

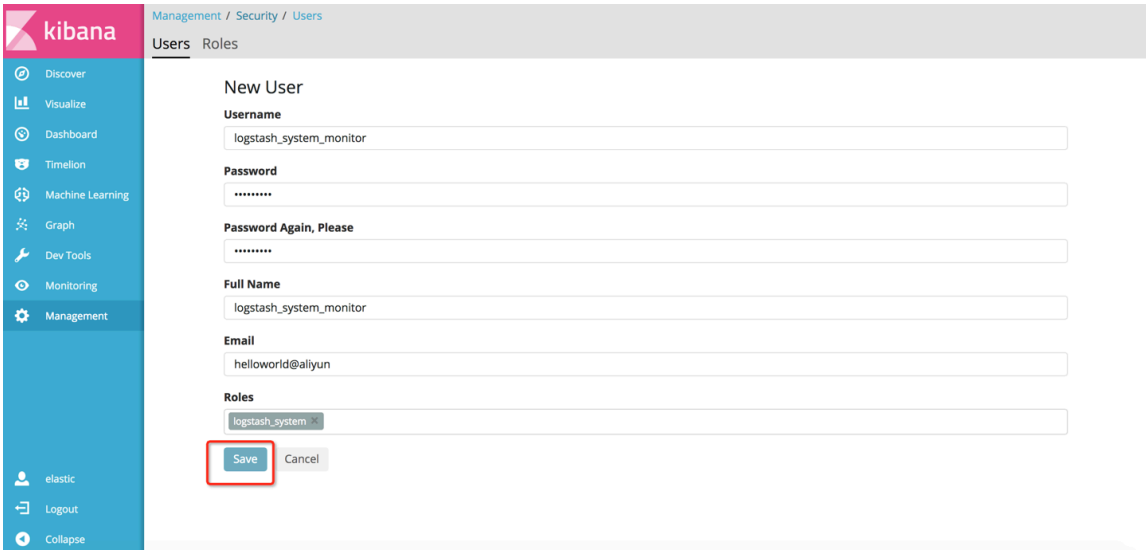
- Install the X-Pack plugin for Logstash. For more information, see *download link*.
- Deploy the X-Pack after download.
- ```
  bin/logstash-plugin install file:///path/to/file/x-pack-5.5.3.zip
  ```
- Add a Logstash monitor user. Alibaba Cloud Elasticsearch cluster disables the logstash_system user 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.

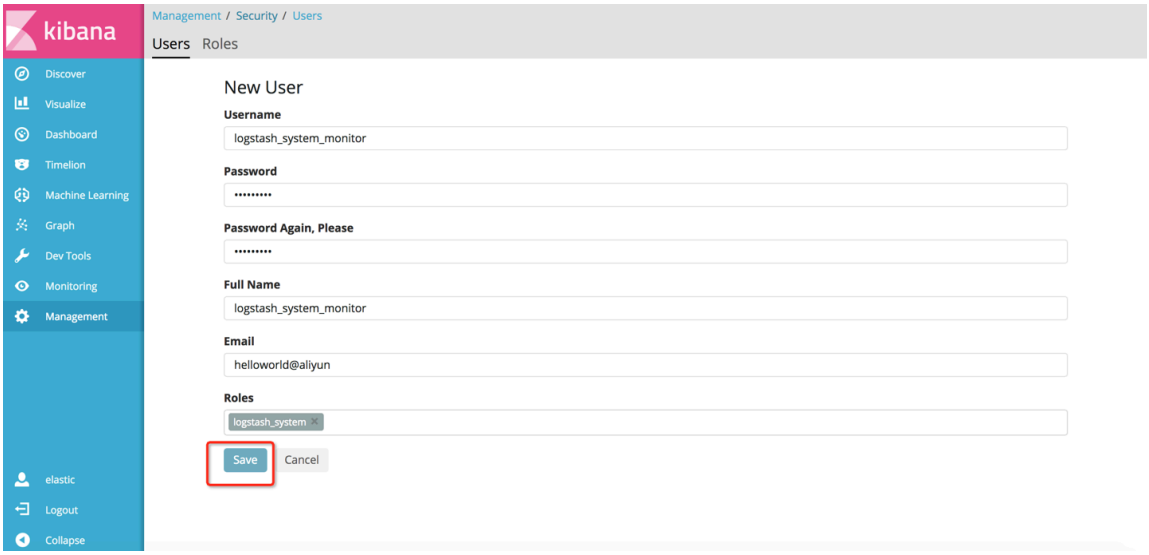
1. Log on to the Kibana management page, and perform the operations according to the following figure:



2. Click the Create User button.



3. Enter the required information. Save and submit the information.



- Add a user through commands

```
 curl  - u   elastic : es - password  - XPOST   http ://***
 instanceId ***. elasticsea  rch . aliyuncs . com : 9200 / _xpack
 / security / user / logstash_s  ystem_moni  tor  - d  '{" password
 " : "*** logstash - monitor - password ***"," roles " : ["
 logstash_s  ystem "]," 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_s  ystem_moni  tor .
```

3.8 Migrate ECS-hosted ES instances

Prerequisites

This document explains how to migrate data from an ECS-hosted Elasticsearch instance to an Alibaba Cloud Elasticsearch instance. You must meet the following requirements before migrating data. If you do not meet the following requirements, see *Logstash deployment* to migrate data through other migration solutions.

- 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 (`indiceCrea te . 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 : applicatio n / json "` parameter to the command and run the command again.

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

```
// Obtain all the indexes on the source cluster . If you do not have the required permission s , remove the "- u user : pass " parameter . Make sure that you have replaced oldCluster Host with the name of the ECS instance that hosts the source cluster .
curl - u user : pass - XGET http :// oldCluster Host / _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 :// oldCluster Host /
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 synchroniz ation process , and
change the number to one after the migration has
completed .
 // ewClusterH ost 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 ://< newCluster Host >/<
testindex > - d '{
   " testindex " : {
       " settings " : {
           " number_of_ shards " : " 5 ", // Set the number
  of shards for the correspond ing 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_tim e " : {
                     " 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_in terval " : "- 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_in terval " : " 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 `YML` 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.

- Migrate small amounts of data

  Run the `reindex . sh` script.

```
#! / bin / bash
# file : reindex . sh
 indexName =" The name of the index "
 newCluster User =" The username that is used to log
 on to the target cluster "
 Newcluster pass = " The password that is used to log
   on to the target cluster "
 newCluster Host =" The ECS instance that hosts the
 target cluster "
 Oldcluster user = " The username that is used to log
   on to the source cluster "
 Oldcluster pass = " 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 .
 Oldcluster host = " The ECS instance that hosts the
 source cluster "
```

```
curl - u ${ newCluster User }:${ newCluster Pass } - XPOST "
http://${ newCluster Host }/ _reindex ? pretty " - H " Content
- Type : applicatio n / json " - d '{
   " source ": {
       " remote ": {
           " host ": "'${ oldCluster Host }'",
           " username ": "'${ oldCluster User }'",
           " password ": "'${ oldCluster Pass }'"
       },
       " index ": "'${ indexName }'",
       " query ": {
           " match_all ": {}
       }
   },
   " dest ": {
      " index ": "'${ indexName }'"
   }
}'
```

- **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 : circleRein dex . sh
# CONTROLLIN G STARTUP :
# This script is used to remotely rebuild the index
  using the reindex operation . Requiremen ts :
# 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 circleRein dex . 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 .
 For example :
         sh circleRein dex . sh 1
         sh circleRein dex . sh 5
         sh circleRein dex . sh - 1 "
 indexName =" The name of the index "
 newCluster User =" The username that is used to log
 on to the target cluster "
 newCluster Pass =" The password that is used to log
 on to the target cluster "
 oldCluster User =" The username that is used to log
 on to the source cluster "
 oldCluster Pass =" The password that is used to log
 on to the source cluster "
```

```
##  http :// myescluste  r . com
 newCluster  Host =" 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
 oldCluster  Host =" The   ECS   instance   that   hosts   the
 source   cluster "
 timeField =" The   field   that   specifies   the   time   window
 during   which   the   incrementa  l   data   is   migrated "
 reindexTim  es = 0
 lastTimest  amp = 0
 curTimesta  mp =` date  +% s `
 hasError = false
 function   reIndexOP () {
     reindexTim  es =$[${ reindexTim  es } +  1 ]
     curTimesta  mp =` date  +% s `
     ret =` curl  - u  ${ newCluster  User }:${ newCluster  Pass } -
 XPOST  "${ newCluster  Host }/ _reindex ?  pretty " - H  " Content
 - Type :  applicatio  n / json " - d  '{
        " source ": {
            " remote ": {
                " host ": "'${ oldCluster  Host }'",
                " username ": "'${ oldCluster  User }'",
                " password ": "'${ oldCluster  Pass }'"
            },
            " index ": "'${ indexName }'",
            " query ": {
                " range " : {
                    "'${ timeField }'" : {
                        " gte " : '${ lastTimest  amp }',
                        " lt " : '${ curTimesta  mp }'
                    }
                }
            }
        },
        " dest ": {
            " index ": "'${ indexName }'"
        }
    }'`
     lastTimest  amp =${ curTimesta  mp }
     echo  "${ reindexTim  es }  reindex   operations   have   been
   performed .  The   last   reindex   operation   is   completed
 at  ${ lastTimest  amp }  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  ${ reindexTim  es }  reindex
 operations "
```

- **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 "
 newCluster  User =" The   username   that   is   used   to   log
 on   to   the   target   cluster "
 Newcluster  pass  = " The   password   that   is   used   to   log
   on   to   the   target   cluster "
 newCluster  Host =" The   ECS   instance   that   hosts   the
 target   cluster "
 Oldcluster  user  = " The   username   that   is   used   to   log
   on   to   the   source   cluster "
 Oldcluster  pass  = " 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 .
 oldCluster  Host =" The   ECS   instance   that   hosts   the
 source   cluster "
 timeField =" updatetime "
 curl  - u  ${ newCluster  User }:${ newCluster  Pass } - XPOST  "
 http ://${ newCluster  Host }/ _reindex ?  pretty " - H  " Content
 - Type :  applicatio  n / json " - d  '{
    " source ": {
        " remote ": {
            " host ": "'${ oldCluster  Host }'",
            " username ": "'${ oldCluster  User }'",
            " password ": "'${ oldCluster  Pass }'"
        },
        " index ": "'${ indexName }'",
        " query ": {
            " bool ": {
                " must_not ": {
                    " exists ": {
                        " field ": "'${ timeField }'"
```

```
                    }
                }
            }
        }
    },
    " dest ": {
        " index ": "'${ indexName }'"
    }
}'
```

- **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 / env   python
# -*-  coding :  UTF - 8  -*-
#  File   name :  indiceCrea  te . py
 import   sys
 import   base64
 import   time
 Import   httplib
 import   json
##  The   ECS   instance   that   hosts   the   source   cluster  ( ip
 + port )
 oldCluster  Host  = " old - cluster . com "
#  The   username   that   is   used   to   log   on   to   the
 source   cluster .  The   username   field   can   be   left   empty
 oldCluster  UserName  = " old - username "
##  The   password   that   is   used   to   log   on   to   the
 source   cluster .  The   password   field   can   be   left   empty
 oldCluster  Password  = " old - password "
##  The   ECS   instance   that   hosts   the   target   cluster  ( ip
 + port )
 newCluster  Host  = " new - cluster . com "
##  The   username   that   is   used   to   log   on   to   the
 target   cluster .  The   username   field   can   be   left   empty
 newCluster  User  = " new - username "
##  The   password   that   is   used   to   log   on   to   the
 target   cluster .  The   password   field   can   be   left   empty
 newCluster  Password  = " new - password "
 DEFAULT_RE  PLICAS  =  0
 def   httpReques  t ( method ,  host ,  endpoint ,  params ="",
 username ="",  password =""):
     conn  =  httplib . HTTPConnec  tion ( host )
     headers  = {}
     if  ( username  ! = "") :
        ' Hello  { name },  your   age   is  { age } !'. format ( name
  = ' Tom ',  age  = ' 20 ')
         base64stri  ng  =  base64 . encodestri  ng ('{ username }:{
 password }'. format ( username  =  username ,  password  =  password
 )). replace ('\ n ', '')
         headers [" Authorizat  ion "] = " Basic  % s " %  base64stri
 ng ;
     if  " GET " ==  method :
         Content - Type :  applicatio  n / x - www - form -
 urlencoded
```

```
        conn . request ( method = method ,  url = endpoint ,  headers
= headers )
    else  :
        Headers  [" Content - Type "] = " applicatio  n / JSON "
        conn . request ( method = method ,  url = endpoint ,  body =
params ,  headers = headers )
    response  =  conn . getrespons  e ()
    res  =  response . read ()
    return   res
def   httpGet ( host ,  endpoint ,  username ="",  password =""):
    return   httpReques  t (" GET ",  host ,  endpoint , "",
username ,  password )
def   httpPost ( host ,  endpoint ,  params ,  username ="",
password =""):
    return   httpReques  t (" POST ",  host ,  endpoint ,  params ,
username ,  password )
def   httpPut ( host ,  endpoint ,  params ,  username ="",  password
=""):
    return   httpReques  t (" PUT ",  host ,  endpoint ,  params ,
username ,  password )
def   getIndices ( host ,  username ="",  password =""):
    endpoint  = "/ _cat / indices "
    indicesRes  ult  =  httpGet ( oldCluster  Host ,  endpoint ,
oldCluster  UserName ,  oldCluster  Password )
    indicesLis  t  =  indicesRes  ult . split ("\ n ")
    indexList  = []
    for   indices   in   indicesLis  t :
        if  ( indices . find (" open ") >  0 ):
            indexList . append ( indices . split ()[ 2 ])
    return   indexList
def   getSetting  s ( index ,  host ,  username ="",  password =""):
    endpoint  = "/" +  index  + "/ _settings "
    indexSetti  ngs  =  httpGet ( host ,  endpoint ,  username ,
password )
    print   index  + "   The   original   settings : \ n " +
indexSetti  ngs
    settingsDi  ct  =  json . loads ( indexSetti  ngs )
   ##  The   number   of   shards   equals   the   number   of
indexes   on   the   source   cluster   by   default
    number_of_  shards  =  settingsDi  ct [ index ][" settings "]["
index "][" number_of_  shards "]
   ##  The   default   number   of   replicas   is   0
    number_of_  replicas  =  DEFAULT_RE  PLICAS
    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 "
    indexMappi  ng  =  httpGet ( host ,  endpoint ,  username ,
password )
    print   index  + " The   original   mappings : \ n " +
indexMappi  ng
    mappingDic  t  =  json . loads ( indexMappi  ng )
    mappings  =  json . dumps ( mappingDic  t [ index ][" mappings
"])
    newMapping  = "\" mappings \" : " +  mappings
    return   newMapping
def   createInde  xStatement ( oldIndexNa  me ):
    settingStr  =  getSetting  s ( oldIndexNa  me ,  oldCluster  Host
,  oldCluster  UserName ,  oldCluster  Password )
    mappingStr  =  getMapping ( oldIndexNa  me ,  oldCluster  Host ,
oldCluster  UserName ,  oldCluster  Password )
```

```
    createstat ement = "{\ n " + str ( settingStr ) + ",\ n " +
 str ( mappingStr ) + "\ n }"
    return createstat ement
 def  createInde x ( oldIndexNa me , newIndexNa me =""):
    if ( newIndexNa me == "") :
        newIndexNa me = oldIndexNa me
    createstat ement = createInde xStatement ( oldIndexNa me )
    print " new index " + newIndexNa me + " settings and
 mappings : \ n " + createstat ement
    endpoint = "/" + newIndexNa me
    createResu lt = httpPut ( newCluster Host , endpoint ,
 createstat ement , newCluster User , newCluster Password )
    print " new index " + newIndexNa me + " creation result
 :" + createResu lt
## main
 indexList = getIndices ( oldCluster Host , oldCluster UserName
 , oldCluster Password )
 systemInde x = []
 for index in indexList :
    if ( index . startswith (".")):
        systemInde x . append ( index )
    else :
        createInde x ( index , index )
 if ( len ( systemInde x ) > 0 ) :
    for index in systemInde x :
        print index + " It may be a system index that
  will not be recreated . Create the index based on
 your needs ."
```