Elasticsearch Service

Getting Started

Product Documentation

Tencent Cloud

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

Getting Started

Evaluation of Cluster Specification and Capacity Configuration

Last updated：2021-10-29 14:58:01

Tencent Cloud Elasticsearch Service (ES) is a distributed multi-node cluster, where each node consists of computing and storage components. This document provides guidance on how to select an appropriate ES configuration based on your business needs in some common use cases. These suggestions are for your reference and should be adjusted according to your actual business conditions. The auto scaling mechanism provided by ES allows you to scale out your cluster at any time to get optimal cluster specifications when performance bottlenecks occur as your business grows.

Storage Capacity Estimation

The storage capacity of ES is mainly affected by the following factors:

Number of replicas: replicas can improve data reliability but also increase storage costs. The default and recommended number of replicas is 1, and for some scenarios where the data loss caused by exceptions is tolerable, you can consider setting the number of replicas to 0.

Data expansion: in addition to raw data, ES also needs to store indices and columnar data, and the data size generally expands by 10% after technologies such as encoding and compression are applied.

Internal task overheads: ES occupies approximately **20%** of the disk space, which is used for segment merges, ES Translog, logs, etc.

OS reserve: Linux reserves 5% of the disk space for root users by default for key process handling, system restoration, and disk defragmentation.

Therefore, the actual space occupied by data in ES can be estimated with the following formula:

```
Actual space = source data * (1+ number of replicas) * (1 + data expansion) / (1 -
        ≈ source data * (1 + number of replicas) * 1.45
```

To ensure the stable operations of the service, we recommend you reserve at least 15% of the disk space. Therefore, the recommended disk capacity is:

```
Storage capacity = source data * (1 + number of replicas) * 1.45 * (1 + reserved sp
        ≈ source data * (1 + number of replicas) * 1.67
```

Generally, to ensure the performance and stability of your cluster, we recommend that the ratio of memory to disk capacity not exceed the following limit:

For hot data, the recommended ratio of memory to disk capacity is 1:96.

For warm data, the recommended ratio of memory to disk capacity is 1:480.

Note:

ES is widely used in various scenarios such as log, website search, metrics, and APM. The above storage capacity estimation method is general, and you can highly optimize ES based on your own needs to reduce your storage costs.

Computing Resource Estimation

ES computing resources are mainly consumed in the write and query processes, and the complexity of writes and queries and their proportion vary by business scenario; therefore, computing resources are more difficult to estimate than storage resources. However, storage resources generally become a bottleneck earlier, so we recommend you estimate the storage capacity first, and then preliminarily select computing resources and check whether they are sufficient in the testing process.

The following provides guidance on how to estimate computing resources in several common use cases:

Log scenario: log is a typical scenario involving more writes than reads, and computing resources are mainly consumed in the write process. Our experience in log scenarios shows that 2 CPU cores and 8 GB memory can sustain up to 5,000 writes per second, subject to the actual business scenario. As instance performance is generally scaled linearly with the total amount of computing resources, you can estimate the write capacity based on the total amount of instance resources; for example, 8 CPU cores and 32 GB memory can sustain up to 20,000 writes per second.

Structured data scenarios such as metrics and APM: these scenarios also involve more writes than reads but consume less computing resources than log scenarios; for example, 2 CPU cores and 8 GB memory generally can sustain 10,000 writes per second. You can estimate the actual write capacity of instances with different specifications by using the linear scaling method as described in the log scenario.

Search scenarios such as website search and application search: these scenarios involve more reads than writes, and computing resources are mainly consumed in the query process. As the query complexity varies significantly by use case, computing resources are the most difficult to estimate. We recommend you preliminarily select computing resources based on storage resources and then perform verification and make adjustments in the testing process.

Instance Type Selection and Testing

After estimating storage and computing resources, you can preliminarily select the instance type, including the node specification and the number of nodes. We recommend you select the instance type as follows:

We recommend you select at least 3 nodes to prevent split-brain problems in your ES instance and ensure its high node fault tolerance.

Note:

Split-Brain: a problem where both nodes believe that they are the only active server and thus scramble for resources.

If you need a high storage capacity, we recommend you select high-specced nodes rather than high numbers of low-specced nodes, as this benefits the performance and stability of large instances significantly. For example, if you need 40 CPU cores, 160 GB memory, and 5 TB storage capacity, we recommend you select a 5-node instance with 8 CPU cores, 32 GB memory, and 1 TB storage capacity each. Likewise, when you need to scale out your instance, we recommend you perform vertical scaling first to expand the node capacity to 8 CPU cores and 32 GB memory or 16 CPU cores and 64 GB memory and then consider horizontal scaling to increase the number of nodes.

After you preliminarily select the instance type, you can test it by using real data and observe monitoring information such as CPU utilization, write metrics (performance and rejection rate), and query metrics (QPS and rejection rate) to further verify whether the instance type is appropriate. In addition, we recommend you configure alarms for the above monitoring information, so that you can promptly identify issues such as resource insufficiency.

Estimation of Number of Shards

Each ES index is split into multiple shards, and data is distributed among different shards according to the hash algorithm. As the number of an index's shards affects the read/write performance and failure recovery speed and usually cannot be easily changed, it needs to be considered in advance. We recommend you configure the number of shards as follows:

The recommended size of a single shard is 10–50 GB, and you can preliminarily determine the number of an index's shards based on this value. Shards should be neither too large nor too small. If they are too large, ES failure recovery may be slow; if they are too small, there may be many shards, and each shard uses certain amounts of CPU and memory resources, which may cause various issues, including poor read/write performance and memory insufficiency.

During testing, you can appropriately adjust the number of shards based on the actual size of each index and expected future growth.

When the number of shards exceeds the number of data nodes, we recommend you adjust the former to make it close to an integer multiple of the latter, so that the shards can be evenly distributed among all the data nodes.

For log and metric scenarios, we recommend you use the rollover index feature of ES to generate new indices on a rolling basis, so that you can promptly adjust the number of shards when you find that the shard size is unreasonable.

For example, if an instance has 5 data nodes and the current index size is 150 GB and is expected to grow by 50% in half a year, and you set the size of each shard to 30 GB, then you need 150 GB * (1 + 50%) / 30 ≈ 7 shards.

Considering that two of the data nodes sustain 2/7 of the data and the data is unevenly distributed among the nodes, you can adjust the number of shards to 10.

Creating Clusters

Last updated：2025-01-24 10:35:12

A cluster is the basic unit where ES provides hosted Elasticsearch services and you use and manage such services. This document describes how to quickly create an Elasticsearch cluster in the Tencent Cloud Console.

Prerequisites

You have a Tencent Cloud account. For more information on how to create an account, please see Signing up for a Tencent Cloud Account.

Directions

Logging in to console

Log in to the ES Console and click **Create** to enter the creation and purchase page.

Creating cluster

1. Select cluster configuration

Billing Mode: **pay-as-you-go billing** is supported.
Region: currently, ES has been launched in multiple regions in and outside Mainland China.
Elasticsearch Version: v5.6.4, v6.4.3, v6.8.2, and v7.5.1 are supported.
X-Pack: Elasticsearch's official commercial features, including capabilities such as data permission management, SQL JDBC, alerting, and machine learning. The features available vary by edition: the Platinum Edition has all the advanced features, the Basic Edition has some advanced features, and the Open Source Edition does not have advanced features. For more information, please see X-Pack.

Note:

The Basic Edition supports ES cluster user authentication starting from v6.8, which greatly improves the cluster security and is therefore recommended to be enabled. If you do not enable it when purchasing your cluster, you need to fully restart the cluster when enabling it subsequently, which will affect your online businesses. This feature is enabled in the Platinum Edition by default and is not supported in the Open Source Edition.
Network/AZ and Subnet: ES is deployed in VPCs. An ES cluster can only be accessed from servers in the same VPC. Therefore, to ensure smooth access to the ES cluster over the private network, you are recommended to select a VPC in the region where your existing cloud-based businesses reside. In multi-AZ deployment mode, you should also select the same VPC. Subnets in different AZs in the same VPC can communicate with each other.

Note:

The VPC cannot be changed or adjusted once the ES cluster is created.

Special notes on network selection

Cross-VPC Access: if you need to access an ES cluster from a CVM instance located in another VPC in the same region, consider the peering connection scheme which can interconnect two different VPCs in the same region.

Basic Network Access: if your business is deployed in the basic network and you have never used a VPC, you can bind the CVM instance located in the basic network to the VPC where the ES cluster resides through Classiclink. For more information, please see Classiclink. Classiclink only supports VPCs in the IP range of `10.[0-47].0.0/16`. If you need to access the ES cluster from the basic network, please select a VPC in this range when creating the cluster.

AZ Deployment Mode: single-AZ is a mode where the ES cluster is deployed in one AZ, while multi-AZ is a mode where the cluster is deployed in two AZs in the same region. The multi-AZ mode can improve the disaster recovery capability of the cluster and ensure the stability of online businesses. For more information, please see Multi-AZ Cluster Deployment.

Data Node Deployment Mode: single mode or hot/warm mode. In hot/warm mode, a cluster can have both warm and hot nodes where hot data that has high requirements for read/write performance can be stored on hot nodes, while indices that have high requirements for storage capacity and relatively low requirements for read/write performance can be stored on warm nodes. This ensures read/write performance of the hot data while reducing storage costs. For more information, please see Hot/Warm Architecture and Index Lifecycle Management.

Node Model: specifications of each node model in the cluster. The core quantity and memory size available vary by model. For more information on the node models supported by ES and how to select an appropriate type, please see Node Type and Storage Configuration.

Node Storage Type: premium or SSD cloud disk.

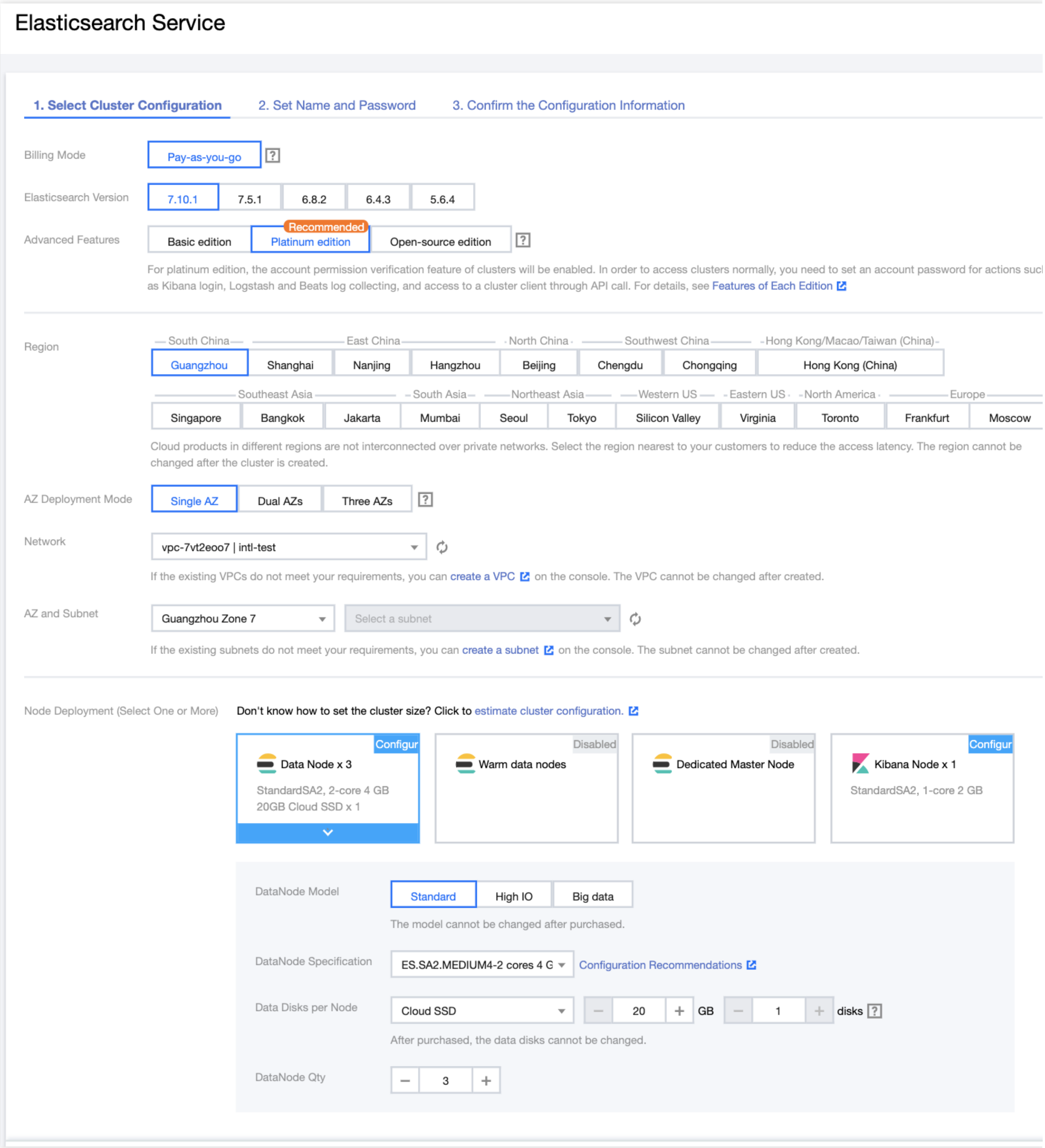
Single-Node Storage: disk capacity configured for each node. The storage capacity of the entire cluster is the single-node storage multiplied by the number of nodes.

Dedicated Master Node: if the cluster is large, you can configure dedicated master nodes to further ensure the cluster stability.

Dedicated Master Node Model: a dedicated master node can use a different model from that of the data nodes.

Number of Dedicated Master Nodes: 3 or 5. An odd number of nodes ensures high availability and prevents the risk of split-brain.

Kibana Node: it is provided free of charge with the 1-core 2 GB MEM specification. Currently, it cannot be modified.



Elasticsearch Service

1. Select Cluster Configuration | 2. Set Name and Password | 3. Confirm the Configuration Information

Billing Mode: Pay-as-you-go

Elasticsearch Version: 7.10.1 | 7.5.1 | 6.8.2 | 6.4.3 | 5.6.4

Advanced Features: Basic edition | Platinum edition (Recommended) | Open-source edition

For platinum edition, the account permission verification feature of clusters will be enabled. In order to access clusters normally, you need to set an account password for actions suc as Kibana login, Logstash and Beats log collecting, and access to a cluster client through API call. For details, see Features of Each Edition

Region:
South China: Guangzhou | East China: Shanghai, Nanjing, Hangzhou | North China: Beijing | Southwest China: Chengdu, Chongqing | Hong Kong/Macao/Taiwan (China): Hong Kong (China)
Southeast Asia: Singapore, Bangkok, Jakarta | South Asia: Mumbai | Northeast Asia: Seoul, Tokyo | Western US: Silicon Valley | Eastern US: Virginia | North America: Toronto | Europe: Frankfurt, Moscow

Cloud products in different regions are not interconnected over private networks. Select the region nearest to your customers to reduce the access latency. The region cannot be changed after the cluster is created.

AZ Deployment Mode: Single AZ | Dual AZs | Three AZs

Network: vpc-7vt2eoo7 | intl-test

If the existing VPCs do not meet your requirements, you can create a VPC on the console. The VPC cannot be changed after created.

AZ and Subnet: Guangzhou Zone 7 | Select a subnet

If the existing subnets do not meet your requirements, you can create a subnet on the console. The subnet cannot be changed after created.

Node Deployment (Select One or More): Don't know how to set the cluster size? Click to estimate cluster configuration.

- Data Node x 3 (Configur): StandardSA2, 2-core 4 GB, 20GB Cloud SSD x 1
- Warm data nodes (Disabled)
- Dedicated Master Node (Disabled)
- Kibana Node x 1 (Configur): StandardSA2, 1-core 2 GB

DataNode Model: Standard | High IO | Big data

The model cannot be changed after purchased.

DataNode Specification: ES.SA2.MEDIUM4-2 cores 4 G | Configuration Recommendations

Data Disks per Node: Cloud SSD, 20 GB, 1 disks

After purchased, the data disks cannot be changed.

DataNode Qty: 3

2. Set the name and password

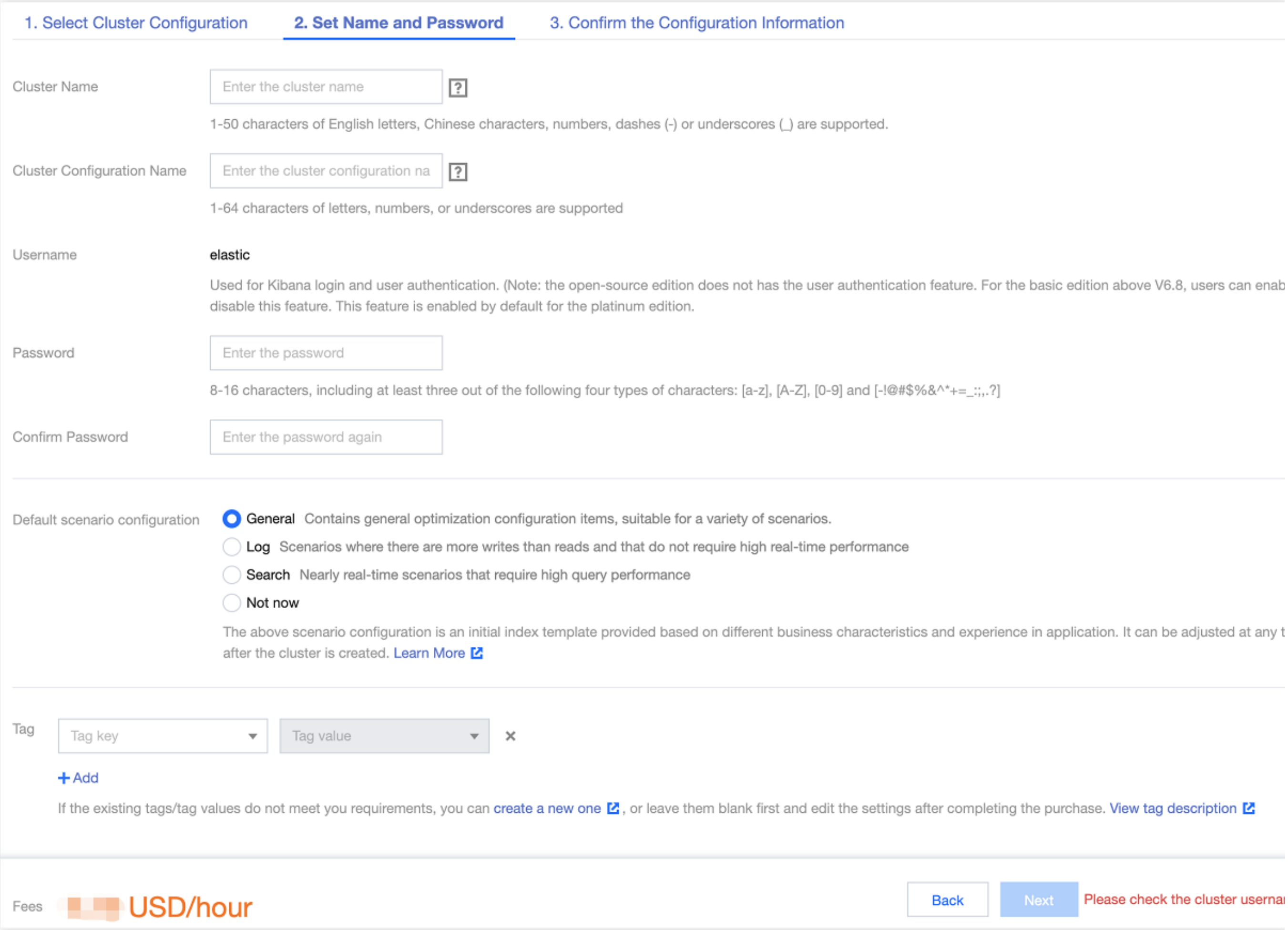
Click **Next** and set a name and password.

Cluster Name: name the cluster as desired. This name is not a globally unique identifier and can be set as a business-related description.

Username: username used for Kibana page login and ES cluster user authentication. It is `elastic` by default and cannot be modified.

Password: password corresponding to the aforementioned username. Please set it as required and keep it private. If you forgot it, you can reset it on the details page.

Tag: tag is a unified management service provided by Tencent Cloud and can be used to manage existing Tencent Cloud resources in a categorized manner. You can create or modify tags either before or after ES cluster purchase. For more information, please see Tag.



1. Select Cluster Configuration
2. Set Name and Password
3. Confirm the Configuration Information

Cluster Name: Enter the cluster name

1-50 characters of English letters, Chinese characters, numbers, dashes (-) or underscores (_) are supported.

Cluster Configuration Name: Enter the cluster configuration na

1-64 characters of letters, numbers, or underscores are supported

Username: elastic

Used for Kibana login and user authentication. (Note: the open-source edition does not has the user authentication feature. For the basic edition above V6.8, users can enab disable this feature. This feature is enabled by default for the platinum edition.

Password: Enter the password

8-16 characters, including at least three out of the following four types of characters: [a-z], [A-Z], [0-9] and [-!@#$%&^*+=_:;,.?]

Confirm Password: Enter the password again

Default scenario configuration:
● General Contains general optimization configuration items, suitable for a variety of scenarios.
○ Log Scenarios where there are more writes than reads and that do not require high real-time performance
○ Search Nearly real-time scenarios that require high query performance
○ Not now

The above scenario configuration is an initial index template provided based on different business characteristics and experience in application. It can be adjusted at any t after the cluster is created. Learn More

Tag: Tag key ▾ Tag value ▾ ×

+Add

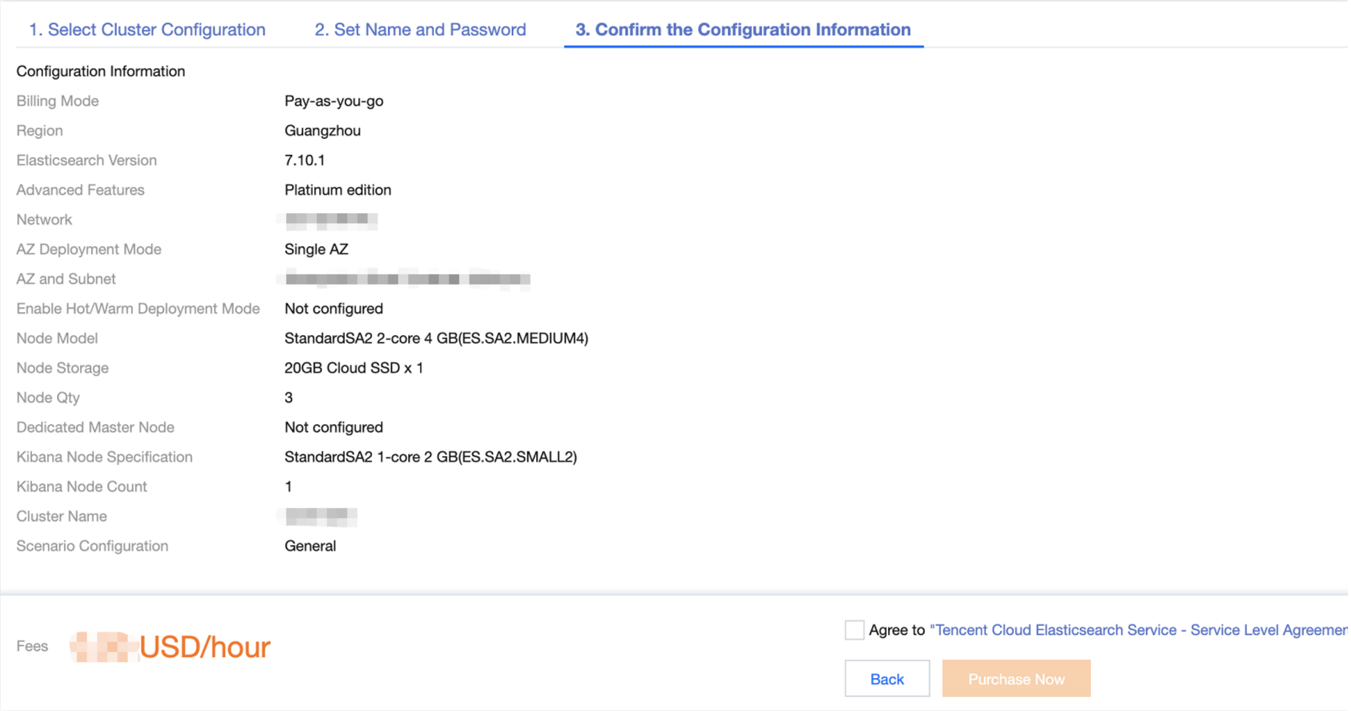
If the existing tags/tag values do not meet you requirements, you can create a new one , or leave them blank first and edit the settings after completing the purchase. View tag description

Fees USD/hour

Please check the cluster userna

3. Confirm configuration information

Click **Next** and confirm configuration information.



1. Select Cluster Configuration
2. Set Name and Password
3. Confirm the Configuration Information

Configuration Information

Billing Mode	Pay-as-you-go
Region	Guangzhou
Elasticsearch Version	7.10.1
Advanced Features	Platinum edition
Network	[redacted]
AZ Deployment Mode	Single AZ
AZ and Subnet	[redacted]
Enable Hot/Warm Deployment Mode	Not configured
Node Model	StandardSA2 2-core 4 GB(ES.SA2.MEDIUM4)
Node Storage	20GB Cloud SSD x 1
Node Qty	3
Dedicated Master Node	Not configured
Kibana Node Specification	StandardSA2 1-core 2 GB(ES.SA2.SMALL2)
Kibana Node Count	1
Cluster Name	[redacted]
Scenario Configuration	General

Fees [redacted] USD/hour

□ Agree to "Tencent Cloud Elasticsearch Service - Service Level Agreemer

Click **Activate** to create the cluster directly if the pay-as-you-go billing mode is selected. You do not need to confirm the order or pay for it, as your account balance will be deducted on an hourly basis during cluster use.

4. Complete the creation

Once successful activated, the pay-as-you-go cluster just created can be viewed in the console and will be completely created in a matter of minutes.

Subsequent Steps

Accessing cluster

To help you get started quickly, ES provides several types of clients for accessing clusters. For more information, please see Accessing Clusters Through API, Accessing Clusters Through Client, and Accessing Clusters Through Kibana.

Monitoring cluster

ES provides a rich set of monitoring metrics to help you view the status of clusters during use. For more information, please see Viewing Monitoring Metrics.

In the Basic Edition and Platinum Edition, Kibana also offers monitoring metrics on the **Monitor** page on the left sidebar.

Adjusting cluster configuration

With the increase in volumes of business data and access requests, the cluster configuration can be elastically adjusted. For more information, please see Adjusting Configuration.

Accessing Clusters

Login User Verification

Last updated：2021-10-29 15:08:07

Before learning how to access an ES cluster, you need to understand user authentication in ES clusters first.

ES Cluster User Authentication

This feature is used to improve the data access security of ES clusters (for more information, please see Protect your data in the Elastic Stack). You must pass the authentication based on username and password before you can access an ES cluster through Kibana, clients, or APIs. For more information, please see Accessing Cluster from API, Accessing Cluster from Client, and Accessing Cluster from Kibana.

Note:

You are required to set the username and password when creating an ES cluster.

For clusters with this feature not enabled, the username and password will be used only for Kibana login.

For clusters with this feature enabled, the username and password can be used for authentication for ES cluster login in any method.

Elasticsearch Editions Supporting ES Cluster User Authentication

Not all Elasticsearch editions support this feature. The support conditions of all editions are as follows:

Open Source Edition: it does not support this feature.

Basic Edition: only v6.8 or above allows you to enable or disable this feature, **while legacy versions do not support this feature**.

Platinum Edition: this feature is enabled by default and cannot be disabled.

Notes on Enabling ES Cluster User Authentication

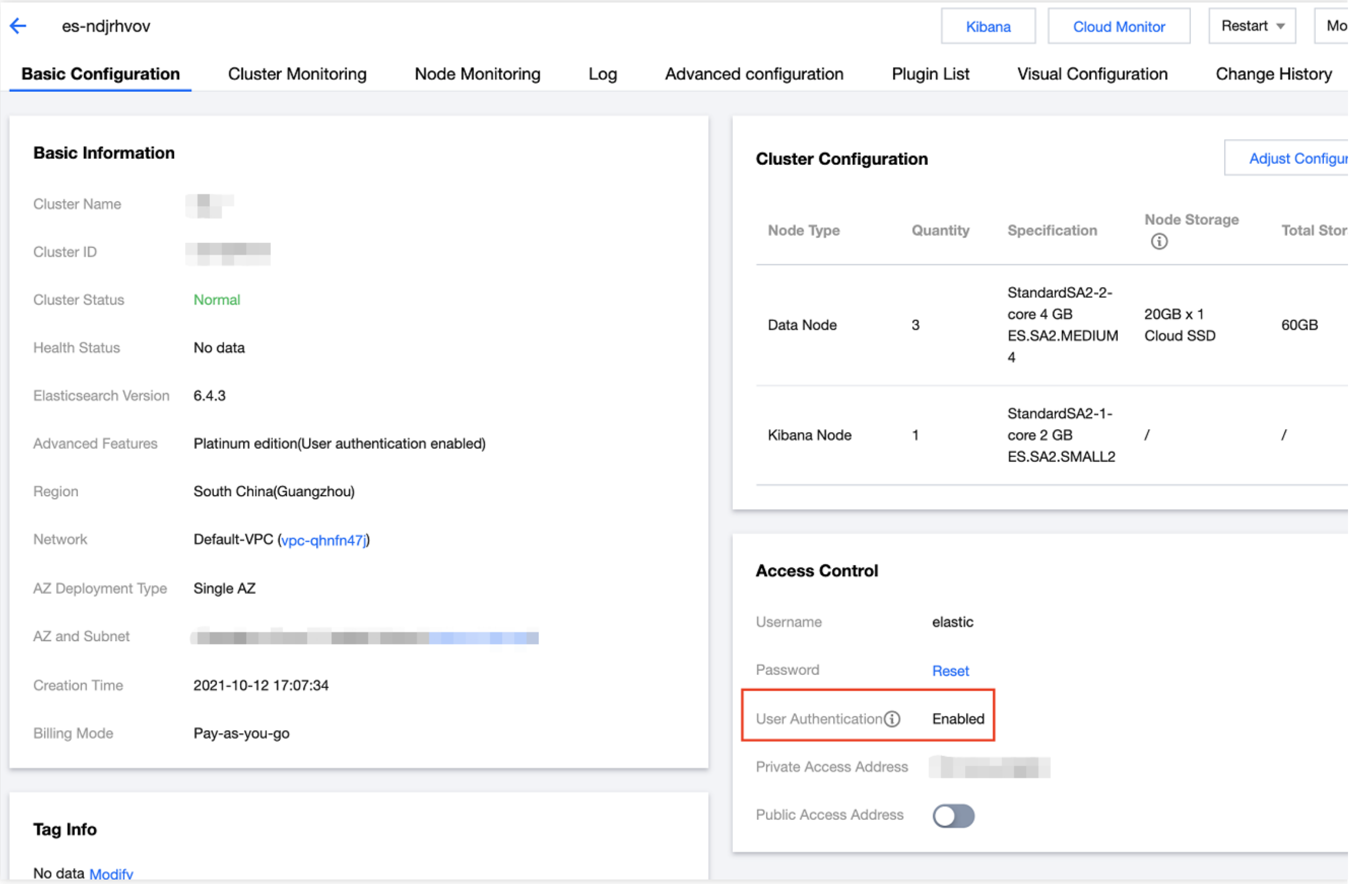
If you did not enable user authentication for an ES cluster previously and now want to enable it by upgrading the cluster or toggling the configuration switch, you need to complete the following steps first:

Modify your business code in advance, so that the username and password can be passed in when relevant APIs are called to normally access the cluster after this feature is enabled.

According to Elasticsearch's official design requirements, you need to fully restart the cluster after enabling this feature. During the restart, the cluster will be unavailable; therefore, please do so at an appropriate time.

Enabling/Disabling ES Cluster User Authentication on Basic Edition v6.8 or Above

When creating a cluster, you can choose whether to enable or disable ES cluster user authentication.
After a cluster is created, if you need to change the feature status, you can enter the cluster details page for configuration.



Basic Information

Cluster Name	[redacted]
Cluster ID	[redacted]
Cluster Status	Normal
Health Status	No data
Elasticsearch Version	6.4.3
Advanced Features	Platinum edition(User authentication enabled)
Region	South China(Guangzhou)
Network	Default-VPC (vpc-qhnfn47j)
AZ Deployment Type	Single AZ
AZ and Subnet	[redacted]
Creation Time	2021-10-12 17:07:34
Billing Mode	Pay-as-you-go

Tag Info

No data Modify

Cluster Configuration

Node Type	Quantity	Specification	Node Storage ⓘ	Total Stor
Data Node	3	StandardSA2-2-core 4 GB ES.SA2.MEDIUM4	20GB x 1 Cloud SSD	60GB
Kibana Node	1	StandardSA2-1-core 2 GB ES.SA2.SMALL2	/	/

Access Control

Username	elastic
Password	Reset
User Authentication ⓘ	Enabled
Private Access Address	[redacted]
Public Access Address	○

Accessing Clusters from Client

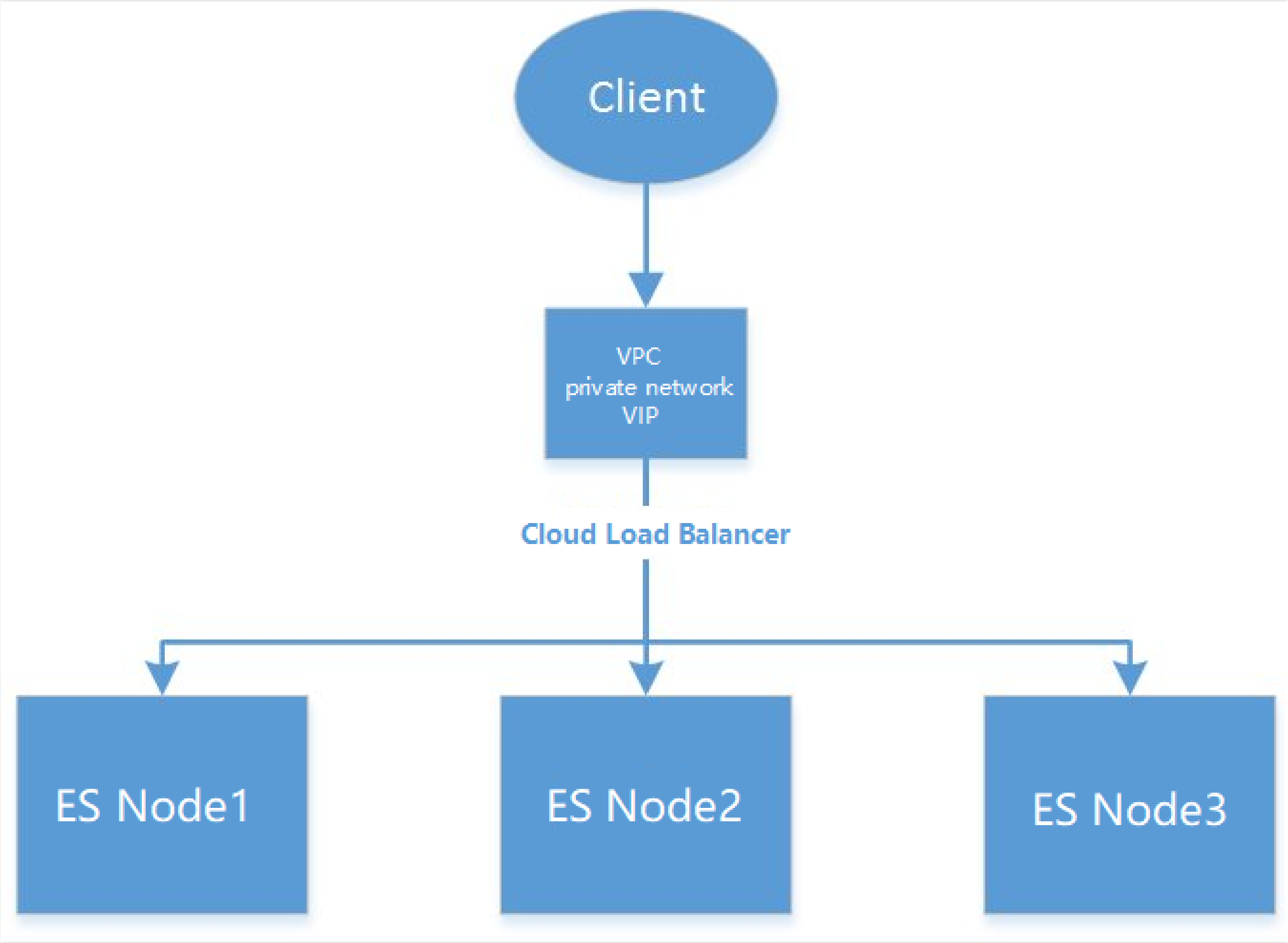
Last updated：2020-08-21 17:33:53

ES offers SDKs for different programming languages through its official website and community.
Elasticsearch provides clients in a variety of programming languages, such as Java and Python, to meet the needs of different developers. For more information, please see Elasticsearch Clients.
Starting from Elasticsearch 5.6.0, a new official Java client has been released: the Java High Level REST Client. This client can be used to perform search, index, delete, update, and bulk operations using the same core Java classes as the Transport Client does. It is actually designed to replace the Transport Client. For more information, please see Java High Level REST Client.

Note:

In terms of version compatibility, you are recommended to choose the client version that is compatible with the server version. For more information, please see Compatibility. Currently, ES is available in multiple Elasticsearch versions, so be sure to select a compatible client version.

ES provides a VIP for accessing your ES cluster in your VPC, to which all nodes in your ES cluster are mounted through load balancing. This aims to adapt to the auto scaling of your cluster and ensure its high availability. In case of any changes in the nodes, the VIP will automatically update node information. In addition, this simplifies operations and eliminates your need to focus on changes in the information about cluster nodes such as IP and port.



Note:

You are not recommended to connect a client to ES nodes directly through Transport Client.

The SDKs of ES allow you configure the address of only one node when creating a connection and use the "node sniffing" feature to sniff out all nodes for connection. However, this goes against our original intention of launching the VIP feature and increases the complexity of using an ES cluster. The following describes how to use the SDKs for different programming languages and how to turn off node sniffing.

Java Client

ES recommends that you connect to your cluster and manipulate data using the Java REST Client, which comes in two types: Low Level and High Level:

Java Low Level REST Client: when using this client, you need to manually splice the body of an HTTP request into JSON format and encapsulate the returned JSON data in the HTTP response into an object.

Java High Level REST Client: this client is implemented based on the Low Level Client and provides APIs to eliminate the need to manually convert data format.

The sample steps and code for accessing a cluster using the Java High Level REST Client are as follows:

Adding Maven dependencies

Note:

You are recommended to choose the Java REST Client API version that is compatible with the server edition of your ES cluster. Currently, ES is available in multiple versions, so be sure to select a compatible client version. For more information on client APIs, please see Elasticsearch API Version Compatibility.

```
<dependency>
       <groupId>org.elasticsearch.client</groupId>
       <artifactId>elasticsearch-rest-high-level-client</artifactId>
       <version>6.4.3</version>
</dependency>
```

Note:

The client version should be compatible with the ES cluster version; otherwise, a compatibility issue may occur. The demo here is applicable to ES 6.4.3. For more information on how to use other versions, please see Java High Level REST Client.

The Java High REST Client is built on the Java Low REST Client, and both of them connect to an ES cluster using the HTTP protocol. The Java High Level REST provides more APIs with version upgrading. If the APIs currently provided by it cannot meet your needs, you can upgrade your ES cluster version and client version.

The Transport Client that uses the TCP protocol to connect to an ES cluster is no longer updated, so you are recommended to use the Java High Level or Low Level Client that uses the HTTP protocol. For more information, please see Migrating from Transport Client to Java High REST Client.

Sample code

```
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.log4j.BasicConfigurator;
import org.apache.log4j.Logger;
import org.elasticsearch.ElasticsearchException;
import org.elasticsearch.action.DocWriteResponse;
import org.elasticsearch.action.index.IndexRequest;
import org.elasticsearch.action.index.IndexResponse;
import org.elasticsearch.action.get.GetRequest;
import org.elasticsearch.action.get.GetResponse;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.rest.RestStatus;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;


public class test_es_sdk {
    private static Logger logger = Logger.getLogger(test_es_sdk.class);

    public  static void main(String[]args){
        BasicConfigurator.configure();
        // Set verification information by entering the username and password
        final CredentialsProvider credentialsProvider = new
BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY,
                new UsernamePasswordCredentials("user", "passwd"));
        // Initialize RestClient. For hostName and port, enter the private VIP
address and port of the cluster respectively
        RestClientBuilder builder = RestClient.builder(new
HttpHost("xx.xx.xx.xx", 9200, "http"));
        // Set authentication information
        builder.setHttpClientConfigCallback(new
RestClientBuilder.HttpClientConfigCallback() {
                    @Override
                    public HttpAsyncClientBuilder
customizeHttpClient(HttpAsyncClientBuilder httpClientBuilder) {
                        return
httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
                    }
                });
        // Set the timeout period
        builder.setMaxRetryTimeoutMillis(10000);
        // Construct the High Level Client based on the Low Level Client
        RestHighLevelClient client = new RestHighLevelClient(builder);

        // Index the document
        Map<String, Object> jsonMap = new HashMap<String, Object>();
        jsonMap.put("user", "bellen");
        jsonMap.put("name", new Date());
        jsonMap.put("message", "trying out Elasticsearch");
        IndexRequest indexRequest = new IndexRequest("posts", "doc", "1")
                .source(jsonMap);

        try {
            // Get the response result
            IndexResponse indexResponse = client.index(indexRequest);
            String index = indexResponse.getIndex();
            String type = indexResponse.getType();
            String id = indexResponse.getId();
            long version = indexResponse.getVersion();

            if (indexResponse.getResult() == DocWriteResponse.Result.CREATED) {
                logger.info("doc indexed, index: "+ index +", type:"+ type
+",id:"+ id+",version:"+version);
            } else if (indexResponse.getResult() ==
DocWriteResponse.Result.UPDATED) {
                logger.info("doc updated, index: "+ index +", type:"+ type
+",id:"+ id+",version:"+version);
            }
        }catch(ElasticsearchException e) {
            if (e.status() == RestStatus.CONFLICT) {
                logger.error("version conflict");

            }
        }catch(Exception e){
            logger.error("execute index api failed, "+ e.toString());
        }



        // Query the document
        GetRequest getRequest = new GetRequest(
                "posts",
                "doc",
                "1");
        try {
            // Get the response result
            GetResponse getResponse = client.get(getRequest);
            String index = getResponse.getIndex();
            String type = getResponse.getType();
            String id = getResponse.getId();
            if (getResponse.isExists()) {
                long version = getResponse.getVersion();
                String sourceAsString = getResponse.getSourceAsString();
                logger.info("get doc, index: "+ index +", type:"+ type +",id:"+
id+",version:"+version +", source:"+ sourceAsString);
            }
        }catch (ElasticsearchException e) {
            if (e.status() == RestStatus.NOT_FOUND) {
                logger.warn("doc not found");
            }
        }
        catch(Exception e){
            logger.error("execute get api failed, "+ e.toString());
        }
        // Close the client
        try {
            client.close();
        }catch (Exception e){
            logger.error("close rest client exception:"+ e.toString());
        }
    }
}
```

Python Client

Installing through pip

```
pip install elasticsearch
```

Sample code

Set the following three parameters of the `Elasticsearch` function to turn off node sniffing:

`sniff_on_start=False`

`sniff_on_connection_fail=False`

`sniffer_timeout=None`

```
from elasticsearch import Elasticsearch

es = Elasticsearch(["http://xx.xx.xx.xx:9200"],
      http_auth=('user', 'passwd'),
          sniff_on_start=False,
          sniff_on_connection_fail=False,
          sniffer_timeout=None)

res = es.index(index="my_index", doc_type="my_type", id=1, body={"title":
"One", "tags": ["ruby"]})
print(res)
res = es.get(index="my_index", doc_type="my_type", id=1)
print(res['_source'])
```

PHP Client

To avoid sniffing of nodes, you can set the connection pool class as `StaticConnectionPool`. **Do not use the node sniffing connection pool**.

Sample code

```
$client = ClientBuilder::create()
    -
>setConnectionPool('\\Elasticsearch\\ConnectionPool\\StaticConnectionPool',
["http://user:passwd@xx.xx.xx.xx:9200"])
    ->build();
```

Go Client

`gopkg.in/olivere/elastic` is a community-contributed SDK that is widely used in the Go language. The demo here is applicable to ES 6.4.3. For information on how to use other versions, please see here.

Installing elastic

```
go get github.com/olivere/elastic
```

Sample code

In the parameters of the `NewClient` function, set `elastic.SetSniff(false)` to turn off node sniffing and set `elastic.SetHealthcheck(false)` to turn off node health check.

```
import (
    "context"
    "fmt"

    "github.com/olivere/elastic"
)

func main() {
    client, err :=
elastic.NewClient(elastic.SetURL("http://user:passwd@xx.xx.xx.xx:9200"),
        elastic.SetSniff(false),elastic.SetHealthcheck(false))
    if err != nil {
        panic(err)
    }
    exists, err := client.IndexExists("twitter").Do(context.Background())
    if err != nil {
        panic(err)
    }
    fmt.Println(exists)
}
```

Accessing Cluster from API

Last updated：2021-07-01 10:02:56

Elasticsearch provides full-featured RESTful APIs for intercalation with clusters. For more information, please see Elasticsearch's official API documentation.

As Tencent Cloud ES is deployed in your VPC, you can use a CVM instance in the same VPC as the client to access the ES cluster over the **private network** or **public network**. **Public network access has security risks**; therefore, please enable it with caution.

Note:

Public network access is used for development and debugging only but cannot be used in the production environment, as the system limits the call frequency.

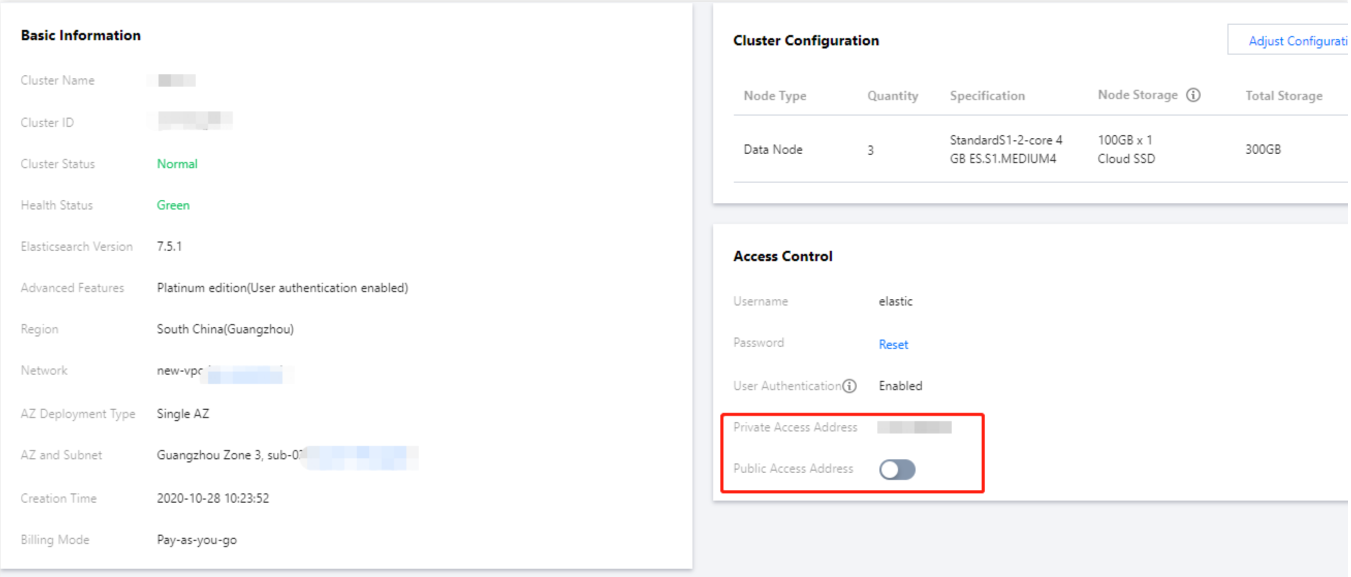
Currently, public network access to ES is free of charge with a bandwidth of 10 Mbps.

Viewing Private/Public Network Access Addresses

On the cluster list page, click a **cluster ID** to enter the cluster details page:

The private address can be directly found in basic configuration.

The public network address is disabled by default for the sake of security. For clusters having ES cluster user authentication enabled, the public network address can be enabled. Doing so may bring security risks to the clusters, as it allows data in ES clusters to be accessed, manipulated, and even deleted directly through APIs; therefore, please enable it with caution.



Basic Information

Cluster Name	[redacted]
Cluster ID	[redacted]
Cluster Status	Normal
Health Status	Green
Elasticsearch Version	7.5.1
Advanced Features	Platinum edition(User authentication enabled)
Region	South China(Guangzhou)
Network	new-vpc [redacted]
AZ Deployment Type	Single AZ
AZ and Subnet	Guangzhou Zone 3, sub-0 [redacted]
Creation Time	2020-10-28 10:23:52
Billing Mode	Pay-as-you-go

Cluster Configuration

Node Type	Quantity	Specification	Node Storage	Total Storage
Data Node	3	StandardS1-2-core 4 GB ES.S1.MEDIUM4	100GB x 1 Cloud SSD	300GB

Access Control

Username	elastic
Password	Reset
User Authentication	Enabled
Private Access Address	[redacted]
Public Access Address	○

Testing Access

You can test access to clusters by running the `curl` command. The `ping` command is not supported for connectivity test.

Testing service accessibility

Note:

For clusters having ES cluster user authentication enabled, login requires authentication of username and password in the format of `curl action -u user:password host ...` , where `user` , `password` , and `host` should be replaced with the actual username, password, and IP.

The following uses a private network access address as an example to describe the access operations.

Enter the following command:

```
curl -XGET http://10.0.17.2:9200
If ES cluster user authentication is enabled, remember to enter the username
and password.
curl -XGET -u user:password http://10.0.17.2:9200
```

The following content is returned, indicating that the cluster can be accessed normally. The specific parameter values vary by cluster version:

```
{
  "name": "15589826570000*****",
  "cluster_name": "es-******",
  "cluster_uuid": "NGIm1M_zRw-L3o_gH****",
  "version": {
    "number": "6.4.3",
    "build_flavor": "default",
    "build_type": "zip",
    "build_hash": "fe40335",
    "build_date": "2019-05-17T14:22:47.286024Z",
    "build_snapshot": false,
    "lucene_version": "7.4.0",
    "minimum_wire_compatibility_version": "5.6.0",
    "minimum_index_compatibility_version": "5.0.0"
  },
  "tagline": "You Know, for Search"
}
```

Creating Document

Creating one document

If user authentication is not enabled for the cluster, enter the following command:

```
curl -XPUT http://10.0.0.2:9200/china/city/beijing -H 'Content-Type:
application/json' -d'
{
"name":"Beijing",
"province":"Beijing",
"lat":39.9031324643,
"lon":116.4010433787,
"x":6763,
"level.range":4,
"level.level":1,
"level.name":"Tier-1 city",
"y":6381,
"cityNo":1
}
'
```

If user authentication is enabled for the cluster, you need to **replace the `user` and `password` below with your actual cluster username and password.** Enter the following command:

```
curl -XPUT -u user:password http://10.0.0.2:9200/china/city/beijing -H
'Content-Type: application/json' -d'
{
"name":"Beijing",
"province":"Beijing",
"lat":39.9031324643,
"lon":116.4010433787,
"x":6763,
"level.range":4,
"level.level":1,
"level.name":"Tier-1 city",
"y":6381,
"cityNo":1
}
'
```

The following response will be returned:

```
{
  "_index":"china",
  "_type":"city",
  "_id":"beijing",
  "_version":1,
  "result":"created",
  "_shards":{
      "total":2,
      "successful":1,
      "failed":0
  },
  "created":true
}
```

Creating multiple documents

Enter the following command:

```
curl -XPOST http://10.0.0.2:9200/_bulk -H 'Content-Type: application/json' -d'
{ "index" : { "_index": "china", "_type" : "city", "_id" : "beijing" } }
{"name":"Beijing","province":"Beijing","lat":39.9031324643,"lon":116.4010433787
,"x":6763,"level.range":4,"level.level":1,"level.name":"Tier-1
city","y":6381,"cityNo":1}
{ "index" : { "_index": "china", "_type" : "city", "_id" : "shanghai" } }
{"name":"Shanghai","province":"Shanghai","lat":31.2319526784,"lon":121.46944324
9,"x":7779,"level.range":4,"level.level":1,"level.name":"Tier-1
city","y":4409,"cityNo":2}
{ "index" : { "_index": "china", "_type" : "city", "_id" : "guangzhou" } }
{"name":"Guangzhou","province":"No.79, Jixiang Road, Yuexiu District, Guangdong
Province","lat":23.1317146641,"lon":113.2595185241,"x":6173,"level.range":4,"le
vel.level":1,"level.name":"Tier-1 city","y":2560,"cityNo":3}
{ "index" : { "_index": "china", "_type" : "city", "_id" : "shenzhen" } }
{"name":"Shenzhen","province":"No.37, Xinyuan Road, Futian District, Guangdong
Province","lat":22.5455465546,"lon":114.0527779134,"x":6336,"level.range":4,"le
vel.level":1,"level.name":"Tier-1 city","y":2429,"cityNo":4}
{ "index" : { "_index": "china", "_type" : "city", "_id" : "chengdu" } }
{"name":"Chengdu","province":"No. 88-1, Hongxing Road 4th Section, Jinjiang
District, Sichuan
Province","lat":30.6522796787,"lon":104.0725574128,"x":4387,"level.level":2,"le
vel.range":19,"level.name":"New tier-1 city","y":4304,"cityNo":5}
{ "index" : { "_index": "china", "_type" : "city", "_id" : "hangzhou" } }
{"name":"Hangzhou","province":"No.316, Huancheng North Road, Gongshu District,
Zhejiang
Province","lat":30.2753694112,"lon":120.1509063337,"x":7530,"level.level":2,"le
vel.range":19,"level.name":"New tier-1 city","y":4182,"cityNo":6}
'
```

The following response will be returned:

```
"took":9,"errors":false,"items":[{"index":
{"_index":"china","_type":"city","_id":"beijing","_version":4,"result":"updated
","_shards":
{"total":2,"successful":2,"failed":0},"created":false,"status":200}},{"index":
```

```
{"_index":"china","_type":"city","_id":"shanghai","_version":2,"result":"update
d","_shards":
{"total":2,"successful":2,"failed":0},"created":false,"status":200}},{"index":
{"_index":"china","_type":"city","_id":"guangzhou","_version":1,"result":"creat
ed","_shards":
{"total":2,"successful":2,"failed":0},"created":true,"status":201}},{"index":
{"_index":"china","_type":"city","_id":"shenzhen","_version":1,"result":"create
d","_shards":
{"total":2,"successful":2,"failed":0},"created":true,"status":201}},{"index":
{"_index":"china","_type":"city","_id":"chengdu","_version":2,"result":"updated
","_shards":
{"total":2,"successful":2,"failed":0},"created":false,"status":200}},{"index":
{"_index":"china","_type":"city","_id":"hangzhou","_version":2,"result":"update
d","_shards":
{"total":2,"successful":2,"failed":0},"created":false,"status":200}}]
```

Updating Document

You can run the command for creating a single document again to update the document whose ID is `beijing` .

The following response will be returned:

```
{"_index":"china","_type":"city","_id":"beijing","_version":2,"result":"updated
","_shards":{"total":2,"successful":2,"failed":0},"created":false}
```

Querying Document

Querying specified ID

Enter the following command:

```
curl -XGET 'http://10.0.0.2:9200/china/city/beijing?pretty' -H 'Content-Type:
application/json'
```

The following response will be returned:

```
{
  "_index" : "china",
  "_type" : "city",
  "_id" : "beijing",
  "_version" : 4,
  "found" : true,
  "_source" : {
    "name" : "Beijing",
    "province" : "Beijing",
    "lat" : 39.9031324643,
    "lon" : 116.4010433787,
    "x" : 6763,
    "level.range" : 4,
    "level.level" : 1,
    "level.name" : "Tier-1 city",
    "y" : 6381,
    "cityNo" : 1
  }
}
```

Querying index

Enter the following command:

```
curl -XGET 'http://10.0.0.2:9200/china/city/_search?pretty' -H 'Content-Type:
application/json'
```

The following response will be returned:

```
{
  "took" : 0,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 6,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "china",
        "_type" : "city",
        "_id" : "guangzhou",
        "_score" : 1.0,
        "_source" : {
          "name" : "Guangzhou",
          "province" : "No.79, Jixiang Road, Yuexiu District, Guangdong
Province",
          "lat" : 23.1317146641,
          "lon" : 113.2595185241,
          "x" : 6173,
          "level.range" : 4,
          "level.level" : 1,
          "level.name" : "Tier-1 city",
          "y" : 2560,
          "cityNo" : 3
        }
      }]
    },
    ......
}
```

Complex query

Sample SQL statement:

```
select * from city where level.level=2
curl -XGET http://10.0.0.2:9200/china/city/_search?pretty -H 'Content-Type:
application/json' -d'
{
    "query" : {
        "constant_score" : {
            "filter" : {
                "term" : {
                    "level.level" : 2
                }
            }
        }
    }
}'
```

The following response will be returned:

```
{
  "took" : 2,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 2,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "china",
        "_type" : "city",
        "_id" : "chengdu",
        "_score" : 1.0,
        "_source" : {
          "name" : "Chengdu",
          "province" : "No. 88-1, Hongxing Road 4th Section, Jinjiang District,
Sichuan Province",
          "lat" : 30.6522796787,
          "lon" : 104.0725574128,
          "x" : 4387,
          "level.level" : 2,
          "level.range" : 19,
          "level.name" : "New tier-1 city",
          "y" : 4304,
          "cityNo" : 5
        }
      },
      {
        "_index" : "china",
        "_type" : "city",
        "_id" : "hangzhou",
        "_score" : 1.0,
        "_source" : {
          "name" : "Hangzhou",
          "province" : "No.316, Huancheng North Road, Gongshu District,
Zhejiang Province",
          "lat" : 30.2753694112,
          "lon" : 120.1509063337,
          "x" : 7530,
          "level.level" : 2,
          "level.range" : 19,
          "level.name" : "New tier-1 city",
          "y" : 4182,
          "cityNo" : 6
        }
      }
    ]
  }
}
```

Aggregation query

Sample SQL statement:

```
select level.level, count(1) from city group by level.level
curl -XGET http://10.0.0.2:9200/china/city/_search?pretty -H 'Content-Type:
application/json' -d'
```

```
{
    "size" : 0,
    "aggs" : {
        "city_level" : {
            "terms" : {
              "field" : "level.level"
            }
        }
    }
}'
```

The following response will be returned:

```
{
  "took" : 10,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 7,
    "max_score" : 0.0,
    "hits" : [ ]
  },
  "aggregations" : {
    "city_level" : {
      "doc_count_error_upper_bound" : 0,
      "sum_other_doc_count" : 0,
      "buckets" : [
        {
          "key" : 1,
          "doc_count" : 4
        },
        {
          "key" : 2,
          "doc_count" : 3
        }
      ]
    }
  }
}
```

Deleting Document

Deleting one document

Enter the following command:

```
curl -XDELETE 'http://10.0.0.2:9200/china/city/beijing?pretty' -H 'Content-Type: application/json'
```

The following response will be returned:

```
{
  "found" : true,
  "_index" : "china",
  "_type" : "city",
  "_id" : "beijing",
  "_version" : 5,
  "result" : "deleted",
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  }
}
```

Deleting type

```
curl -XDELETE 'http://10.0.0.2:9200/china/city?pretty' -H 'Content-Type: application/json'
```

Deleting index

```
curl -XDELETE 'http://10.0.0.2:9200/china?pretty' -H 'Content-Type: application/json'
```

Accessing Clusters from Kibana

Last updated：2025-01-24 11:32:43

ES comes with a Kibana module. You can access the Kibana page of your cluster to visually query, analyze, and manage your data. This tutorial is designed to help you get started with Kibana.

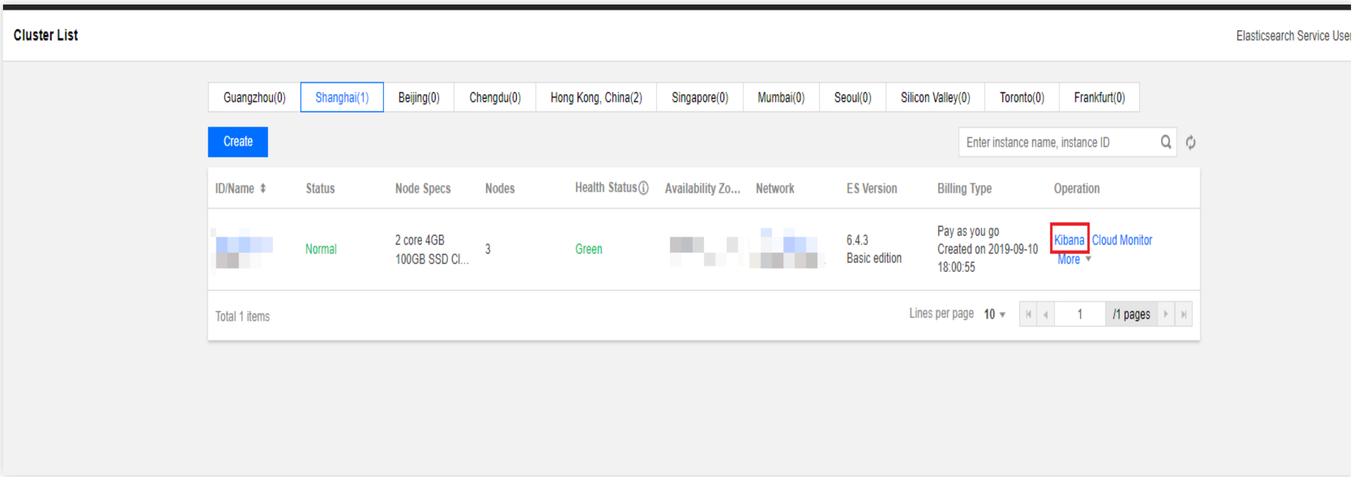
Accessing Kibana Page

Entries

There are two entries to the Kibana page, which are located on the cluster list page and the cluster details page, as shown below. Click either of them to jump to the Kibana login page.

Note:

By default, Kibana is accessed at the public address. If you are concerned that accessing Kibana over the public network will cause security problems, you can disable the Kibana public address and enable the Kibana private address for access on the cluster details page.

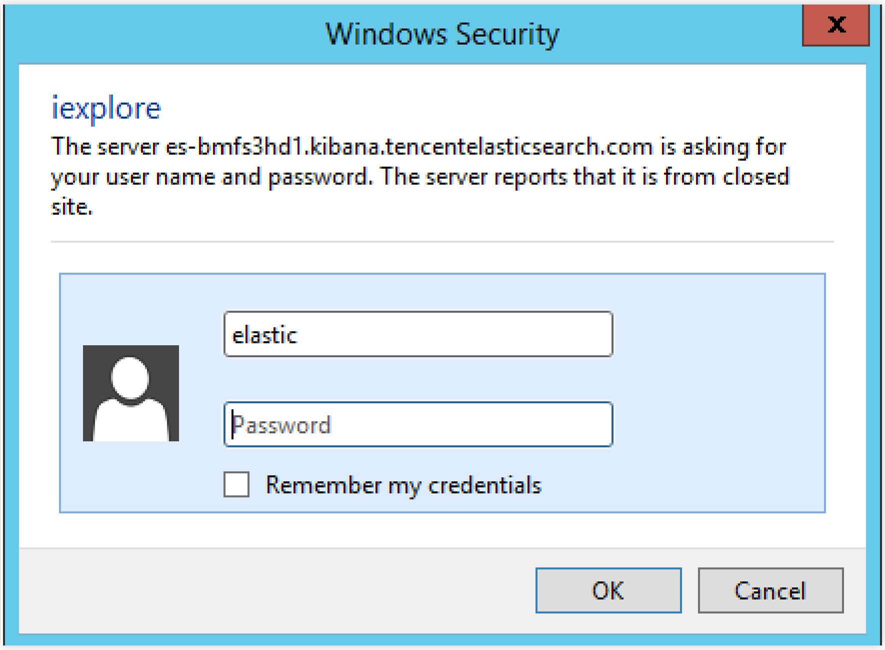


ID/Name	Status	Node Specs	Nodes	Health Status	Availability Zo...	Network	ES Version	Billing Type	Operation
[redacted]	Normal	2 core 4GB 100GB SSD Cl...	3	Green	[redacted]	[redacted]	6.4.3 Basic edition	Pay as you go Created on 2019-09-10 18:00:55	Kibana Cloud Monitor More

Login

To access the Kibana page, you need to log in with the username "elastic" and the Kibana password you set when you created your cluster. If you forgot your password, you can reset it on the cluster details page. For security reasons, you can configure an access blocklist/allowlist for the public address of the Kibana page. For more information, please see Kibana Access Settings.

If "ES cluster user authentication" is not enabled, the Kibana login page is as shown below:

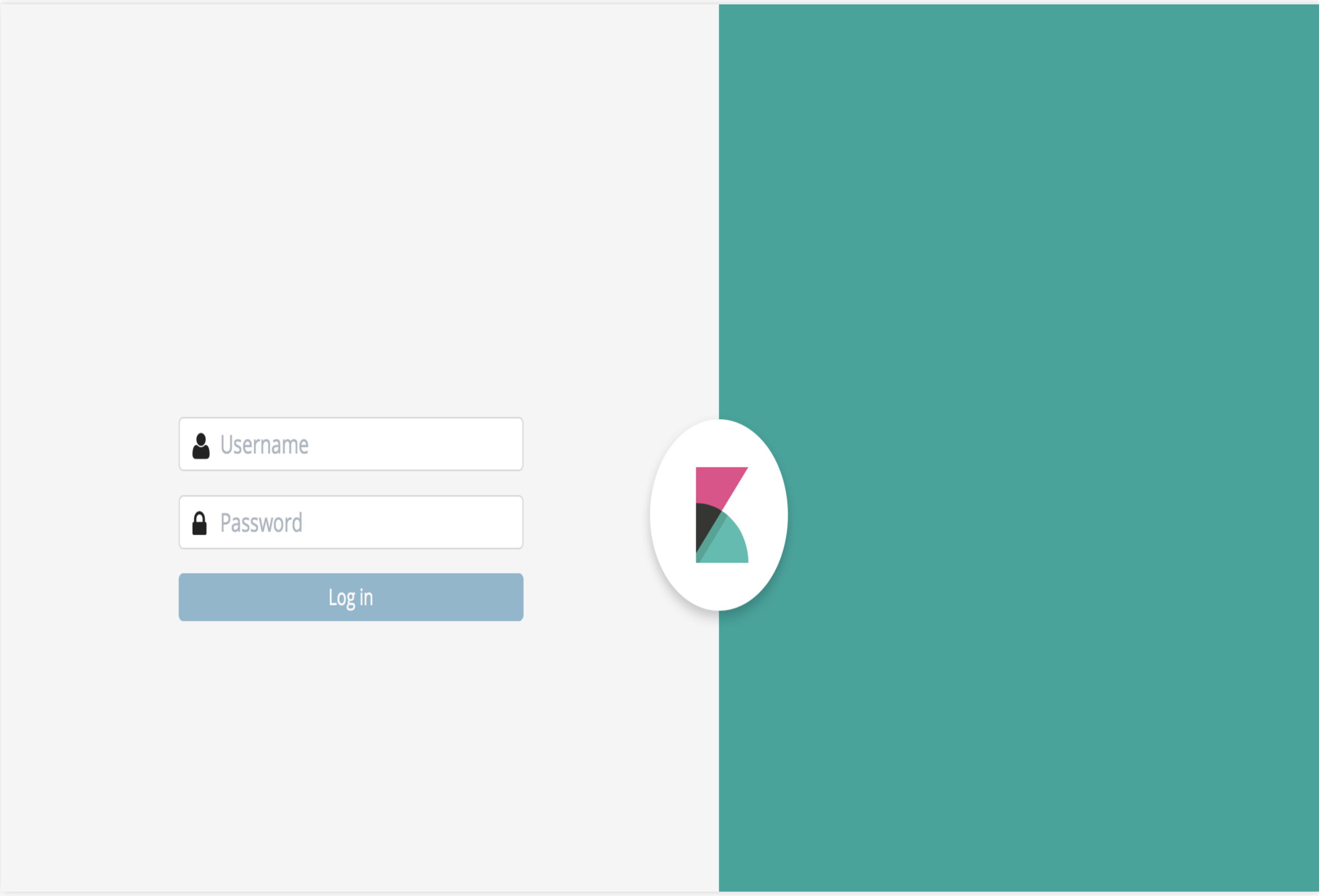


Windows Security

iexplore

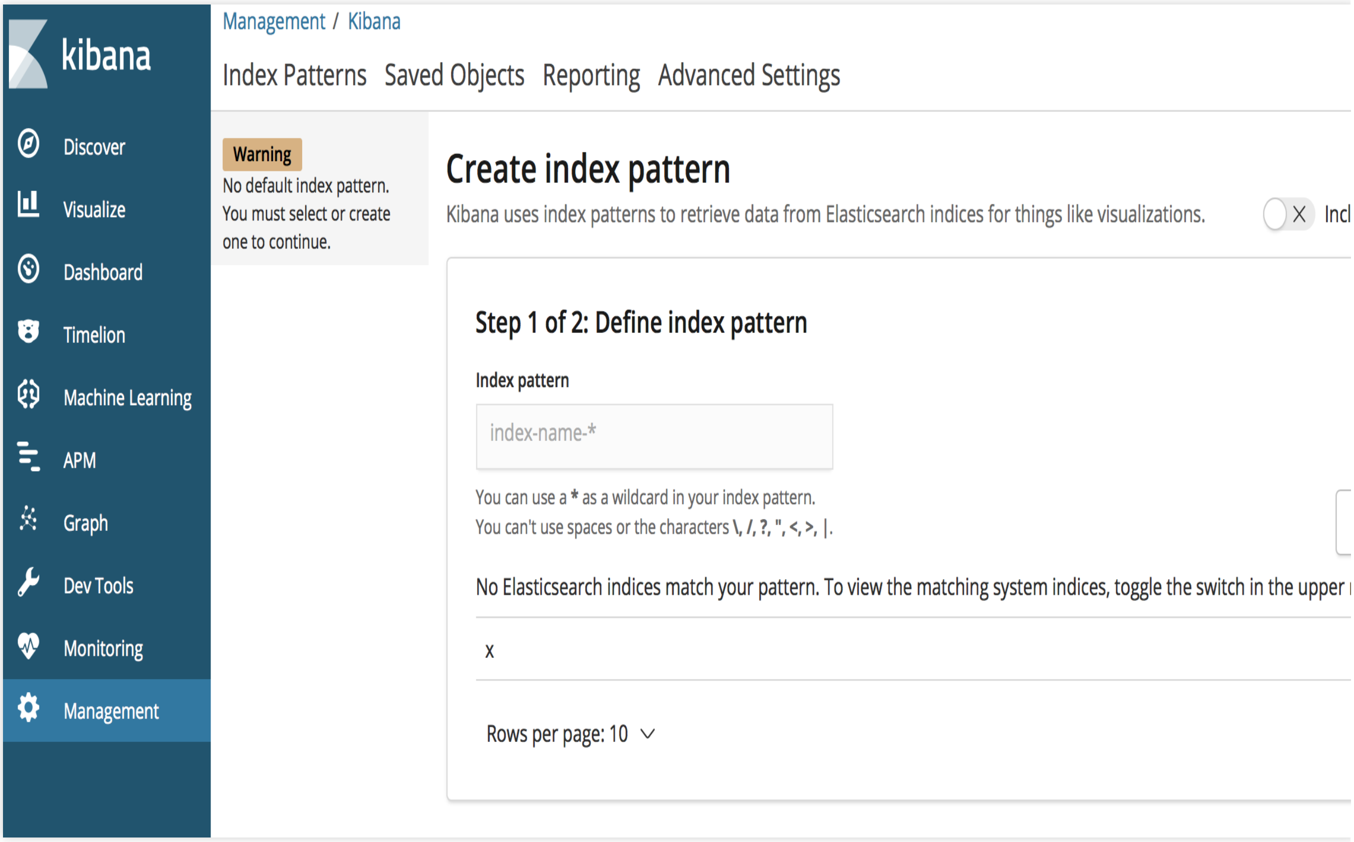
The server es-bmfs3hd1.kibana.tencentelasticsearch.com is asking for your user name and password. The server reports that it is from closed site.

If "ES cluster user authentication" is enabled, the Kibana login page is as shown below:



Access

After logging in to the Kibana page, if you are a new user, your cluster has not stored any custom indexed data, and you will be prompted to configure an index. For more information, please see Adding and Accessing Index.



No default index pattern. You must select or create one to continue.

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

Step 1 of 2: Define index pattern

Index pattern

index-name-*

You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, ", <, >, |.

No Elasticsearch indices match your pattern. To view the matching system indices, toggle the switch in the upper

x

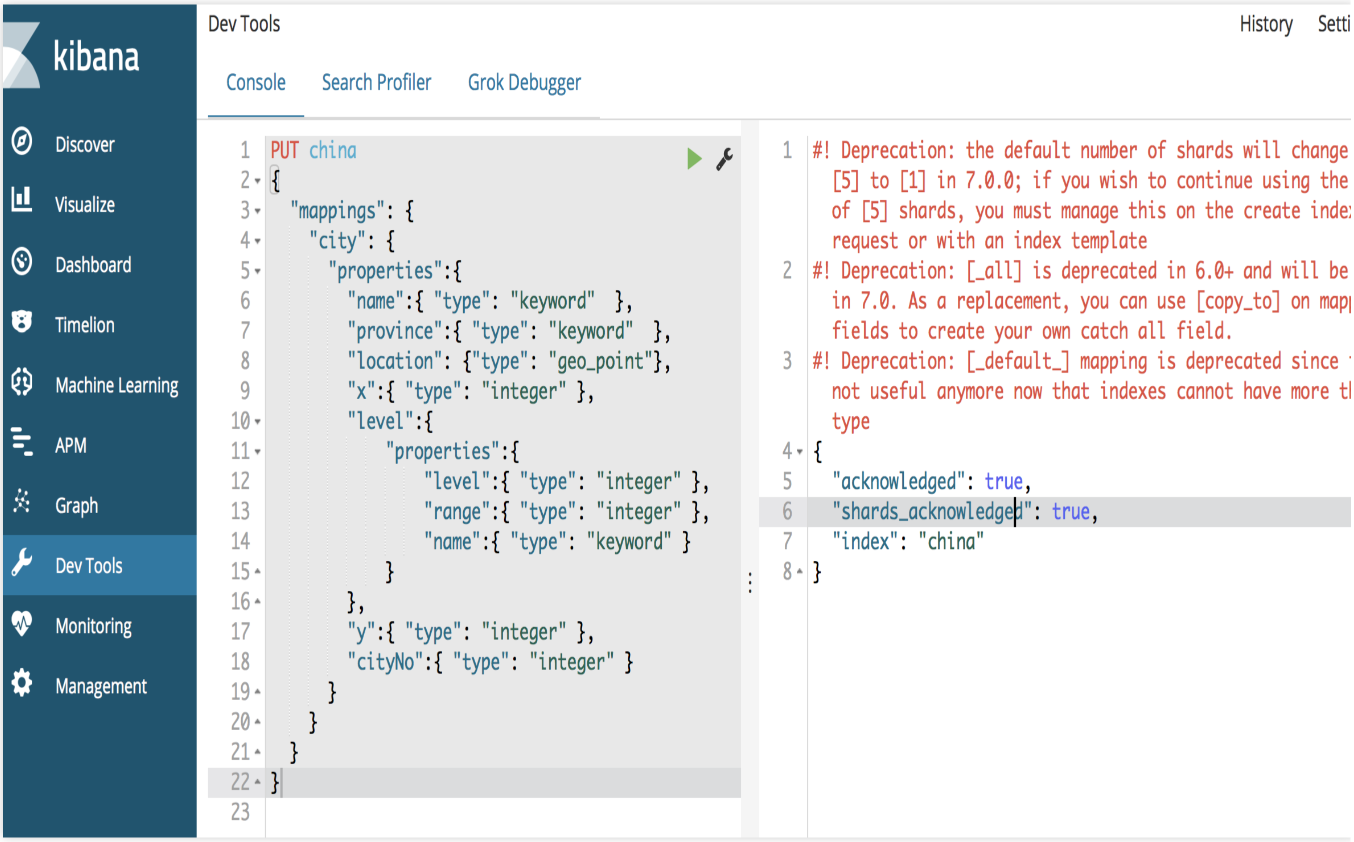
Adding and Accessing Index (Storing Data)

On the left sidebar on the Kibana page, click **Dev Tools** to enter the development tools page, where you can send various operation requests to your cluster through the console. The following shows how to manipulate your cluster and store data with sample city information.

Adding index

Defining the mapping of index

Specify the index name as `china` , type name as `city` , and detailed field and type information. The type of the `location` field is `geo_point` which can represent location information, and `level` is the object type and contains subfield information. For more information on field types, please see Field Datatypes.

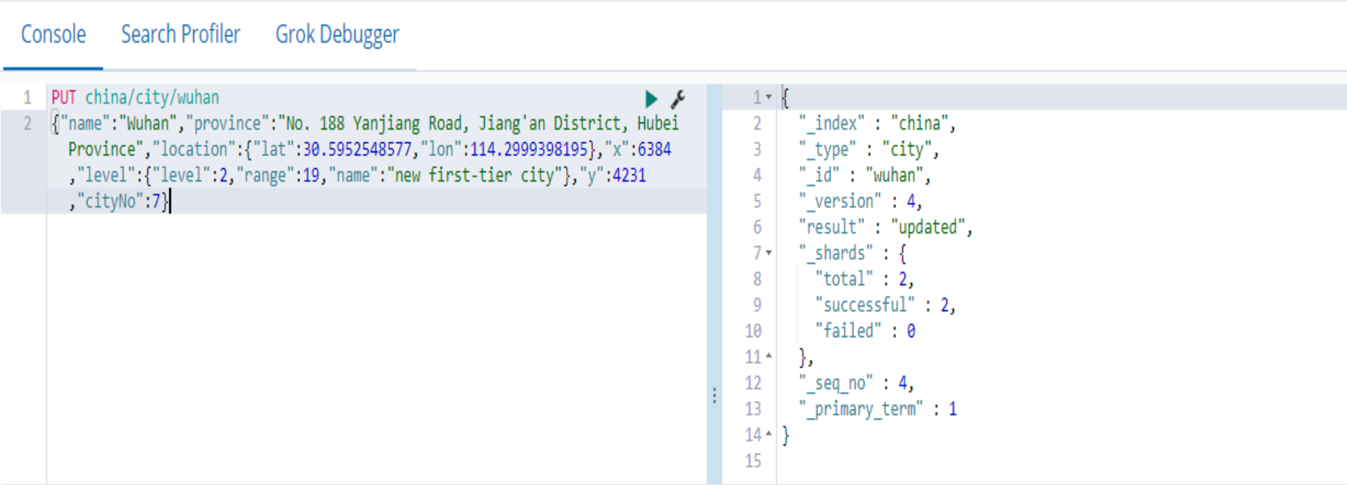


```
PUT china
{
  "mappings": {
    "city": {
      "properties":{
        "name":{ "type": "keyword"  },
        "province":{ "type": "keyword"  },
        "location": {"type": "geo_point"},
        "x":{ "type": "integer" },
        "level":{
            "properties":{
                "level":{ "type": "integer" },
                "range":{ "type": "integer" },
                "name":{ "type": "keyword" }
            }
        },
        "y":{ "type": "integer" },
        "cityNo":{ "type": "integer" }
      }
    }
  }
}
```

```
#! Deprecation: the default number of shards will change
  [5] to [1] in 7.0.0; if you wish to continue using the
  of [5] shards, you must manage this on the create inde
  request or with an index template
#! Deprecation: [_all] is deprecated in 6.0+ and will be
  in 7.0. As a replacement, you can use [copy_to] on map
  fields to create your own catch all field.
#! Deprecation: [_default_] mapping is deprecated since
  not useful anymore now that indexes cannot have more t
  type
{
  "acknowledged": true,
  "shards_acknowledged": true,
  "index": "china"
}
```

```
PUT china
{
  "mappings": {
    "city": {
      "properties":{
        "name":{ "type": "keyword"  },
        "province":{ "type": "keyword"  },
        "location": {"type": "geo_point"},
        "x":{ "type": "integer" },
        "level":{
            "properties":{
                "level":{ "type": "integer" },
                "range":{ "type": "integer" },
                "name":{ "type": "keyword" }
            }
        },
        "y":{ "type": "integer" },
        "cityNo":{ "type": "integer" }
      }
    }
  }
}
```

Adding one single document



```
PUT china/city/wuhan
{"name":"Wuhan","province":"No. 188 Yanjiang Road, Jiang'an District, Hubei
  Province","location":{"lat":30.5952548577,"lon":114.2999398195},"x":6384
  ,"level":{"level":2,"range":19,"name":"new first-tier city"},"y":4231
  ,"cityNo":7}
```

```
{
  "_index" : "china",
  "_type" : "city",
  "_id" : "wuhan",
  "_version" : 4,
  "result" : "updated",
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  },
  "_seq_no" : 4,
  "_primary_term" : 1
}
```

```
PUT china/city/wuhan
{"name":"Wuhan","province":"No.188, Yanjiang Avenue, Jiang'an District, Hubei
Province","location":
{"lat":30.5952548577,"lon":114.2999398195},"x":6384,"level":
{"level":2,"range":19,"name":"New first-tier city"},"y":4231,"cityNo":7}
```

Querying one single document

```
GET /china/city/wuhan
```

Adding multiple documents

```
POST _bulk
{ "index" : { "_index": "china", "_type" : "city", "_id" : "beijing" } }
{"name":"Beijing","province":"Beijing","location":
{"lat":39.9031324643,"lon":116.4010433787},"x":6763,"level":
{"range":4,"level":1,"name":"First-tier city"},"y":6381,"cityNo":1}
{ "index" : { "_index": "china", "_type" : "city", "_id" : "shanghai" } }
{"name":"Shanghai","province":"Shanghai","location":
{"lat":31.2319526784,"lon":121.469443249},"x":7779,"level":
{"range":4,"level":1,"name":"First-tier city"},"y":4409,"cityNo":2}
{ "index" : { "_index": "china", "_type" : "city", "_id" : "guangzhou" } }
{"name":"Guangzhou","province":"No.79, Jixiang Road, Yuexiu District, Guangdong
Province","location":
{"lat":23.1317146641,"lon":113.2595185241},"x":6173,"level":
{"range":4,"level":1,"name":"First-tier city"},"y":2560,"cityNo":3}
{ "index" : { "_index": "china", "_type" : "city", "_id" : "shenzhen" } }
```

```
{"name":"Shenzhen","province":"No.37, Xinyuan Road, Futian District, Guangdong
Province","location":
{"lat":22.5455465546,"lon":114.0527779134},"x":6336,"level":
{"range":4,"level":1,"name":"First-tier city"},"y":2429,"cityNo":4}
{ "index" : { "_index": "china", "_type" : "city", "_id" : "chengdu" } }
{"name":"Chengdu","province":"No. 88-1, Hongxing Road 4th Section, Jinjiang
District, Sichuan Province","location":
{"lat":30.6522796787,"lon":104.0725574128},"x":4387,"level":
{"level":2,"range":19,"name":"New first-tier city"},"y":4304,"cityNo":5}
{ "index" : { "_index": "china", "_type" : "city", "_id" : "hangzhou" } }
{"name":"Hangzhou","province":"No.316, Huancheng North Road, Gongshu District,
Zhejiang Province","location":
{"lat":30.2753694112,"lon":120.1509063337},"x":7530,"level":
{"level":2,"range":19,"name":"New first-tier city"},"y":4182,"cityNo":6}
```

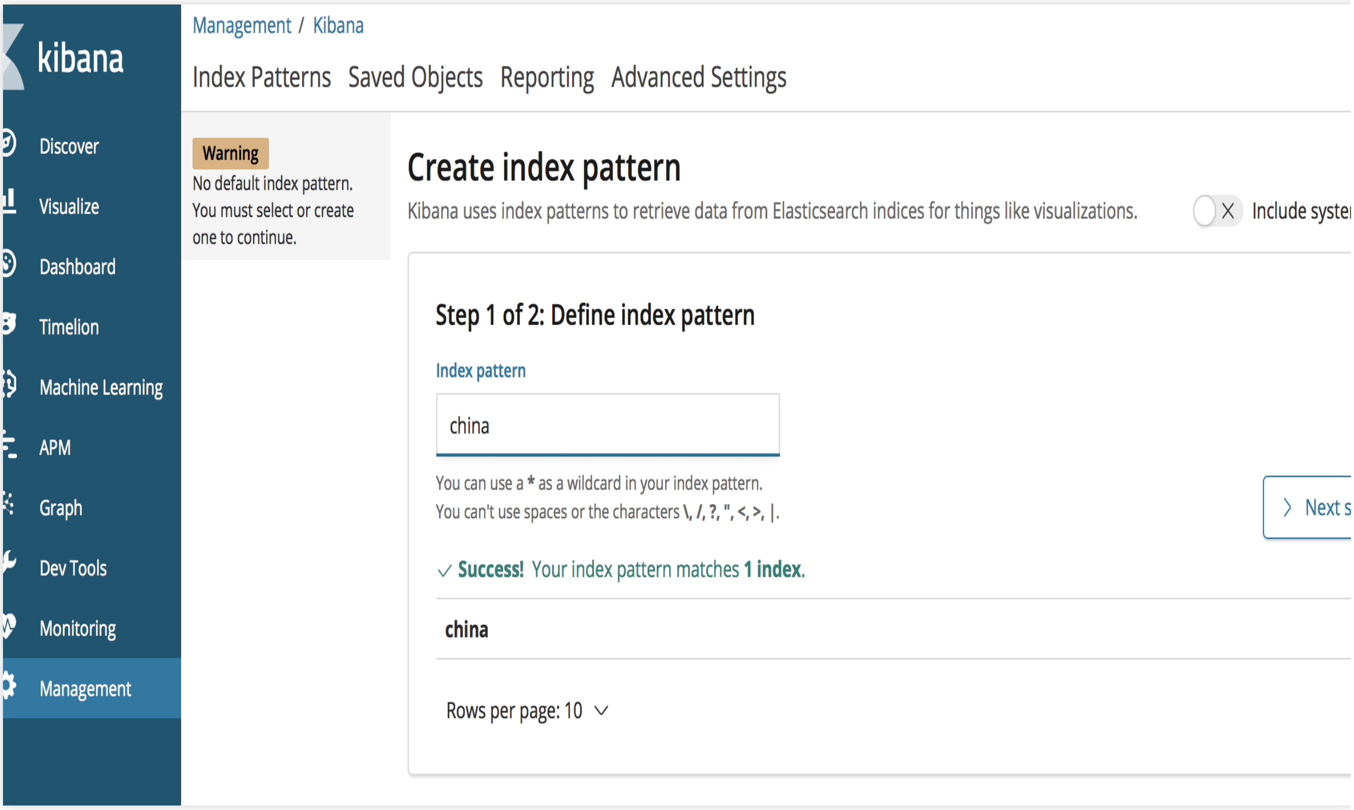
Querying multiple documents

```
GET /china/city/_search
```

Accessing index

Configuring Kibana to access index

To use Kibana, you need to configure at least one index that can be matched. Enter the index `china` created above and click **Next step** to proceed to the next step.



Warning

No default index pattern. You must select or create one to continue.

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

Step 1 of 2: Define index pattern

Index pattern: china

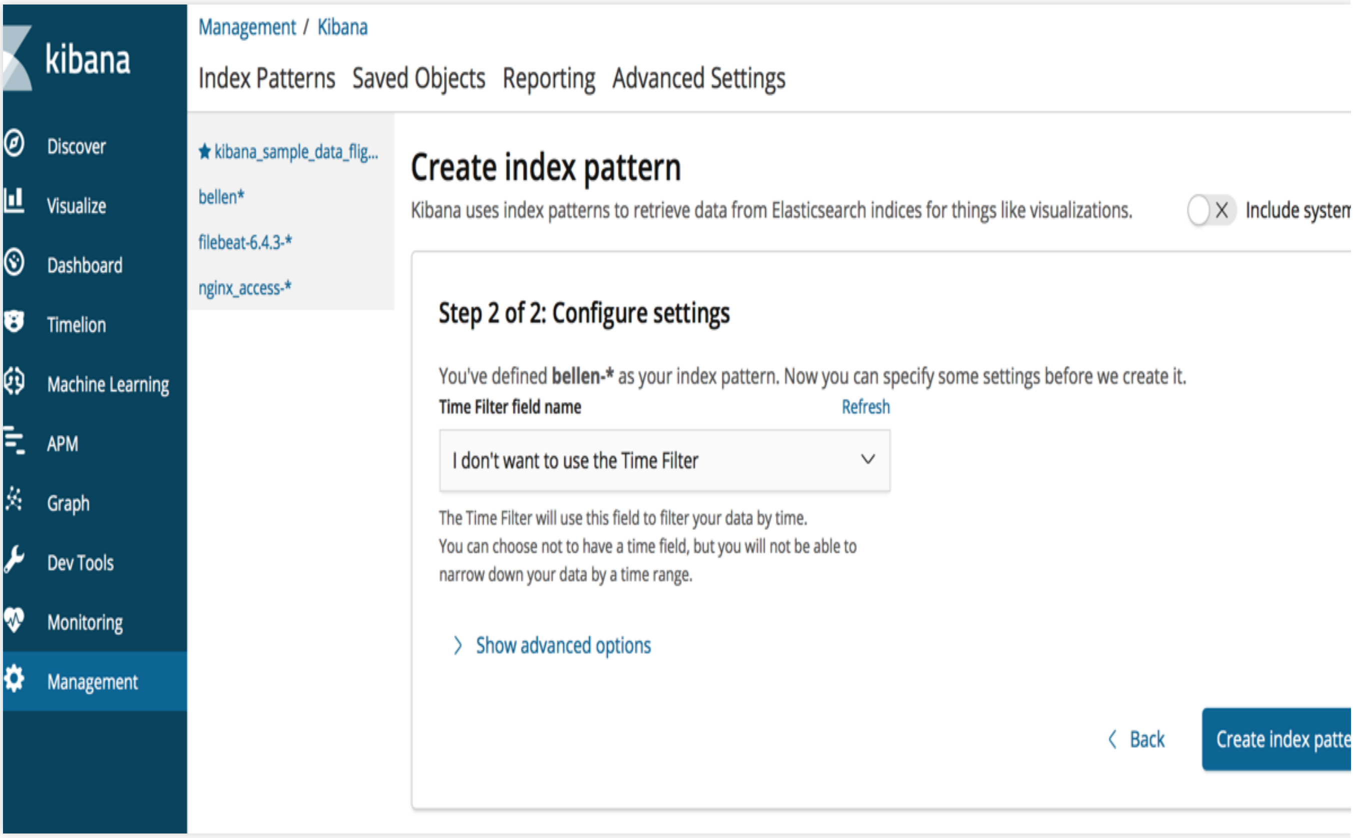
You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, ", <, >, |.

✓ **Success!** Your index pattern matches **1 index.**

china

Rows per page: 10

You need to **configure a time filter field to filter** the data in an index by time. If there is no field in the index that indicates time, you can choose not to use the time filter feature. Click **Create index pattern** to create an index pattern.



Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

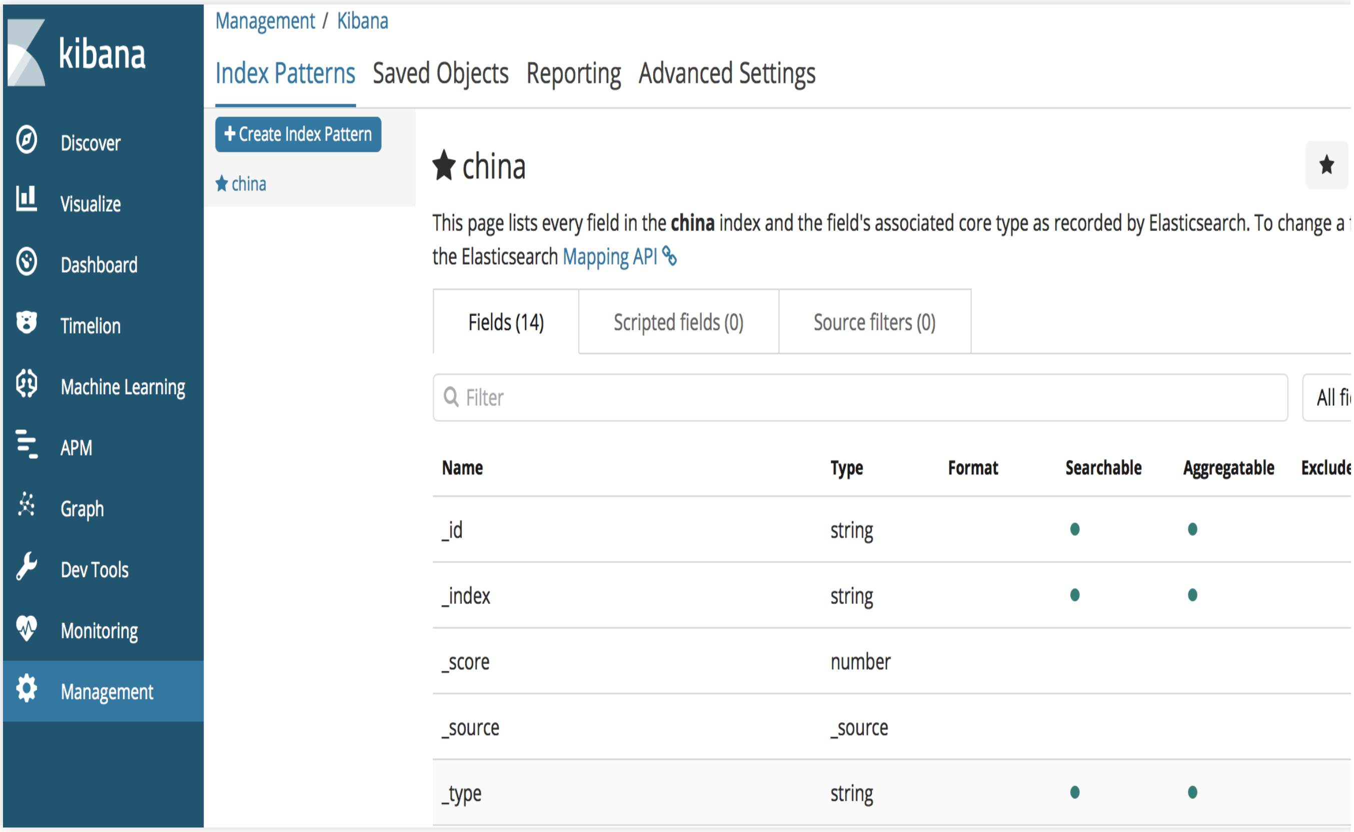
Step 2 of 2: Configure settings

You've defined **bellen-*** as your index pattern. Now you can specify some settings before we create it.

Time Filter field name: I don't want to use the Time Filter

The Time Filter will use this field to filter your data by time.
You can choose not to have a time field, but you will not be able to narrow down your data by a time range.

View the fields for the index.

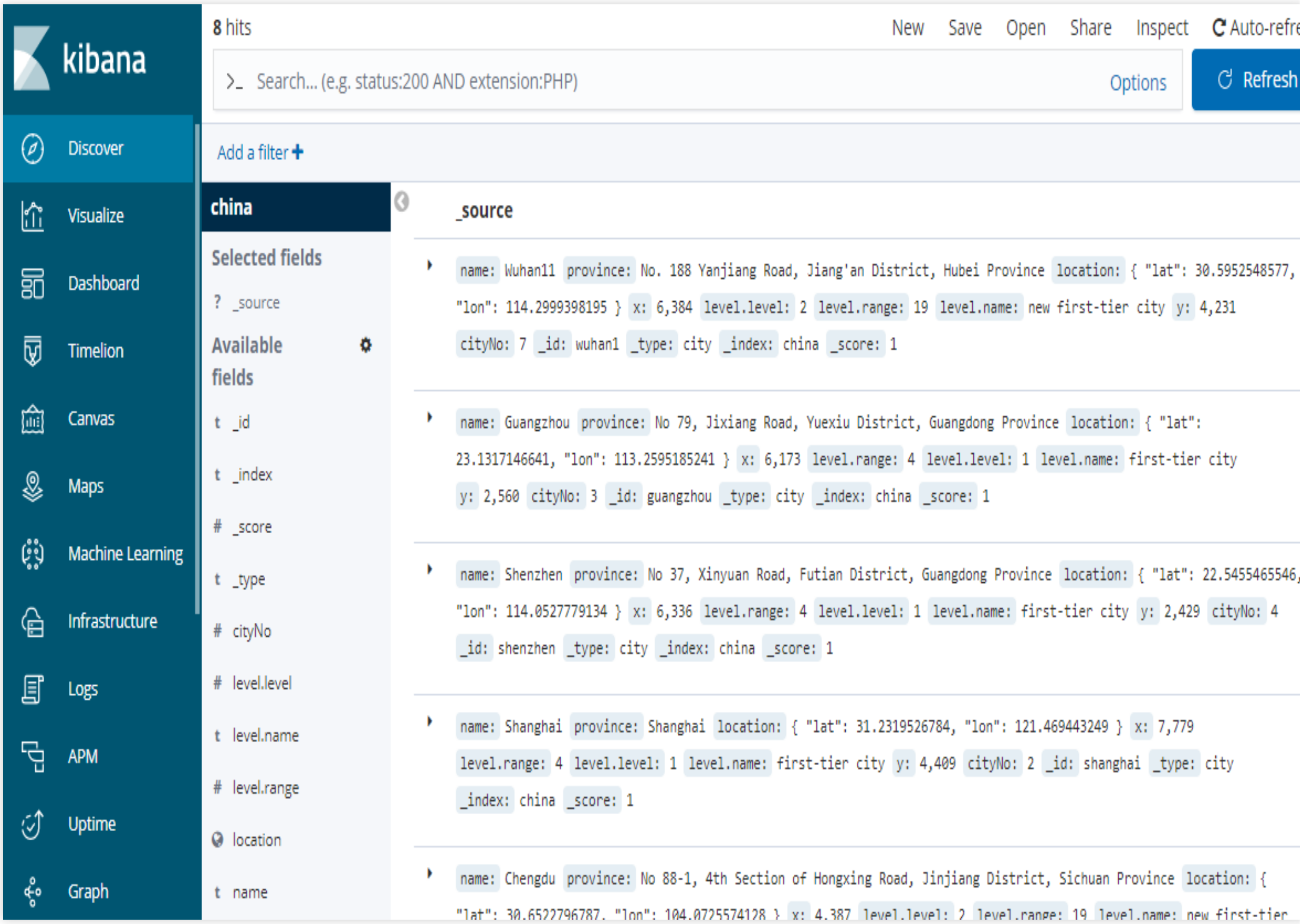


china

This page lists every field in the **china** index and the field's associated core type as recorded by Elasticsearch. To change a f
the Elasticsearch Mapping API

Name	Type	Format	Searchable	Aggregatable	Exclude
_id	string		●	●	
_index	string		●	●	
_score	number				
_source	_source				
_type	string		●	●	

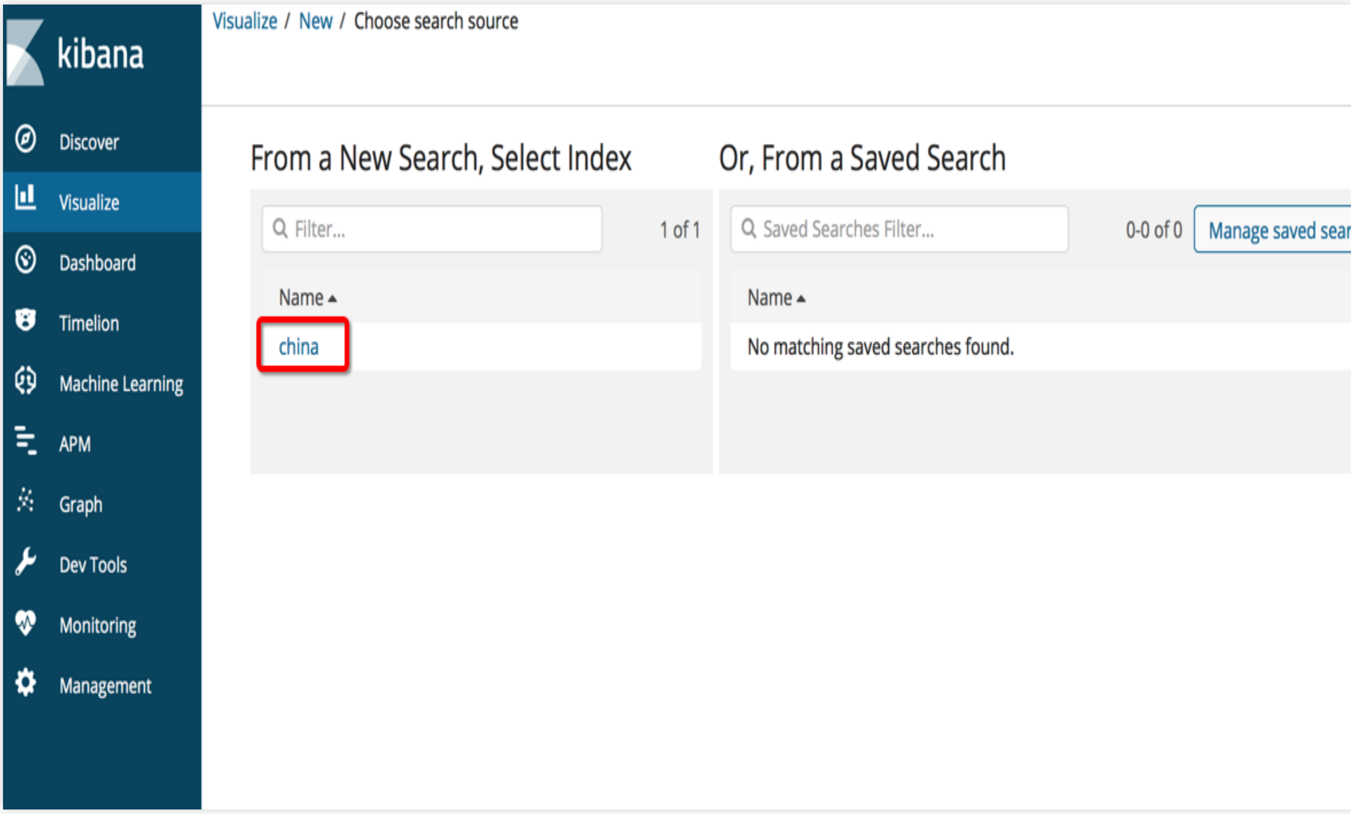
Click **Discover** on the left sidebar to view the documents that have been added under the index.



```
name: Wuhan11 province: No. 188 Yanjiang Road, Jiang'an District, Hubei Province location: { "lat": 30.5952548577,
"lon": 114.2999398195 } x: 6,384 level.level: 2 level.range: 19 level.name: new first-tier city y: 4,231
cityNo: 7 _id: wuhan1 _type: city _index: china _score: 1

name: Guangzhou province: No 79, Jixiang Road, Yuexiu District, Guangdong Province location: { "lat":
23.1317146641, "lon": 113.2595185241 } x: 6,173 level.range: 4 level.level: 1 level.name: first-tier city
y: 2,560 cityNo: 3 _id: guangzhou _type: city _index: china _score: 1

name: Shenzhen province: No 37, Xinyuan Road, Futian District, Guangdong Province location: { "lat": 22.5455465546,
"lon": 114.0527779134 } x: 6,336 level.range: 4 level.level: 1 level.name: first-tier city y: 2,429 cityNo: 4
_id: shenzhen _type: city _index: china _score: 1

name: Shanghai province: Shanghai location: { "lat": 31.2319526784, "lon": 121.469443249 } x: 7,779
level.range: 4 level.level: 1 level.name: first-tier city y: 4,409 cityNo: 2 _id: shanghai _type: city
_index: china _score: 1

name: Chengdu province: No 88-1, 4th Section of Hongxing Road, Jinjiang District, Sichuan Province location: {
"lat": 30.6522796787, "lon": 104.0725574128 } x: 4,387 level.level: 2 level.range: 19 level.name: new first-tier
```

Visual Query and Analysis

Kibana can perform visual statistical analysis of data. Click **Visualize** on the left sidebar to configure various visual charts for data analysis. For example, to count the different levels under the `china` index mentioned above, follow the steps below:



From a New Search, Select Index

Filter... 1 of 1

Name ▲

china

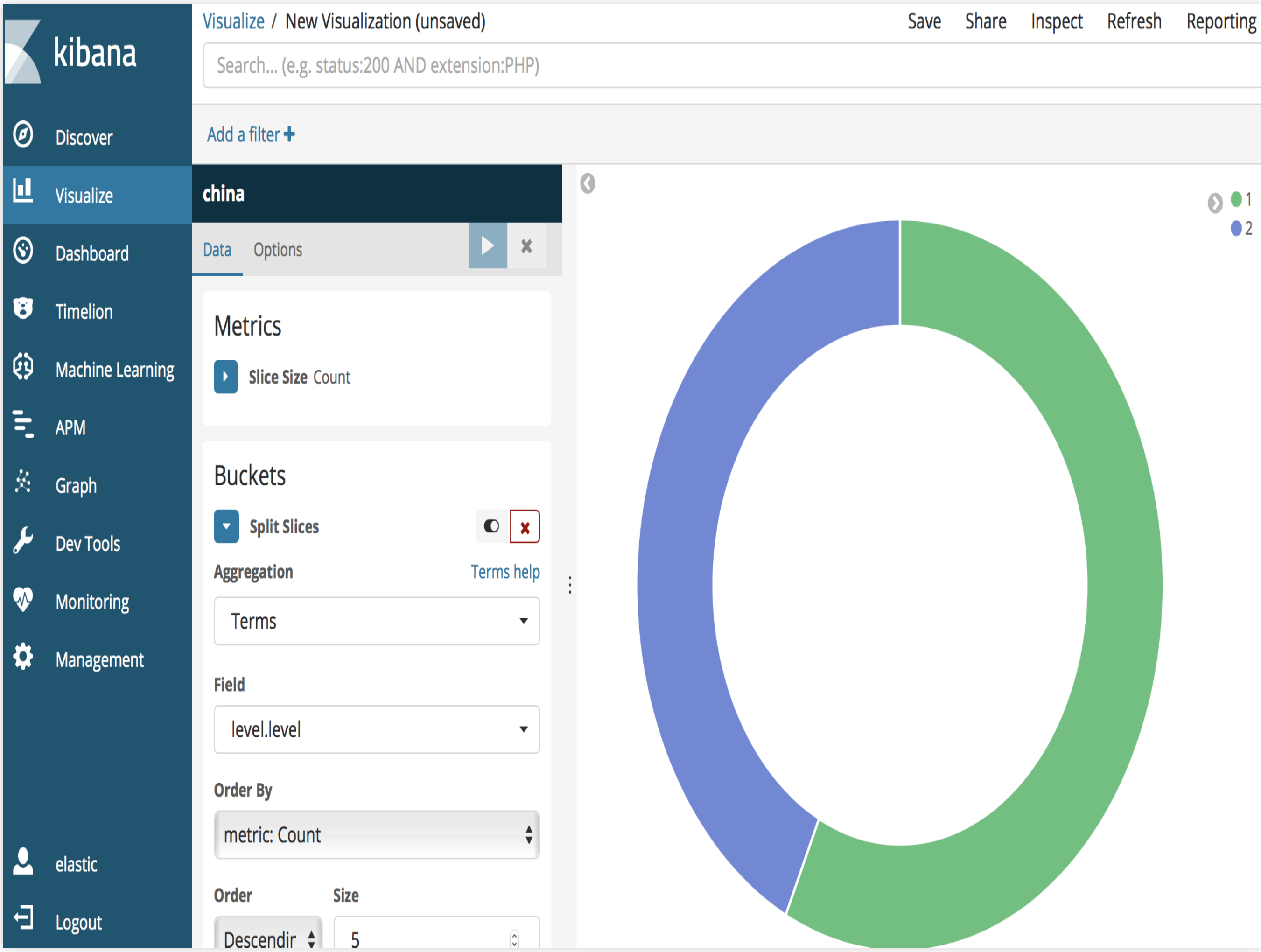
Or, From a Saved Search

Saved Searches Filter... 0-0 of 0 Manage saved sear

Name ▲

No matching saved searches found.

Select the `count` metric, aggregate the statistics by the `level.level` field, and click **Save**.



Visualize / New Visualization (unsaved)

Search... (e.g. status:200 AND extension:PHP)

Add a filter +

china

Metrics

Slice Size Count

Buckets

Split Slices

Aggregation: Terms

Field: level.level

Order By: metric: Count

Order: Descendir

Size: 5

For more information on how to use Kibana, please see Kibana's official documentation.