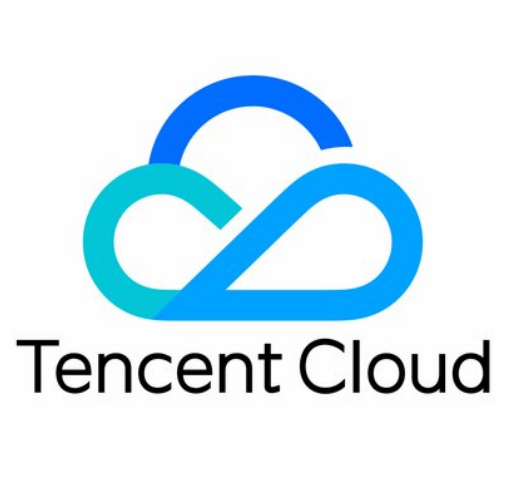
Elasticsearch Service

Tutorials

Product Documentation



©2013-2019 Tencent Cloud. All rights reserved.

Copyright Notice

©2013-2019 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice

Tencent Cloud

All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

©2013-2019 Tencent Cloud. All rights reserved.

Tutorials
Syncing MySQL Data to ES in Real Time

Last updated : 2019-11-08 17:54:15

This document describes how to sync data in MySQL to ES in real time by syncing the MySQL binlog, which can implement low-latency data search and analysis. This method has been proven feasible and is illustrated for your reference.

MySQL binlog

MySQL's binlog is mainly used for master-slave replication and data recovery in databases. The binlog records the CRUD operations performed on the data. During the master-slave replication process, the master database syncs its binlog to the slave database which replays the events stored in the binlog to achieve master-slave sync.

MySQL binlog has three logging modes, namely:

- ROW: It records the changes to each data row, which, however, generates a high number of logs.
- STATEMENT: It records each executed SQL statement that modifies the data, which reduces the number of logs; however, master-slave inconsistency often occurs when SQL statements use functions or triggers.
- MIXED: It combines the advantages of ROW and STATEMENT by choosing either ROW or STATEMENT to generate logs according to the specific SQL statement that performs data operations.

To sync data to an ES cluster with MySQL binlog, you can only use ROW mode, because only this mode knows what is modified in the MySQL data. The following shows the binlog contents in ROW and STATEMENT modes with an UPDATE operation as an example.

- The binlog contents in ROW mode are as follows:

```
SET TIMESTAMP=1527917394/*!*/;
BEGIN
/*!*/;
# at 3751
#180602 13:29:54 server id 1 end_log_pos 3819 CRC32 0x8dabdf01 Table_map: `webservice`.`buildi
ng` mapped to number 74
# at 3819
#180602 13:29:54 server id 1 end_log_pos 3949 CRC32 0x59a8ed85 Update_rows: table id 74 flags:
STMT_END_F
```

©2013-2019 Tencent Cloud. All rights reserved.

```
BINLOG '
UisSWxMBAAAARAAAAOsOAAAAAEoAAAAAAAEACndlYnNlcnZpY2UACGJ1aWxkaW5nAAYIDwEPEREG
wACAAQAAAAHfq40=
UisSWx8BAAAAggAAAG0PAAAAAEoAAAAAAAEAAgAG///A1gcAAAAAAAALYnVpbGRpbmctMTAADwB3
UkRNbjNLYlV5d1k3ajVbD64WWw+uFsDWBwAAAAAAAAtidWlsZGluZy0xMAEPAHdSRE1uM0tiVXl3
WTdqNVsPrhZbD64Whe2oWQ==
'/*!*/;
### UPDATE `webservice`.`building`
### WHERE
### @1=2006 /* LONGINT meta=0 nullable=0 is_null=0 */
### @2='building-10' /* VARSTRING(192) meta=192 nullable=0 is_null=0 */
### @3=0 /* TINYINT meta=0 nullable=0 is_null=0 */
### @4='wRDMn3KbUywY7j5' /* VARSTRING(384) meta=384 nullable=0 is_null=0 */
### @5=1527754262 /* TIMESTAMP(0) meta=0 nullable=0 is_null=0 */
### @6=1527754262 /* TIMESTAMP(0) meta=0 nullable=0 is_null=0 */
### SET
### @1=2006 /* LONGINT meta=0 nullable=0 is_null=0 */
### @2='building-10' /* VARSTRING(192) meta=192 nullable=0 is_null=0 */
### @3=1 /* TINYINT meta=0 nullable=0 is_null=0 */
### @4='wRDMn3KbUywY7j5' /* VARSTRING(384) meta=384 nullable=0 is_null=0 */
### @5=1527754262 /* TIMESTAMP(0) meta=0 nullable=0 is_null=0 */
### @6=1527754262 /* TIMESTAMP(0) meta=0 nullable=0 is_null=0 */
# at 3949
#180602 13:29:54 server id 1 end_log_pos 3980 CRC32 0x58226b8f Xid = 182
COMMIT/*!*/;
```

- The binlog contents in STATEMENT mode are as follows:

```
SET TIMESTAMP=1527919329/*!*/;
update building set Status=1 where Id=2000
/*!*/;
# at 688
#180602 14:02:09 server id 1 end_log_pos 719 CRC32 0x4c550a7d Xid = 200
COMMIT/*!*/;
```

As can be seen from the log contents of the UPDATE operation in ROW and STATEMENT modes, the ROW mode completely records the values of all fields in the row to be modified before and after the update, while the STATEMENT mode only records the SQL statement of the UPDATE operation. Therefore, if you need to sync MySQL data to ES in real time, you can only choose the binlog in ROW mode, obtain and parse the data contents in the binlog, and execute the ES document API to sync the data to the ES cluster.

mysqldump tool

©2013-2019 Tencent Cloud. All rights reserved.

`mysqldump` is a tool for exporting full data from a MySQL database. It can be used in the following way:

```
mysqldump -uelastic -p'Elastic_123' --host=172.16.32.5 -F webservice > dump.sql
```

The command above indicates to export all data of `database:webservice` from the remote database `172.16.32.5:3306` and write to the `dump.sql` file. The `-F` parameter indicates to generate a new binlog file after exporting the data to log all subsequent data operations. The contents of the `dump.sql` file are as follows:

```
-- MySQL dump 10.13 Distrib 5.6.40, for Linux (x86_64)
--
-- Host: 172.16.32.5 Database: webservice
-- ------------------------------------------------------
-- Server version 5.5.5-10.1.9-MariaDBV1.0R012D002-20171127-1822

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `building`
--

DROP TABLE IF EXISTS `building`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `building` (
`Id` bigint(20) unsigned NOT NULL AUTO_INCREMENT COMMENT 'ID',
`BuildingId` varchar(64) NOT NULL COMMENT 'Virtual building ID',
`Status` tinyint(4) NOT NULL DEFAULT '0' COMMENT 'Virtual building status. 0: processing; 1: norm
al; -1: stopped; -2: terminating; -3: terminated',
`BuildingName` varchar(128) NOT NULL DEFAULT '' COMMENT 'Virtual building name',
`CreateTime` timestamp NOT NULL DEFAULT '2017-12-03 16:00:00' COMMENT 'Creation time',
`UpdateTime` timestamp NOT NULL DEFAULT '2017-12-03 16:00:00' COMMENT 'Update time',
PRIMARY KEY (`Id`),
UNIQUE KEY `BuildingId` (`BuildingId`)
) ENGINE=InnoDB AUTO_INCREMENT=2010 DEFAULT CHARSET=utf8 COMMENT='Virtual building table';
/*!40101 SET character_set_client = @saved_cs_client */;
```

©2013-2019 Tencent Cloud. All rights reserved.

```
--
-- Dumping data for table `building`
--

LOCK TABLES `building` WRITE;
/*!40000 ALTER TABLE `building` DISABLE KEYS */;
INSERT INTO `building` VALUES (2000,'building-2',0,'6YFcmntKrNBIeTA','2018-05-30 13:28:31','2018-05-30 13:28:31'),(2001,'building-4',0,'4rY8PcVUZB1vtrL','2018-05-30 13:28:34','2018-05-30 13:28:34'),(2002,'building-5',0,'uyjHVUYrg9KeGqi','2018-05-30 13:28:37','2018-05-30 13:28:37'),(2003,'building-7',0,'DNhyEBO4XEkXpgW','2018-05-30 13:28:40','2018-05-30 13:28:40'),(2004,'building-1',0,'TmtYX6ZC0RNB4Re','2018-05-30 13:28:43','2018-05-30 13:28:43'),(2005,'building-6',0,'t8YQcjeXefWpcyU','2018-05-30 13:28:49','2018-05-30 13:28:49'),(2006,'building-10',0,'WozgBc2IchNyKyE','2018-05-30 13:28:55','2018-05-30 13:28:55'),(2007,'building-3',0,'yJk27cmLOVQLHf1','2018-05-30 13:28:58','2018-05-30 13:28:58'),(2008,'building-9',0,'RSbjotAh8tymfxs','2018-05-30 13:29:04','2018-05-30 13:29:04'),(2009,'building-8',0,'IBOMlhaXV6k226m','2018-05-30 13:29:31','2018-05-30 13:29:31');
/*!40000 ALTER TABLE `building` ENABLE KEYS */;
UNLOCK TABLES;

/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2018-06-02 14:23:51
```

As can be seen from the above, the SQL file exported by mysqldump contains SQL statements for creating table, dropping table, and inserting data, but does not contain the statement for creating database.

Using the open-source go-mysql-elasticsearch tool to sync data to ES

`go-mysql-elasticsearch` is an open-source tool for syncing MySQL data to an ES cluster. For more information, see its GitHub page.

The following describes how `go-mysql-elasticsearch` works. When you launch it for the first time, use the `mysqldump` tool to perform a full sync of the source MySQL database first and write the data to ES through the Elasticsearch client; then, implement a MySQL client as the slave, connect it to the

©2013-2019 Tencent Cloud. All rights reserved.

source MySQL database which, as the master, will sync all data updates to the slave through binlog events. By parsing such events, the updated contents of the data can be obtained and written to ES.

In addition, this tool also provides the feature to count operations. When a data addition, deletion, or update is performed, the count of the corresponding operation will be increased by 1. When the program starts, an HTTP service will be launched, and the number of additions, deletions, and updates can be viewed by calling the HTTP API.

Use limits

1. MySQL binlog must use ROW mode (which is enabled in TencentDB for MySQL by default).
2. The MySQL data table to be synced must contain a primary key; otherwise, it will be ignored directly. If the data table has no primary key, the UPDATE and DELETE operations cannot find the corresponding documents in ES, causing a sync failure.
3. Table structure cannot be modified when the program is running.
4. You need to grant the account used to connect to MySQL the RELOAD and REPLICATION permission.

```
GRANT REPLICATION SLAVE ON *.* TO 'elastic'@'172.16.32.44';
GRANT RELOAD ON *.* TO 'elastic'@'172.16.32.44';
```

How to use

1. Install Go 1.10+. You can simply install the latest version of Go and then set the `GOPATH` environment variable.

2. Run the `go get github.com/siddontang/go-mysql-elasticsearch` command.

3. Run the `cd $GOPATH/src/github.com/siddontang/go-mysql-elasticsearch` command.

4. Run the `make` command to start compiling. After the compilation is successful, an executable file named `go-mysql-elasticsearch` will be generated in the `go-mysql-elasticsearch/bin` directory.

5. Run the `vi etc/river.toml` command to modify the configuration file so as to sync the `webservice.building` table in the `172.16.0.101:3306` database to the building index of the `172.16.32.64:9200` ES cluster. (For detailed description of the configuration file, see the project documentation.)

```
# MySQL address, user and password
# user must have replication privilege in MySQL.
my_addr = "172.16.0.101:3306"
```

©2013-2019 Tencent Cloud. All rights reserved.

```
my_user = "bellen"
my_pass = "Elastic_123"
my_charset = "utf8"

# Set true when elasticsearch use https
#es_https = false
# Elasticsearch address
es_addr = "172.16.32.64:9200"
# Elasticsearch user and password, maybe set by shield, nginx, or x-pack
es_user = ""
es_pass = ""

# Path to store data, like master.info, if not set or empty,
# we must use this to support breakpoint resume syncing.
# TODO: support other storage, like etcd.
data_dir = "./var"

# Inner Http status address
stat_addr = "127.0.0.1:12800"

# pseudo server id like a slave
server_id = 1001

# mysql or mariadb
flavor = "mariadb"

# mysqldump execution path
# if not set or empty, ignore mysqldump.
mysqldump = "mysqldump"

# if we have no privilege to use mysqldump with --master-data,
# we must skip it.
#skip_master_data = false

# minimal items to be inserted in one bulk
bulk_size = 128

# force flush the pending requests if we don't have enough items >= bulk_size
flush_bulk_time = "200ms"

# Ignore table without primary key
skip_no_pk_table = false

# MySQL data source
[[source]]
schema = "webservice"
tables = ["building"]
[[rule]]
```

©2013-2019 Tencent Cloud. All rights reserved.

```
schema = "webservice"
table = "building"
index = "building"
type = "buildingtype"
```

6. Create a building index in the ES cluster, because the tool does not use the "auto create index" feature of ES, and an error will be displayed if the index does not exist.

7. Run the `./bin/go-mysql-elasticsearch -config=./etc/river.toml` command.

8. Output the result in the console.

```
2018/06/02 16:13:21 INFO create BinlogSyncer with config {1001 mariadb 172.16.0.101 3306 belle
n utf8 false false <nil> false false 0 0s 0s 0}
2018/06/02 16:13:21 INFO run status http server 127.0.0.1:12800
2018/06/02 16:13:21 INFO skip dump, use last binlog replication pos (mysql-bin.000001, 120) or
GTID %!s(<nil>)
2018/06/02 16:13:21 INFO begin to sync binlog from position (mysql-bin.000001, 120)
2018/06/02 16:13:21 INFO register slave for master server 172.16.0.101:3306
2018/06/02 16:13:21 INFO start sync binlog at binlog file (mysql-bin.000001, 120)
2018/06/02 16:13:21 INFO rotate to (mysql-bin.000001, 120)
2018/06/02 16:13:21 INFO rotate binlog to (mysql-bin.000001, 120)
2018/06/02 16:13:21 INFO save position (mysql-bin.000001, 120)
```

9. Test: Data insertion, update, and deletion performed in MySQL can be reflected in ES.

User experience

- `go-mysql-elasticsearch` provides the most basic capability to sync data from MySQL to ES in real time. If your business requires more complex features such as modifying the MySQL table structure during operation, you can customize it for secondary development.
- The tool is not good at handling exceptions. If parsing binlog events fails, it will throw an exception directly.
- As described by the tool developer, the tool has not been applied to a production environment, so you are recommended to read the source code carefully to understand its pros and cons.

Syncing data to an ES cluster using mypipe

mypipe is a MySQL binlog sync tool initially designed to send binlog events to Kafka. It can also sync data to any storage media based on your actual business needs. For more information, see its

©2013-2019 Tencent Cloud. All rights reserved.

GitHub page.

Use limits

1. MySQL binlog must use ROW mode.
2. You need to grant the account used to connect to MySQL the REPLICATION permission.

```
GRANT REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'elastic'@'%' IDENTIFIED BY 'Elastic_123'
```

3. mypipe just parses the contents of the binlog, encodes them into Avro format, and pushes them to a Kafka broker instead of directly pushing data to Kafka. If you need to sync data to an ES cluster, you can consume the data from Kafka first and then write it to ES.
4. To consume messages in Kafka (operations log of MySQL), you need to perform Avro parsing on the message contents and obtain the corresponding data operations for further processing. mypipe encapsulates a `KafkaGenericMutationAvroConsumer` class that can be directly inherited for use. You can also implement the parsing on your own.
5. mypipe supports syncing only binlog but not existing data, i.e., after the mypipe program is started, existing data in MySQL cannot be synced.

How to use

1. Run the `git clone https://github.com/mardambey/mypipe.git` command.

2. Run the `./sbt package` command.

3. Configure `mypipe-runner/src/main/resources/application.conf` .

```
mypipe {

# Avro schema repository client class name
schema-repo-client = "mypipe.avro.schema.SchemaRepo"

# consumers represent sources for mysql binary logs
consumers {

localhost {
# database "host:port:user:pass" array
source = "172.16.0.101:3306:elastic:Elastic_123"
}
}

# data producers export data out (stdout, other stores, external services, etc.)
producers {
```

©2013-2019 Tencent Cloud. All rights reserved.

```
kafka-generic {
class = "mypipe.kafka.producer.KafkaMutationGenericAvroProducer"
}
}

# pipes join consumers and producers
pipes {

kafka-generic {
enabled = true
consumers = ["localhost"]
producer {
kafka-generic {
metadata-brokers = "172.16.16.22:9092"
}
}
binlog-position-repo {
# saved to a file, this is the default if unspecified
class = "mypipe.api.repo.ConfigurableFileBasedBinaryLogPositionRepository"
config {
file-prefix = "stdout-00" # required if binlog-position-repo is specifiec
data-dir = "/tmp/mypipe/data" # defaults to mypipe.data-dir if not present
}
}
}
}
}
```

4. Configure `mypipe-api/src/main/resources/reference.conf` by modifying the `include-event-condition` option to specify the database and table to be synced.

```
include-event-condition = """ db == "webservice" && table =="building" """
```

5. Create `topic: webservice_building_generic` on the Kafka broker. By default, mypipe uses `${db}_${table}_generic` as the topic name to send data to the topic.

6. Run the `./sbt "project runner" "runMain mypipe.runner.PipeRunner"` command.

7. Test: Insert data into the MySQL building table and write a simple consumer to consume the messages pushed to Kafka by mypipe.

8. The consumed data that has not been parsed yet is as follows:

©2013-2019 Tencent Cloud. All rights reserved.

```
ConsumerRecord(topic=u'webservice_building_generic', partition=0, offset=2, timestamp=None, timestamp_type=None, key=None, value='¥x00¥x01¥x00¥x00¥x14webservice¥x10building¥xcc¥x01¥x02¥x91,¥xae¥xa3fc¥x11¥xe8¥xa1¥xaaRT¥x00Z¥xf9¥xab¥x00¥x00¥x04¥x18BuildingName¥x06xxx¥x14BuildingId¥nId-10¥x00¥x02¥x04Id¥xd4%¥x00', checksum=128384379, serialized_key_size=-1, serialized_value_size=88)
```

User experience

- mypipe is more sophisticated than go-mysql-elasticsearch. It supports performing ALTER TABLE operations during running and configuring different policies to handle exceptions of binlog parsing.
- mypipe cannot sync existing data. If you need this feature, you can use other means to sync full data and then use mypipe for incremental sync.
- mypipe only syncs binlog. If you need to sync data to ES, separate development will be required.

©2013-2019 Tencent Cloud. All rights reserved.

Building a Log Analysis System

Last updated : 2019-11-12 16:08:31

An instance provided by ES consists of an ES cluster and a Kibana Console. The former can be accessed via the VIP address and port of your VPC, and the latter provides a public IP address for you to access from a browser. Currently, you can only ingest your data into the ES cluster on your own. The following describes how to import your logs into ES and access Kibana from a browser to perform query and analysis by taking the most typical log analysis architectures Filebeat + Elasticsearch + Kibana and Logstash + Elasticsearch + Kibana as examples.

Filebeat + Elasticsearch + Kibana

Deploying Filebeat

1. Download the Filebeat package and decompress it

   > The Filebeat version should be compatible with the ES version.

   ```
   wget https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-6.4.3-linux-x86_64.tar.gz
   tar xvf filebeat-6.4.3-linux-x86_64.tar.gz
   ```

2. Configure Filebeat
   In this example, NGINX log is used as the input source, and the output is configured as the private VIP address and port of the ES cluster. If you use a Platinum Edition cluster, you need to add username and password to the output.
   Enter the `filebeat-6.4.3-linux-x86_64` directory and modify the `filebeat.yml` configuration file, as shown below:

   ```
   filebeat.inputs:
   - type: log
   enabled: true
   paths:
   - /var/log/nginx/access.log
   output.elasticsearch:
   hosts: ["10.0.130.91:9200"]
   protocol: "http"
   ```

©2013-2019 Tencent Cloud. All rights reserved.

```
username: "elastic"
password: "test"
```

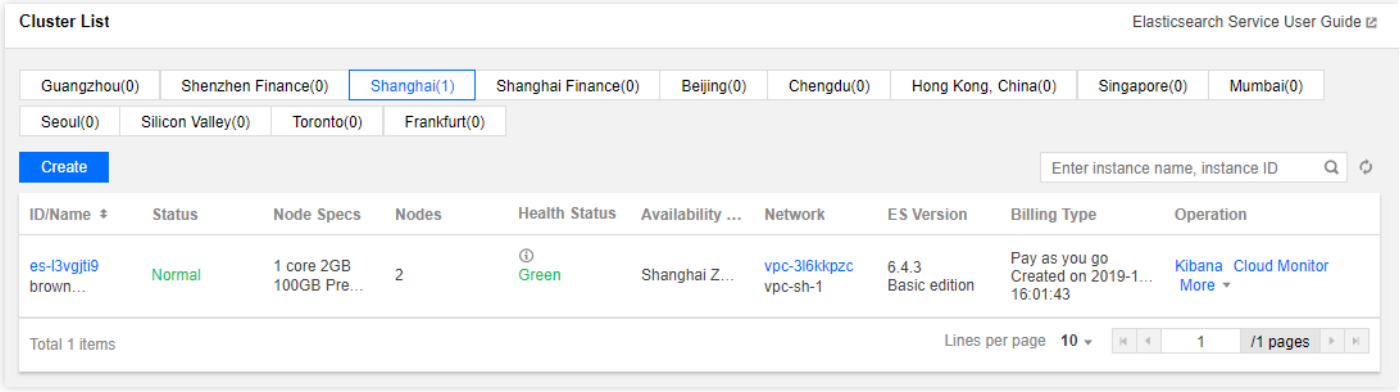
3. Run Filebeat

   In the `filebeat-6.4.3-linux-x86_64` directory, run:

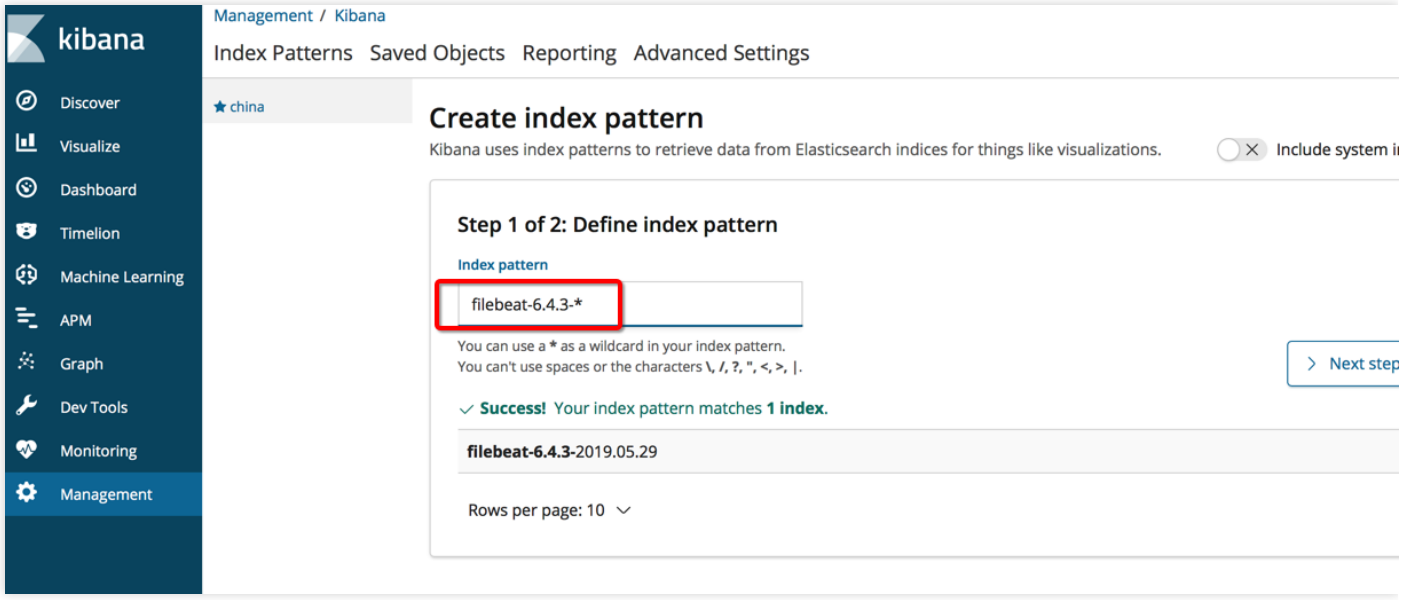
```
nohup ./filebeat -c filebeat.yml 2>&1 >/dev/null &
```

Querying a log

1. On the cluster list page in the ES Console, select **Operation** > **Kibana** to enter the Kibana Console.

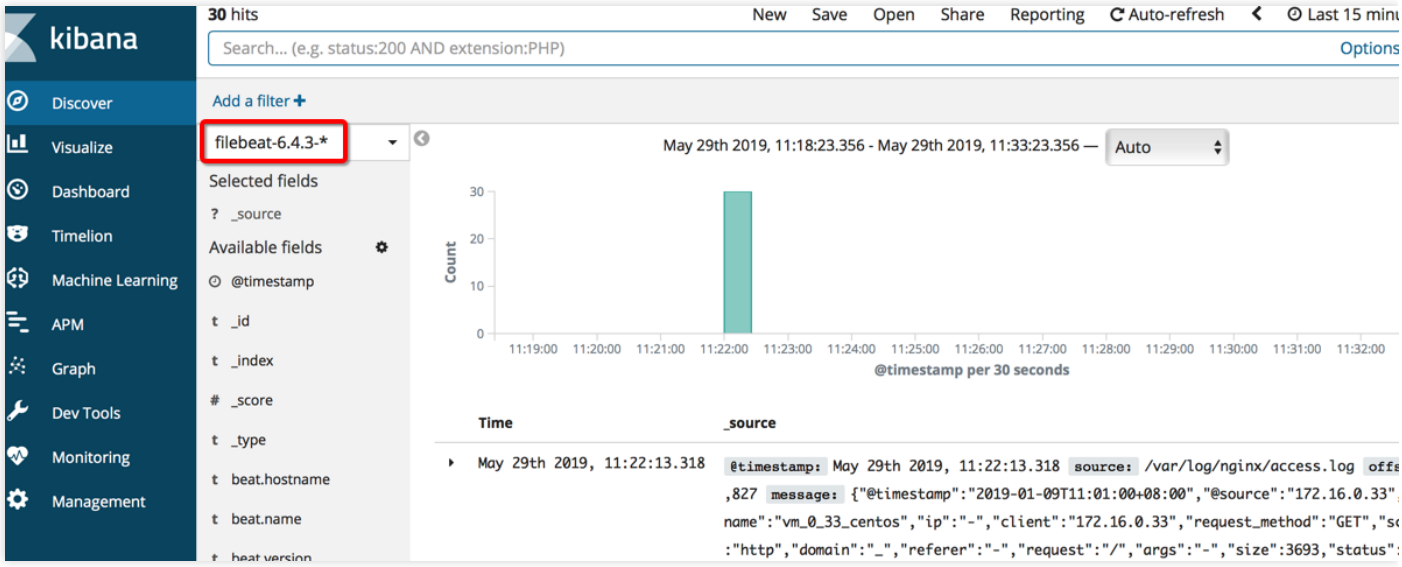


2. Go to **Management** > **Index Patterns** and add an index pattern named `filebeat-6.4.3-*` .



©2013-2019 Tencent Cloud. All rights reserved.

3. Click **Discover** and select the `filebeat-6.4.3-*` index to retrieve the NGINX access logs.



Logstash + Elasticsearch + Kibana

Environment preparations

- You need to create one or more CVM instances as needed in the same VPC as the ES cluster and deploy the Logstash component on them;
- A CVM instance should have at least 2 GB of memory;
- You need to install Java 8 or above on the CVM instances.

Deploying Logstash

1. Download the Logstash package and decompress it

> The Logstash version should be compatible with the ES version.

```
wget https://artifacts.elastic.co/downloads/logstash/logstash-6.4.3.tar.gz
tar xvf logstash-6.4.3.tar.gz
```

2. Configure Logstash
   In this example, NGINX log is used as the input source, and the output is configured as the private VIP address and port of the ES cluster. Create a `test.conf` configuration file, as shown below:

©2013-2019 Tencent Cloud. All rights reserved.

```
input {
file {
path => "/var/log/nginx/access.log" # Path to the NGINX access log
start_position => "beginning" # Read the log from the beginning of the file. If this parameter is not set, the log will be read when data is written to the file, just like `tail -f`
}
}
filter {
}
output {
elasticsearch {
hosts => ["http://172.16.0.145:9200"] # Private VIP address and port of the ES cluster
index => "nginx_access-%{+YYYY.MM.dd}" # Index name. Indices are automatically created on a daily basis
user => "elastic" # Username
password => "yinan_test" # Password
}
}
```

The ES cluster is configured to automatically create an index by default. The `nginx_access-%{+YYYY.MM.dd}` index will be automatically created, and unless you need to set the mapping of the fields in the index in advance, you don't need to additionally call the API of ES to create indices.

3. Start Logstash
   Enter the extracted `logstash-6.4.3` directory of the Logstash package and run the following command to start Logstash in the background. Be sure to enter the path you create as the path to the configuration file.

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

View the `logs` directory in the `logstash-6.4.3` directory to confirm that Logstash has been started and can record the following logs:

```
Sending Logstash logs to /root/logstash-6.4.3/logs which is now configured via log4j2.properties
[2019-05-29T12:20:26,630][INFO ][logstash.setting.writabledirectory] Creating directory {:setting=>"path.queue", :path=>"/root/logstash-6.4.3/data/queue"}
[2019-05-29T12:20:26,639][INFO ][logstash.setting.writabledirectory] Creating directory {:setting=>"path.dead_letter_queue", :path=>"/root/logstash-6.4.3/data/dead_letter_queue"}
[2019-05-29T12:20:27,125][WARN ][logstash.config.source.multilocal] Ignoring the 'pipelines.yml' file because modules or command line options are specified
[2019-05-29T12:20:27,167][INFO ][logstash.agent ] No persistent UUID file found. Generating ne
```

©2013-2019 Tencent Cloud. All rights reserved.

```
w UUID {:uuid=>"2e19b294-2b69-4da1-b87f-f4cb4a171b9c", :path=>"/root/logstash-6.4.3/data/uui
d"}
[2019-05-29T12:20:27,843][INFO ][logstash.runner ] Starting Logstash {"logstash.version"=>"6.
4.3"}
[2019-05-29T12:20:30,067][INFO ][logstash.pipeline ] Starting pipeline {:pipeline_id=>"main",
"pipeline.workers"=>1, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50}
[2019-05-29T12:20:30,871][INFO ][logstash.outputs.elasticsearch] Elasticsearch pool URLs updat
ed {:changes=>{:removed=>[], :added=>[http://elastic:xxxxxx@10.0.130.91:10880/]}}
[2019-05-29T12:20:30,901][INFO ][logstash.outputs.elasticsearch] Running health check to see i
f an Elasticsearch connection is working {:healthcheck_url=>http://elastic:xxxxxx@10.0.130.91:
10880/, :path=>"/"}
[2019-05-29T12:20:31,449][WARN ][logstash.outputs.elasticsearch] Restored connection to ES ins
tance {:url=>"http://elastic:xxxxxx@10.0.130.91:10880/"}
[2019-05-29T12:20:31,567][INFO ][logstash.outputs.elasticsearch] ES Output version determined
{:es_version=>6}
[2019-05-29T12:20:31,574][WARN ][logstash.outputs.elasticsearch] Detected a 6.x and above clus
ter: the `type` event field won't be used to determine the document _type {:es_version=>6}
[2019-05-29T12:20:31,670][INFO ][logstash.outputs.elasticsearch] New Elasticsearch output {:cl
ass=>"LogStash::Outputs::ElasticSearch", :hosts=>["http://10.0.130.91:10880"]}
[2019-05-29T12:20:31,749][INFO ][logstash.outputs.elasticsearch] Using mapping template from
{:path=>nil}
[2019-05-29T12:20:31,840][INFO ][logstash.outputs.elasticsearch] Attempting to install templat
e {:manage_template=>{"template"=>"logstash-*", "version"=>60001, "settings"=>{"index.refresh_
interval"=>"5s"}, "mappings"=>{"_default_"=>{"dynamic_templates"=>[{"message_field"=>{"path_ma
tch"=>"message", "match_mapping_type"=>"string", "mapping"=>{"type"=>"text", "norms"=>fals
e}}}, {"string_fields"=>{"match"=>"*", "match_mapping_type"=>"string", "mapping"=>{"type"=>"te
xt", "norms"=>false, "fields"=>{"keyword"=>{"type"=>"keyword", "ignore_above"=>256}}}}}], "pro
perties"=>{"@timestamp"=>{"type"=>"date"}, "@version"=>{"type"=>"keyword"}, "geoip"=>{"dynami
c"=>true, "properties"=>{"ip"=>{"type"=>"ip"}, "location"=>{"type"=>"geo_point"}, "latitude"=>
{"type"=>"half_float"}, "longitude"=>{"type"=>"half_float"}}}}}}}}
[2019-05-29T12:20:32,094][INFO ][logstash.outputs.elasticsearch] Installing elasticsearch temp
late to _template/logstash
[2019-05-29T12:20:33,242][INFO ][logstash.inputs.file ] No sincedb_path set, generating one ba
sed on the "path" setting {:sincedb_path=>"/root/logstash-6.4.3/data/plugins/inputs/file/.sinc
edb_d883144359d3b4f516b37dba51fab2a2", :path=>["/var/log/nginx/access.log"]}
[2019-05-29T12:20:33,329][INFO ][logstash.pipeline ] Pipeline started successfully {:pipeline_
id=>"main", :thread=>"#<Thread:0x12bdd65 run>"}
[2019-05-29T12:20:33,544][INFO ][logstash.agent ] Pipelines running {:count=>1, :running_pipel
ines=>[:main], :non_running_pipelines=>[]}
[2019-05-29T12:20:33,581][INFO ][filewatch.observingtail ] START, creating Discoverer, Watch w
ith file and sincedb collections
[2019-05-29T12:20:34,368][INFO ][logstash.agent ] Successfully started Logstash API endpoint
{:port=>9600}
```

For more information on the features of Logstash, see Elastic's official documentation.

©2013-2019 Tencent Cloud. All rights reserved.

Querying a log

See Querying a log.

For more information on the features of the Kibana Console, see Elastic's official documentation.

©2013-2019 Tencent Cloud. All rights reserved.

Managing Indices with Curator

Last updated : 2020-07-14 12:12:37

Curator is Elasticsearch's official tool for managing indices. It can perform a wide variety of index lifecycle management tasks, such as clearing indices created 7 days ago, backing up specified indices regularly every day, and migrating indices from a hot node to a warm node regularly. For more information on operations in Curator, please see Actions.

Curator provides a CLI which allows you to configure the tasks to be executed using parameters. It also comes with a complete set of Python APIs that can be used together with SCF, such as automatically deleting expired data from Elasticsearch using Curator. SCF has a pre-defined template for Curator, which can be run after simple parameter configuration. For more information on how to use SCF, please see here.

Curator Usage Example

The following describes how to configure and run Curator to delete expired indices regularly.

Installation

Purchase a CVM instance in the VPC where your Elasticsearch cluster resides and install the Curator package via pip.

```
pip install elasticsearch-curator
```

Running as command line parameters

The following command filters out the indices named in the format of `logstash-20xx-xx-xx` and created 7 days ago and then deletes them.

> ⚠ **Note :**
>
> The sample code performs a deletion operation to clear your data. Make sure that the above statement has been tested in a non-production environment. You can add the `--dry-run` parameter for testing purpose to avoid actually deleting data.

```
curator_cli --host 10.0.0.2:9200 --http_auth 'user:passwd' delete_indices --filter_list '[{"filte
rtype": "pattern", "kind": "prefix", "value": "logstash-"}, {"filtertype": "age", "source": "nam
e", "direction": "older", "timestring": "%Y.%m.%d", "unit": "days", unit_count: 7}]'
```

©2013-2019 Tencent Cloud. All rights reserved.

Running as configuration files

If your operations are complex, there are a lot of parameters, or you don't want to use command line parameters, you can place the parameter in configuration files.
In the specified `config` directory, you need to edit the config.yml and action.yml configuration files.

```
curator_cli --config PATH
```

Scheduled run

If you need a scheduled run, you can configure the command to a crontab on Linux, or directly use the timer trigger feature of SCF mentioned above.

Using the API

For more information on how to use the Python APIs, please see here.

©2013-2019 Tencent Cloud. All rights reserved.

Data Migration

Last updated : 2020-05-12 15:01:54

If you want to migrate your Elasticsearch cluster built in Tencent Cloud or purchased from another cloud vendor to Tencent Cloud ES, you can choose a suitable migration solution based on your business needs. If your business can be paused for write operations, you can migrate your data using the following tools:

- elasticsearch-dump
- snapshot
- reindex
- logstash

elasticsearch-dump

Use cases

Applicable to scenarios where the amount of data is small and there are only a few indices to be migrated.

How to use

elasticsearch-dump is an open-source Elasticsearch data migration tool available on GitHub.

1. Install elasticsearch-dump
   elasticsearch-dump is developed using Node.js and can be installed directly using the npm package manager:

```
npm install elasticdump -g
```

2. Main Parameters

```
--input: the source address in the format of {protocol}://{host}:{port}/{index}, which can be an Elasticsearch cluster URL, file, or stdin and allows you to specify an index
--input-index: the index in the source cluster
--output: the target address in the format of {protocol}://{host}:{port}/{index}, which can be an Elasticsearch cluster URL, file, or stdout and allows you to specify an index
--output-index: the index in the target cluster
--type: the migration type, which is data by default, indicating that only data will be migrated. Value range: settings, analyzer, data, mapping, alias
```

3. Migrate a single index
   The following operation migrates the `companydatabase` index in the `172.16.0.39` cluster to the

©2013-2019 Tencent Cloud. All rights reserved.

`172.16.0.20` cluster by running the `elasticdump` command.

> The first command migrates the settings of the index. If you directly migrate the mapping or data, the configuration information of the index in the source cluster will be lost, such as the number of shards and number of replicas. Of course, you can also directly create an index in the target cluster first before syncing the mapping and data.

```
elasticdump --input=http://172.16.0.39:9200/companydatabase --output=http://172.16.0.20:9200/c
ompanydatabase --type=settings
elasticdump --input=http://172.16.0.39:9200/companydatabase --output=http://172.16.0.20:9200/c
ompanydatabase --type=mapping
elasticdump --input=http://172.16.0.39:9200/companydatabase --output=http://172.16.0.20:9200/c
ompanydatabase --type=data
```

4. Migrate all indices
   The following operation migrates all the indices in the `172.16.0.39` cluster to the `172.16.0.20` cluster by running the `elasticdump` command.

> This operation cannot migrate the configurations of indices, such as the number of shards and number of replicas. You must migrate the configuration of each index separately, or directly create an index in the target cluster first before migrating the data.

```
elasticdump --input=http://172.16.0.39:9200 --output=http://172.16.0.20:9200
```

snapshot

Use cases

Suitable for migrating a large amount of data.

How to use

The snapshot API is an API used by Elasticsearch to back up and restore data. You can use it to migrate data from one cluster to another by creating a snapshot of the data in the source cluster and then restoring it to the target cluster.

©2013-2019 Tencent Cloud. All rights reserved.

> The major version number of the target cluster (such as 5 in 5.6.4) should be greater than or equal to that of the source cluster. A snapshot created from a 1.x cluster cannot be restored to a 5.x cluster.

1. Create a repository in the source cluster
   Before creating a snapshot, you must create a repository, which can contain multiple snapshot files. There are several types of repositories as detailed below:
   - fs: a shared file system where snapshot files are stored.
   - url: the URL path to a specified file system, which supports HTTP, HTTPS, FTP, file, and jar protocols.
   - s3: AWS S3. The snapshot is stored in S3, which is supported as a plugin, so the repository-s3 plugin should be installed.
   - hdfs: the snapshot is stored in HDFS, which is supported as a plugin, so the repository-hdfs plugin should be installed.
   - cos: the snapshot is stored in COS, which is supported as a plugin, so the elasticsearch-repository-cos plugin should be installed.

   To migrate your data from a self-built Elasticsearch cluster to a Tencent Cloud ES cluster, you can directly use an fs repository, but you need to set the repository path in the `elasticsearch.yml` configuration file.

   ```
   path.repo: ["/usr/local/services/test"]
   ```

   Then, call the snapshot API to create a repository.

   ```
   curl -XPUT http://172.16.0.39:9200/_snapshot/my_backup -H 'Content-Type: application/json' -d
   '{
   "type": "fs",
   "settings": {
   "location": "/usr/local/services/test"
   "compress": true
   }
   }'
   ```

To migrate your data from an Elasticsearch cluster provided by another cloud vendor to a Tencent Cloud ES cluster or between ES clusters, you can use the corresponding repository

©2013-2019 Tencent Cloud. All rights reserved.

type provided by the cloud vendor, such as S3 by AWS, OSS by Alibaba Cloud, and COS by Tencent Cloud.

```
curl -XPUT http://172.16.0.39:9200/_snapshot/my_s3_repository
{
"type": "s3",
"settings": {
"bucket": "my_bucket_name",
"region": "us-west"
}
}
```

2. Create a snapshot in the source cluster
   You can call the snapshot API to create a snapshot in the created repository. When creating a snapshot, you can specify an index or the contents of the snapshot. For more information on API parameters, please see Snapshot and Restore.
   ```
   curl -XPUT http://172.16.0.39:9200/_snapshot/my_backup/snapshot_1?wait_for_completion=true
   ```
3. Create a repository in the target cluster
   Creating a repository in the target cluster is similar to creating a repository in the source cluster. You can create a COS bucket in Tencent Cloud for storing the repository.
4. Move the snapshot created in the source cluster to the repository in the target cluster
   Upload the snapshot created in the source cluster to the repository created in the target cluster.
5. Restore data from the snapshot
   ```
   curl -XPUT http://172.16.0.20:9200/_snapshot/my_backup/snapshot_1/_restore
   ```
6. View the status of snapshot restoration
   ```
   curl http://172.16.0.20:9200/_snapshot/_status
   ```

reindex

The reindex API provided by Elasticsearch allows you to migrate data by importing data from the source cluster to the current cluster; however, this can be implemented only in Elasticsearch and is not supported by the current version. The following describes how to use the reindex API:

1. Configure the `reindex.remote.whitelist` parameter
   You need to configure this parameter in the target cluster to specify the whitelist of remote clusters that can be reindexed.

©2013-2019 Tencent Cloud. All rights reserved.

2. Call the reindex API

   The following operation involves querying the index named `test1` by `elasticsearch` in the `title` field from the source cluster and writing the result to the `test2` index of the current cluster.

```
POST _reindex
{
"source": {
"remote": {
"host": "http://172.16.0.39:9200"
},
"index": "test1",
"query": {
"match": {
"title": "elasticsearch"
}
}
},
"dest": {
"index": "test2"
}
}
```

logstash

Logstash allows you to read data from an Elasticsearch cluster and then write it to another cluster; therefore, you can use Logstash for data migration. The specific configuration file is as follows:

```
input {
elasticsearch {
hosts => ["http://172.16.0.39:9200"]
index => "*"
docinfo => true
}
}
output {
elasticsearch {
hosts => ["http://172.16.0.20:9200"]
index => "%{[@metadata][_index]}"
}
}
```

©2013-2019 Tencent Cloud. All rights reserved.

The above configuration file syncs all the indices in the source cluster to the target cluster, and it can also be set to only sync specified indices. For more information on the features of Logstash, please see Configuring Logstash.

Summary

1. When elasticsearch-dump or Logstash is used to migrate data from one cluster to another, the machine used to perform the migration task is required to have access to both clusters at the same time, because migration cannot be performed if there is no network connection. This limitation does not apply if snapshot is used, as it is completely offline. Therefore, elasticsearch-dump and Logstash are more suitable for migrating data between clusters on the same network. If you want to migrate your data between cloud vendors, such as from an Alibaba Cloud Elasticsearch cluster to a Tencent Cloud ES cluster, you can use snapshot or establish a connection between the vendors to achieve cluster connectivity, which, however, is costly.
2. elasticsearchdump is similar to MySQL's mysqldump which is used for data backup, and both of them perform logical backup that involves exporting data entries one by one and then importing them. Therefore, this tool is suitable for migrating a small amount of data.
3. Snapshot is suitable for migrating a large amount of data.

©2013-2019 Tencent Cloud. All rights reserved.

Data Ingestion into ES

Last updated : 2020-05-12 15:03:31

ES allows access to your cluster through private VIP within your VPC. You can write code to access your cluster through the Elasticsearch REST client and import your data into the cluster. You can also ingest your data through Elasticsearch's official components such as Logstash and Beats.
This document takes the official components Logstash and Beats as examples to describe how to connect your data source of different types to ES.

Preparations

You need to create a CVM instance or a Docker cluster in the same VPC as the ES cluster, as accessing the ES cluster needs to be done within the VPC.

Using Logstash to Access ES Cluster

Accessing ES cluster from CVM

1. Install and deploy Logstash and Java 8.

```
wget https://artifacts.elastic.co/downloads/logstash/logstash-5.6.4.tar.gz
tar xvf logstash-5.6.4.tar.gz
yum install java-1.8.0-openjdk java-1.8.0-openjdk-devel -y
```

> Please note that the Logstash version should be the same as the Elasticsearch version.

2. Customize the `¥*.conf` configuration file based on the data source type. For more information, please see Data Source Configuration File Description.
3. Run Logstash.

```
nohup ./bin/logstash -f ~/*.conf 2>&1 >/dev/null &
```

Accessing ES cluster from Docker

Creating Docker cluster

1. Pull the official image of Logstash.

©2013-2019 Tencent Cloud. All rights reserved.

```
docker pull docker.elastic.co/logstash/logstash:5.6.9
```

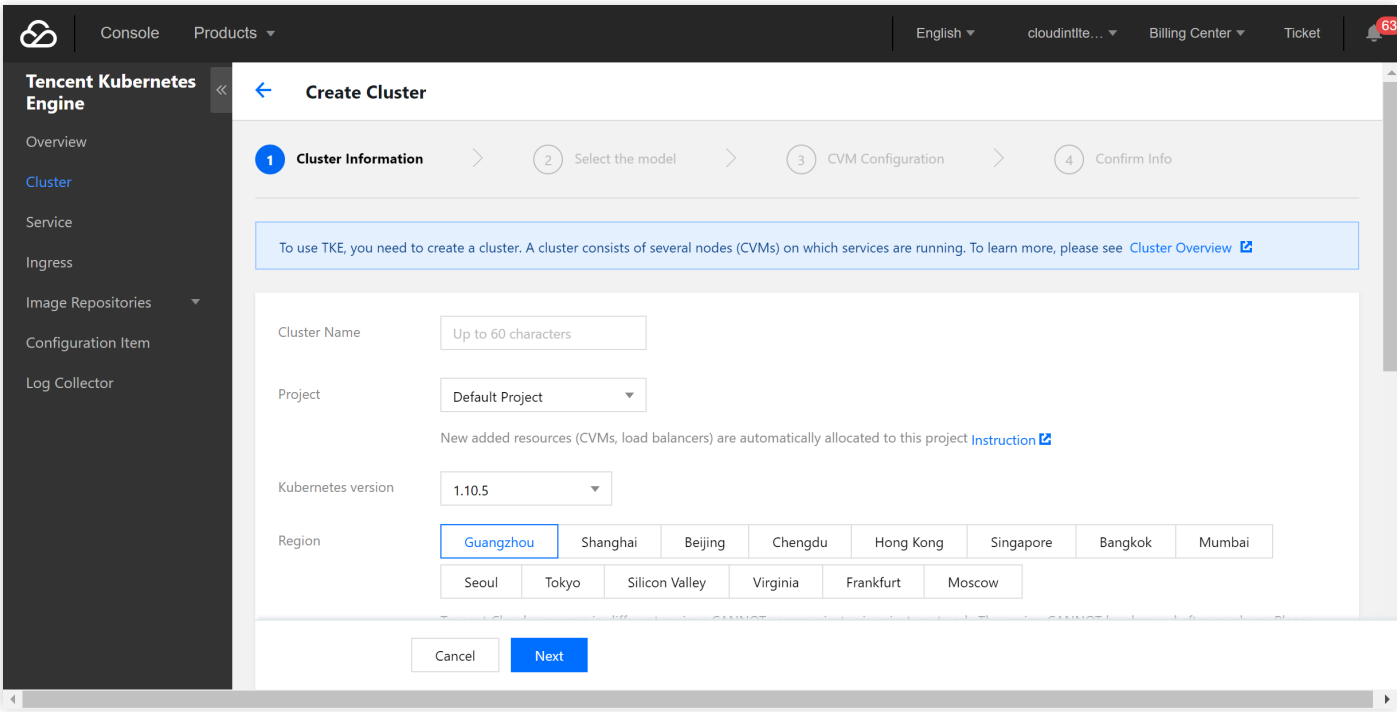
2. Customize the `¥*.conf` configuration file based on the data source type and place it in the `/usr/share/logstash/pipeline/` directory which can be customized.
3. Run Logstash.

```
docker run --rm -it -v ~/pipeline/:/usr/share/logstash/pipeline/ docker.elastic.co/logstash/logstash:5.6.9
```

Using TKE

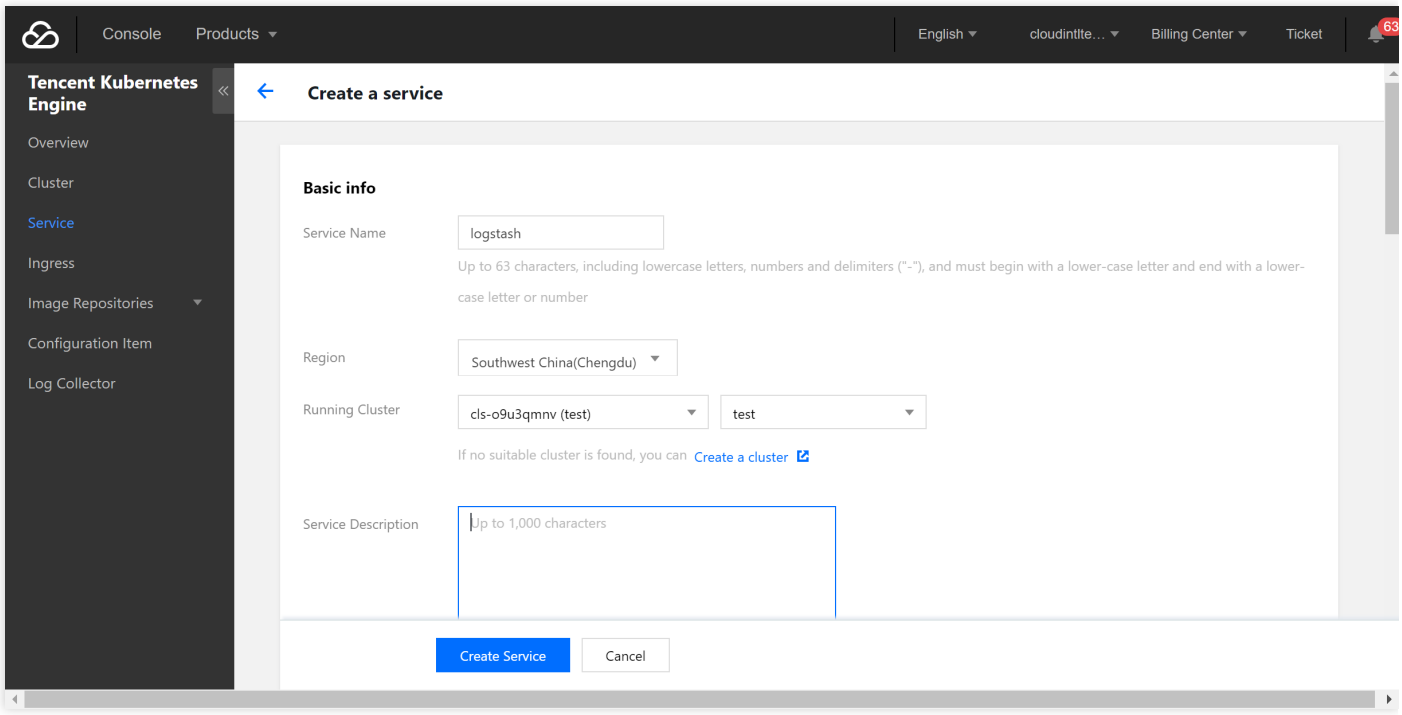
Tencent Cloud Docker clusters run on CVM instances, so you need to create a CVM cluster in the TKE Console first.

1. Log in to the TKE Console and select **Cluster** > **Create** on the left sidebar to create a cluster.



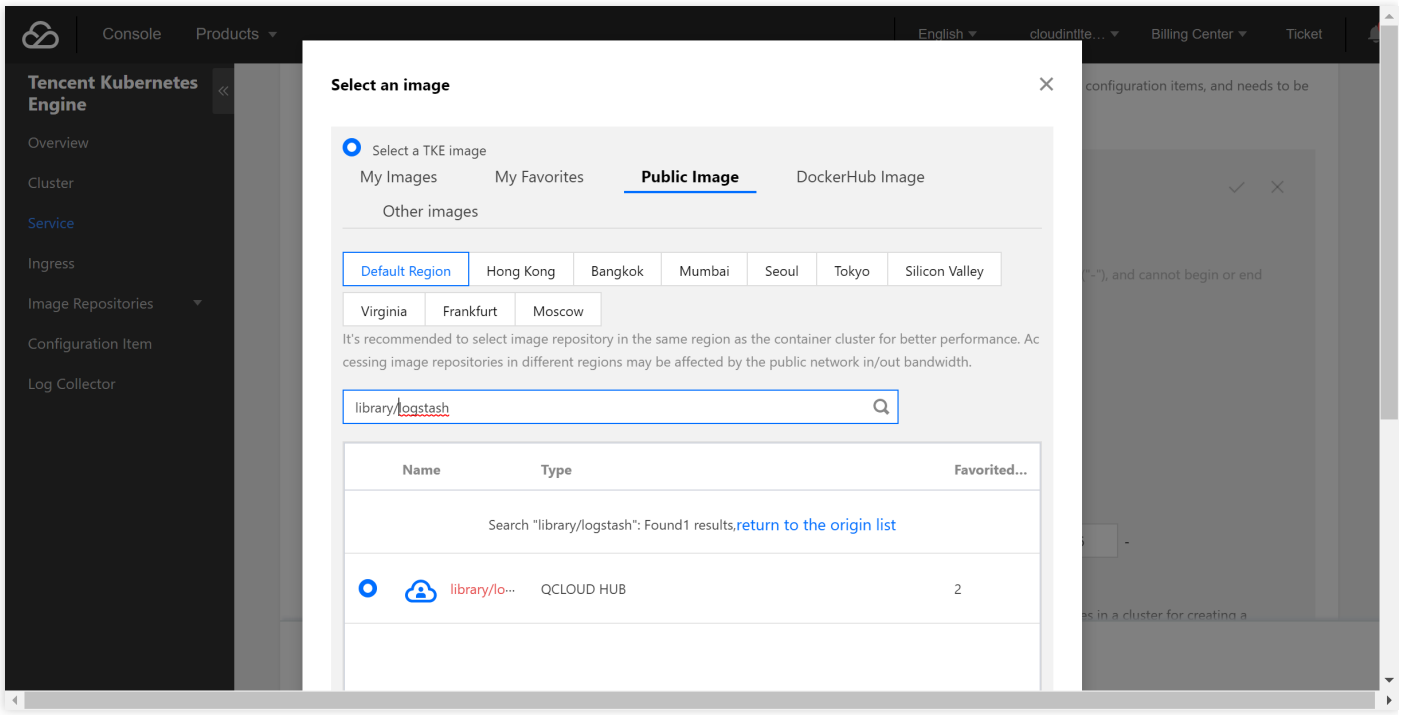
©2013-2019 Tencent Cloud. All rights reserved.

2. Select **Service** on the left sidebar and click **Create** to create a service.



3. Select the official image of Logstash.

   In this example, the Logstash image provided by TencentHub image registry is used. You can also create a Logstash image by yourself.

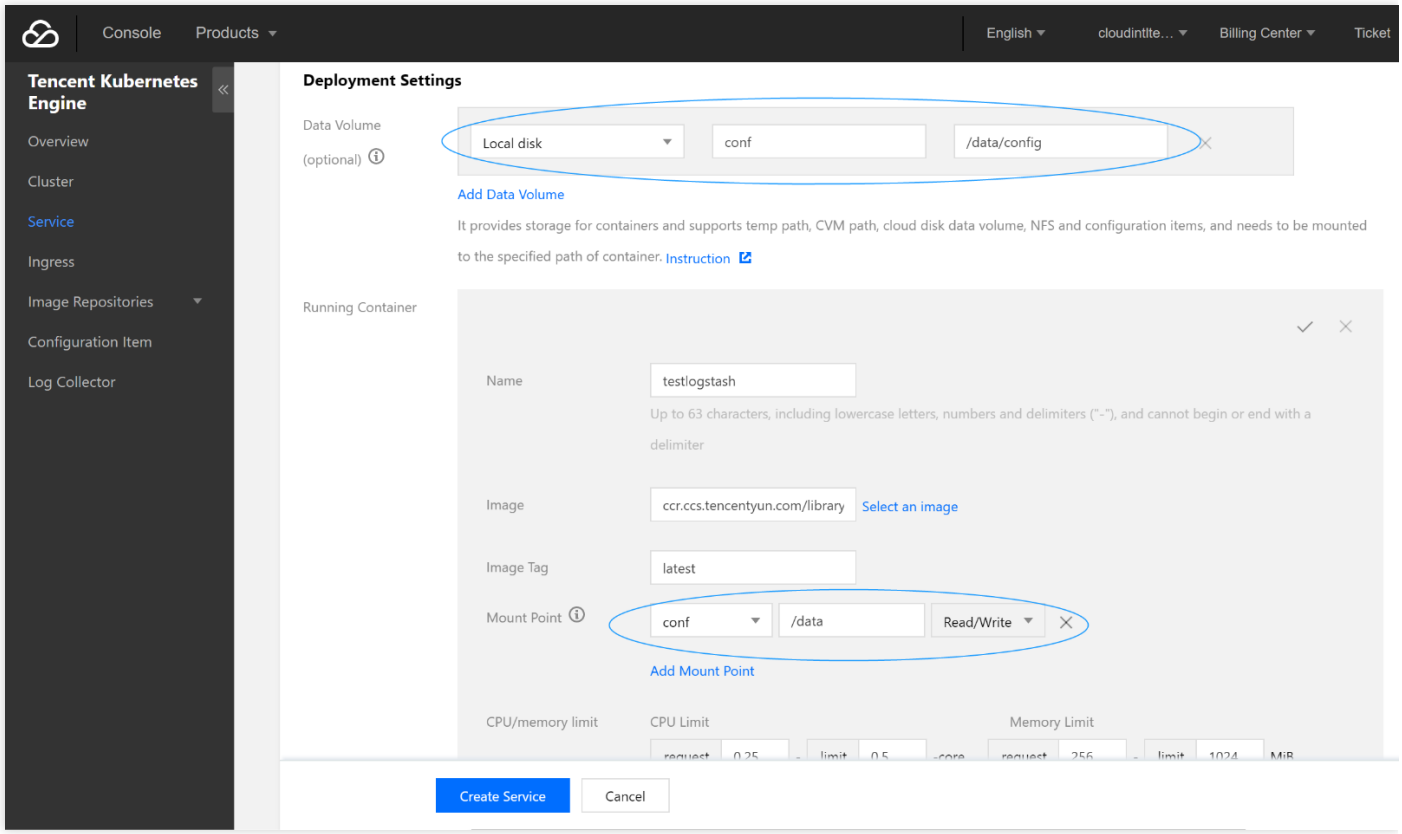


4. Create a data volume.

   Create a data volume to store the Logstash configuration file. In this example, a configuration file

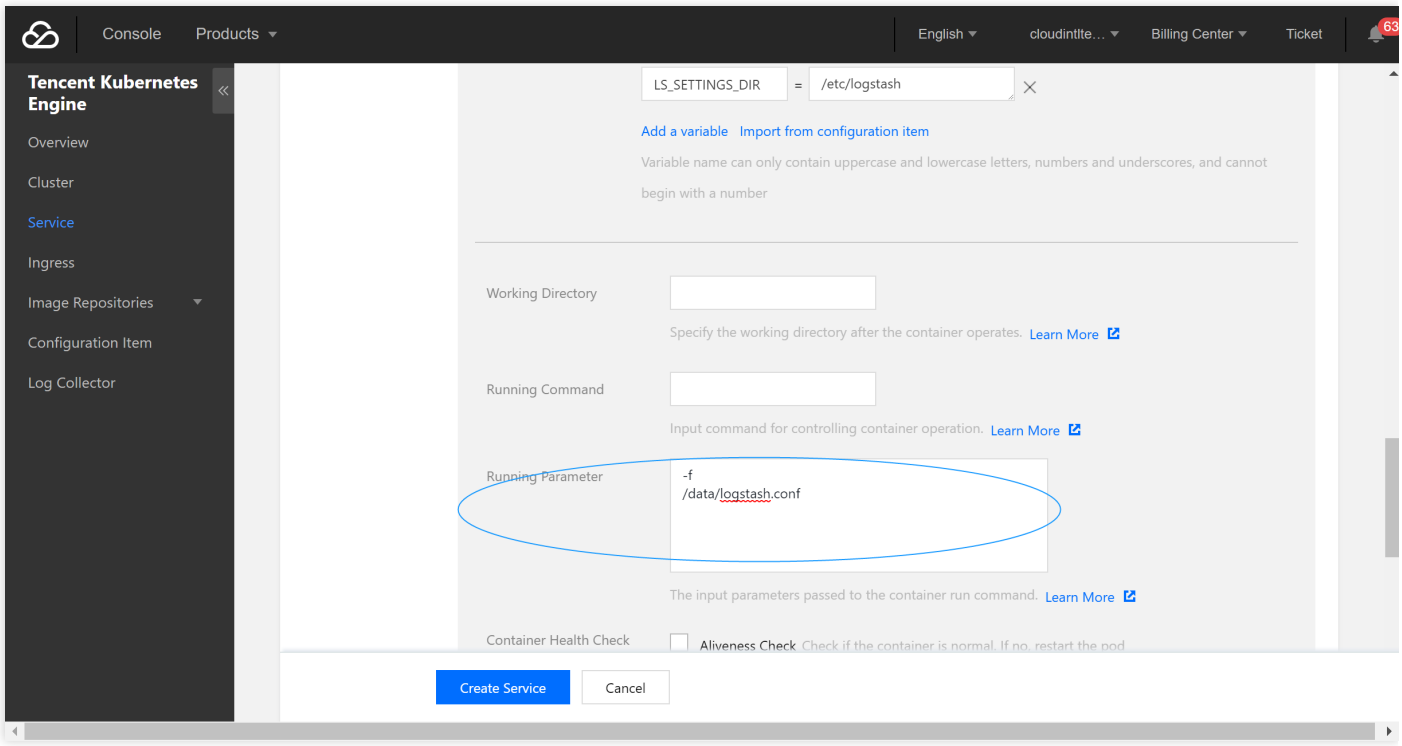
©2013-2019 Tencent Cloud. All rights reserved.

named `logstash.conf` is added to the `/data/config` directory on the CVM instance and mounted to the `/data` directory of Docker, so that the `logstash.conf` file can be read when the container starts.

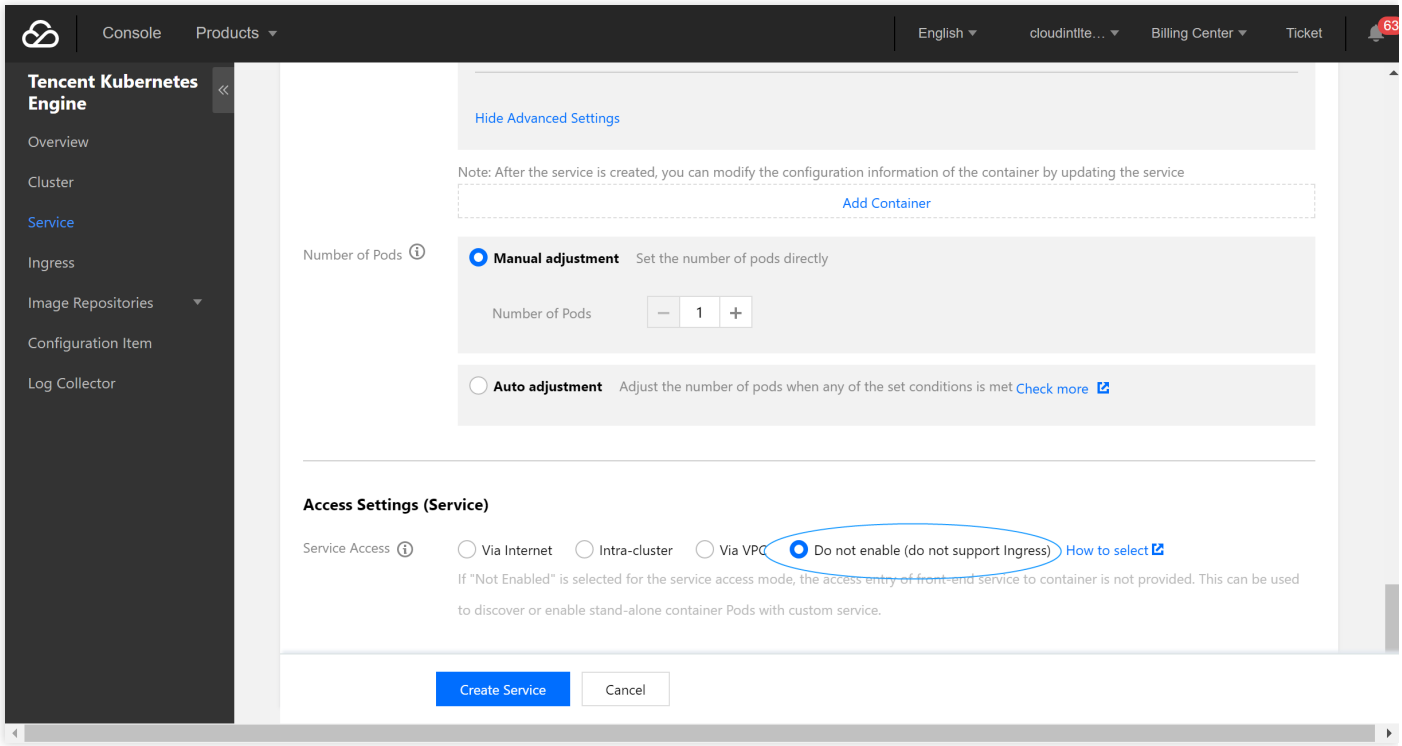


©2013-2019 Tencent Cloud. All rights reserved.

5. Configure the execution parameters.



6. Configure the service parameters and create a service as needed.



Configuration file description

©2013-2019 Tencent Cloud. All rights reserved.

File data sources

```
input {
file {
path => "/var/log/nginx/access.log" # File path
}
}
filter {
}
output {
elasticsearch {
hosts => ["http://172.16.0.89:9200"] # Private VIP address and port of the ES cluster
index => "nginx_access-%{+YYYY.MM.dd}" # Custom index name suffixed with date. One index is gener
ated per day
}
}
```

For more information on connecting file data sources, please see File input plugin.

Kafka data sources

```
input{
kafka{
bootstrap_servers => ["172.16.16.22:9092"]
client_id => "test"
group_id => "test"
auto_offset_reset => "latest" # Start consumption from the latest offset
consumer_threads => 5
decorate_events => true # This attribute will bring the current topic, offset, group, partition,
and other information into the message
topics => ["test1","test2"] # Array type. Multiple topics can be configured
type => "test" # Data source identification field
}
}

output {
elasticsearch {
hosts => ["http://172.16.0.89:9200"] # Private VIP address and port of the ES cluster
index => "test_kafka"
}
}
```

For more information on connection Kafka data sources, please see Kafka input plugin.

Database data sources connected with JDBC

©2013-2019 Tencent Cloud. All rights reserved.

```
input {
jdbc {
# MySQL database address
jdbc_connection_string => "jdbc:mysql://172.16.32.14:3306/test"
# Username and password
jdbc_user => "root"
jdbc_password => "Elastic123"
# Driver jar package. You need to download the jar when installing and deploying Logstash on your
own as it is not provided by Logstash by default
jdbc_driver_library => "/usr/local/services/logstash-5.6.4/lib/mysql-connector-java-5.1.40.jar"
# Driver class name
jdbc_driver_class => "com.mysql.jdbc.Driver"
jdbc_paging_enabled => "true"
jdbc_page_size => "50000"
# Path and name of the SQL file to be executed
#statement_filepath => "test.sql"
# SQL statement to be executed
statement => "select * from test_es"
# Set the monitoring interval. The meanings of each field (from left to right) are minutes, hour
s, days, months, and years. If all of them are `*`, it indicates to update once every minute by d
efault
schedule => "* * * * *"
type => "jdbc"
}
}

output {
elasticsearch {
hosts => ["http://172.16.0.30:9200"]
index => "test_mysql"
document_id => "%{id}"
}
}
```

For more information on connecting JDBC data sources, please see JDBC input plugin.

Using Beats to Access ES Cluster

Beats contains a variety of single-purpose collectors. These collectors are relatively lightweight and can be deployed and run on servers to collect data such as logs and monitoring information. Beats occupies less system resources than Logstash does.

Beats includes FileBeat for collecting file-type data, MetricBeat for collecting monitoring metric data, PacketBeat for collecting network packet data, etc. You can also develop your own Beats components based on the official `libbeat` library as needed.

©2013-2019 Tencent Cloud. All rights reserved.

Accessing ES cluster from CVM

1. Install and deploy Filebeat.

```
wget https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-5.6.4-linux-x86_64.tar.gz
tar xvf filebeat-5.6.4.tar.gz
```

2. Configure `filebeat.yml`.
3. Run Filebeat.

```
nohup ./filebeat 2>&1 >/dev/null &
```

Accessing ES cluster from Docker

Creating Docker cluster

1. Pull the official image of Filebeat.

```
docker pull docker.elastic.co/beats/filebeat:5.6.9
```

2. Customize the `¥*.conf` configuration file based on the data source type and place it in the `/usr/share/logstash/pipeline/` directory which can be customized.
3. Run Filebeat.

```
docker run docker.elastic.co/beats/filebeat:5.6.9
```

Using TKE

The deployment method of Filebeat through TKE is similar to that of Logstash, and you can use the Filebeat image provided by Tencent Cloud.

Configuration file description

Configure the `filebeat.yml` file as follows:

```
// Input source configuration
filebeat.prospectors:
- input_type: log
paths:
- /usr/local/services/testlogs/*.log

// Output to ES
output.elasticsearch:
# Array of hosts to connect to.
hosts: ["172.16.0.39:9200"]
```

©2013-2019 Tencent Cloud. All rights reserved.