

ALIBABA CLOUD

阿里云

阿里云Elasticsearch 开发指南

文档版本：20210115

 阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击 确定 。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

- 1.Restful API ----- 05
- 2.Java API ----- 08
 - 2.1. 概述 ----- 08
 - 2.2. High Level REST Client (7.x) ----- 09
 - 2.3. High Level REST Client (6.7.x) ----- 12
 - 2.4. High Level REST Client (6.3.x) ----- 15
 - 2.5. Low Level REST Client (5.x) ----- 19
 - 2.6. Transport Client (5.x) ----- 21
- 3.通过客户端访问阿里云Elasticsearch ----- 26
- 4.通过API操作阿里云Elasticsearch ----- 32
- 5.Java API FAQ ----- 37

1. Restful API

Elasticsearch采用REST API，所有的操作都可通过HTTP API完成，例如增删改查、别名配置等。本文介绍Restful API的使用方法。

Elasticsearch官方参考文档


- 单文档API:
 - [Index API](#)
 - [Get API](#)
 - [Delete API](#)
 - [Update API](#)
- 多文档API:
 - [Multi Get API](#)
 - [Bulk API](#)
 - [Delete By Query API](#)
 - [Update By Query API](#)
 - [Reindex API](#)

使用REST Client交互

客户端访问仅支持HTTP / TCP方式，建议您采用Elasticsearch官方提供的[Java REST Client](#)。

使用Java API交互

Elasticsearch为Java用户提供了内置客户端，包括Transport Client和Java REST Client。Elasticsearch 7.0中已经弃用Transport Client，在8.0中完全移除它。因此在实际开发中建议您使用Java REST Client。关于Java API的更多信息，请参见官方[Java API](#)以及[Java API使用说明](#)文档。

 **说明** Java客户端所使用的Elasticsearch版本必须与Elasticsearch集群版本一致，否则可能无法相互识别。

RESTful API (HTTP) 请求参数说明

您可以使用RESTful API或者curl命令，通过9200端口与Elasticsearch进行通信。

 **说明** Elasticsearch官方提供了多种程序语言的客户端，例如 Groovy、Javascript、.NET、PHP、Perl、Python 以及 Ruby 以及很多由社区提供的客户端和插件，您可以在[Community Contributed Clients](#)中获取。

curl命令 (HTTP) 由以下几部分组成。

```
curl -X<VERB> '<PROTOCOL>://<HOST>:<PORT>/<PATH>?<QUERY_STRING>' -d '<BODY>'
```

参数	说明
VERB	HTTP方法, 包括 GET 、 POST 、 PUT 、 HEAD 、 DELETE 。
PROTOCOL	http 或者 https 协议 (只有在Elasticsearch前面有 https 代理的时候可用)。
HOST	Elasticsearch集群中的任何一个节点的主机名, 对于本地节点, 可使用 localhost 。
PORT	Elasticsearch HTTP服务所在的端口, 默认为 9200 。
PATH	API路径 (例如 _count 将返回集群中文档的数量), PATH 可以包含多个组件, 例如 _cluster/stats 或者 _nodes/stats/jvm 。
QUERY_STRING	可选的查询请求参数, 例如 ?pretty 参数可使请求返回的JSON数据更加美观易读。
BODY	JSON格式的请求主体 (可选)。

RESTful API (HTTP) 请求示例

通过以下命令, 统计Elasticserach集群中文档的数量。

```
curl -XGET 'http://localhost:9200/_count?pretty' -d '{
  "query": {
    "match_all": {}
  }
}'
```

返回结果如下。

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

使用 curl-i 显示HTTP请求头, 例如 curl-i -XGET 'localhost:9200/' , 完整请求示例如下。

```
curl -i -XGET 'localhost:9200/_count?pretty' -d '{
  "query": {
    "match_all": {}
  }
}'
```

简写请求示例如下。

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

2.Java API

2.1. 概述

本章节介绍Elasticsearch的Java Client的原理、版本兼容性以及使用示例，帮助您快速地使用Java客户端与Elasticsearch集群进行交互，完成检索、分析等相关业务。

Transport Client迁移至REST Client

Transport Client随着Elasticsearch的第一个版本诞生，是一个特别的客户端。特别之处在于，它使用TCP协议与Elasticsearch通信，这也造成了当客户端与不同版本的Elasticsearch通信时，会存在兼容性问题。详情请参见[Motivations around a new Java client](#)。

Elasticsearch官网于2016年发布Low Level REST客户端，该客户端基于Apache HTTP客户端，允许通过HTTP协议与任何版本的Elasticsearch集群通信。在Low Level REST客户端的基础上，Elasticsearch发布了High Level REST Client。

Elasticsearch 7.0中已经弃用Transport Client，在8.0中完全移除它。因此在实际开发中建议您使用Java REST Client。REST Client通过HTTP请求，帮助您处理请求和返回的序列化问题，为您的业务开发带来便捷。

注意

- 如果您需要使用Transport Client

通过9300端口访问阿里云Elasticsearch实例，请购买5.5或5.6版本的实例（6.x及以上版本不支持）。

- 使用Transport Client 5.5或5.6版本与Elasticsearch建立连接时会提示NoNodeAvailableException的错误。推荐使用Transport Client 5.3.3或[Java Low Level REST Client](#)来访问Elasticsearch集群，以保障版本的兼容性。如果使用Transport Client，需要在代码中将 `client.transport.sniff` 设置为false，详情请参见[Transport Client \(5.x\)](#)。

Java REST客户端

Java REST客户端有两种类型：

- Java Low Level REST Client：Elasticsearch Client低级别客户端。它允许通过HTTP请求与Elasticsearch集群进行通信。API本身不负责数据的编码解码，由用户去编码解码。它与所有的Elasticsearch版本兼容。
- Java High Level REST Client：Elasticsearch Client官方高级客户端。基于低级客户端，主要目标是为了暴露各API特定的方法。Java High Level REST Client依赖于Elasticsearch核心项目，将Request对象作为参数，返回一个Response对象。所有API都可以同步或异步调用。
 - 同步调用方法立即返回一个Response对象。
 - 而异步调用方法（方法名以async结尾）依赖于监听，当有请求返回或是错误返回时，该监听会通知到对应的方法继续执行。

本章节为您讲解Java Client的用法（建议优先使用REST Client），包括：

- [High Level REST Client \(7.x\)](#)
- [High Level REST Client \(6.3.x\)](#)
- [High Level REST Client \(6.7.x\)](#)
- [Low Level REST Client \(5.x\)](#)
- [Transport Client \(5.x\)](#)

相关文档

[Elasticsearch使用指南（云栖社区）](#)

2.2. High Level REST Client（7.x）

本文基于Java High Level REST Client 7.x版本，为您介绍Elasticsearch Java API的用法。


准备工作

- 安装Java，要求JDK版本为1.8及以上。

安装方法请参见[安装JDK](#)。

- 创建阿里云Elasticsearch实例，版本要求大于等于Java High Level REST Client的版本。

本文创建一个7.4.0版本的实例，创建方法请参见[创建阿里云Elasticsearch实例](#)。

 **注意** High Level Client能够向上兼容，例如7.4.0版本的Java High Level REST Client能确保与大于等于7.4.0版本的Elasticsearch集群通信。为了保证最大程度地使用最新版客户端的特性，推荐High Level Client版本与集群版本一致。

- 开启阿里云Elasticsearch实例的自动创建索引功能。

具体操作步骤请参见[开启自动创建索引](#)。

如果未开启会提示如下报错。

```
IndexNotFoundException[no such index and [action.auto_create_index] contains [-*] which forbids automatic creation of the index]
```

- 配置阿里云Elasticsearch实例的白名单，确保网络互通。

- 如果运行Java代码的服务器在公网环境下，可通过阿里云Elasticsearch实例的公网地址进行连通。连通前，需要开启阿里云Elasticsearch实例的公网地址，并修改公网地址访问白名单，将服务器的公网IP地址加入白名单中。具体操作步骤请参见[配置ES公网或私网访问白名单](#)。

 **注意**

- 如果您使用的是WIFI、宽带等网络，需要将公网出口的跳板机IP地址配置进去。建议您通过[淘宝IP地址库](#)查询。
- 您也可以将白名单配置为0.0.0.0/0，允许所有IPv4地址访问阿里云Elasticsearch实例。此配置会导致实例完全暴露在公网中，增加安全风险，配置前请确认您是否可以接受这个风险。

- 如果运行Java代码的服务器与阿里云Elasticsearch实例在同一专有网络VPC（Virtual Private Cloud）中，可通过阿里云Elasticsearch实例的内网地址进行连通。连通前，需要确保VPC私网访问白名单（默认为0.0.0.0/0）中已添加了服务器的内网IP地址。

- 创建Java Maven工程，并将如下的pom依赖添加到Java工程的pom.xml文件中。

pom依赖

```
<dependency>
  <groupId>org.elasticsearch.client</groupId>
  <artifactId>elasticsearch-rest-high-level-client</artifactId>
  <version>7.4.0</version>
</dependency>
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-core</artifactId>
  <version>2.7</version>
</dependency>
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-api</artifactId>
  <version>2.7</version>
</dependency>
```

示例

单击下载[完整示例代码](#)。

以下代码使用Index API创建索引，使用Delete API删除该索引，并演示了在JVM内存分配比较有限的客户端环境中，通过调整ResponseConsumer配置，限制异步响应所占用的缓存的大小。

```
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.elasticsearch.action.delete.DeleteRequest;
import org.elasticsearch.action.delete.DeleteResponse;
import org.elasticsearch.action.index.IndexRequest;
import org.elasticsearch.action.index.IndexResponse;
import org.elasticsearch.client.*;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
public class RestClientTest74 {
  private static final RequestOptions COMMON_OPTIONS;
  static {
    RequestOptions.Builder builder = RequestOptions.DEFAULT.toBuilder();
    // 默认缓存限制为100MB，此处修改为30MB。
    builder.setHttpAsyncResponseConsumerFactory(

```

```
builder.setHttpAsyncResponseConsumerFactory(
    new HttpAsyncResponseConsumerFactory
        .HeapBufferedResponseConsumerFactory(30 * 1024 * 1024));
COMMON_OPTIONS = builder.build();
}

public static void main(String[] args) {
    // 阿里云Elasticsearch集群需要basic auth验证。
    final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
    //访问用户名和密码为您创建阿里云Elasticsearch实例时设置的用户名和密码，也是Kibana控制台的登录用户名和
    密码。
    credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials("{访问用户名}",
        "{访问密码}"));
    // 通过builder创建rest client，配置http client的HttpClientConfigCallback。
    // 单击所创建的Elasticsearch实例ID，在基本信息页面获取公网地址，即为ES集群地址。
    RestClientBuilder builder = RestClient.builder(new HttpHost("{ES集群地址}", 9200))
        .setHttpClientConfigCallback(new RestClientBuilder.HttpClientConfigCallback() {
            @Override
            public HttpAsyncClientBuilder customizeHttpClient(HttpAsyncClientBuilder httpClientBuilder) {
                return httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
            }
        });
    // RestHighLevelClient实例通过REST low-level client builder进行构造。
    RestHighLevelClient highClient = new RestHighLevelClient(builder);
    try {
        // 创建request。
        Map<String, Object> jsonMap = new HashMap<>();
        // field_01、field_02为字段名，value_01、value_02为对应的值。
        jsonMap.put("{field_01}", "{value_01}");
        jsonMap.put("{field_02}", "{value_02}");
        //index_name为索引名称；type_name为类型名称,7.0及以上版本必须为_doc；doc_id为文档的id。
        IndexRequest indexRequest = new IndexRequest("{index_name}", "_doc", "{doc_id}").source(jsonMap)
;
        // 同步执行，并使用自定义RequestOptions（COMMON_OPTIONS）。
        IndexResponse indexResponse = highClient.index(indexRequest, COMMON_OPTIONS);
        long version = indexResponse.getVersion();
        System.out.println("Index document successfully! " + version);
        //index_name为索引名称；type_name为类型名称,7.0及以上版本必须为_doc；doc_id为文档的id。与以上创
        建索引的名称和id相同。
        DeleteRequest request = new DeleteRequest("{index_name}", "_doc", "{doc_id}");
        DeleteResponse deleteResponse = highClient.delete(request, COMMON_OPTIONS);
        System.out.println("Delete document successfully! \n" + deleteResponse.toString() + "\n" + deleteRes
```

```
ponse.status());
    highClient.close();
} catch (IOException ioException) {
    // 异常处理。
}
}
```

以上示例代码中带 {} 的参数需要替换为您具体业务的参数，详情请参见代码注释。


更多Java High Level REST Client的使用特性，请参见[Java High Level REST Client 官方文档](#)。

2.3. High Level REST Client (6.7.x)

本文基于Java High Level REST Client 6.7.x版本，为您介绍Elasticsearch Java API的用法。


准备工作

- 安装Java，要求JDK版本为1.8及以上。
安装方法请参见[安装JDK](#)。
- 创建阿里云Elasticsearch实例，版本要求大于等于Java High Level REST Client的版本。
本文创建一个6.7.0版本的实例，创建方法请参见[创建阿里云Elasticsearch实例](#)。

 **注意** High Level Client能够向上兼容，例如6.7.0版本的Java High Level REST Client能确保与大于等于6.7.0版本的Elasticsearch集群通信。为了保证最大程度地使用最新版客户端的特性，推荐High Level Client版本与集群版本一致。

- 开启阿里云Elasticsearch实例的自动创建索引功能。
具体操作步骤请参见[开启自动创建索引](#)。
如果未开启会提示如下报错。

```
IndexNotFoundException[no such index and [action.auto_create_index] contains [-*] which forbids automatic creation of the index]
```
- 配置阿里云Elasticsearch实例的白名单，确保网络互通。
 - 如果运行Java代码的服务器在公网环境下，可通过阿里云Elasticsearch实例的公网地址进行连通。连通前，需要开启阿里云Elasticsearch实例的公网地址，并修改公网地址访问白名单，将服务器的公网IP地址加入白名单中。具体操作步骤请参见[配置ES公网或私网访问白名单](#)。

 **注意**

- 如果您使用的是WIFI、宽带等网络，需要将公网出口的跳板机IP地址配置进去。建议您通过[淘宝IP地址库](#)查询。
- 您也可以将白名单配置为0.0.0.0/0，允许所有IPv4地址访问阿里云Elasticsearch实例。此配置会导致实例完全暴露在公网中，增加安全风险，配置前请确认您是否可以接受这个风险。

- 如果运行Java代码的服务器与阿里云Elasticsearch实例在同一专有网络VPC（Virtual Private Cloud）中，可通过阿里云Elasticsearch实例的内网地址进行连通。连通前，需要确保VPC私网访问白名单（默认为0.0.0.0/0）中已添加了服务器的内网IP地址。

- 创建Java Maven工程，并将如下的pom依赖添加到Java工程的pom.xml文件中。

pom依赖

```
<dependency>
  <groupId>org.elasticsearch.client</groupId>
  <artifactId>elasticsearch-rest-high-level-client</artifactId>
  <version>6.7.0</version>
</dependency>
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-core</artifactId>
  <version>2.7</version>
</dependency>
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-api</artifactId>
  <version>2.7</version>
</dependency>
```

RequestOptions新特性

相较于6.3.2的REST Client，6.7.0增加了RequestOptions选项。您可以通过它对请求进行更多自定义的配置，且不影响正常的Elasticsearch请求。

以下示例将演示在JVM内存分配比较有限的客户端环境中，通过调整ResponseConsumer配置，限制异步响应所占用的缓存的大小。完整示例请参见[示例](#)。

```
private static final RequestOptions COMMON_OPTIONS;
static {
  RequestOptions.Builder builder = RequestOptions.DEFAULT.toBuilder();
  // 默认缓存限制为100MB，此处修改为30MB。
  builder.setHttpAsyncResponseConsumerFactory(
    new HttpAsyncResponseConsumerFactory
      .HeapBufferedResponseConsumerFactory(30 * 1024 * 1024));
  COMMON_OPTIONS = builder.build();
}
```

```
// 同步执行，并使用自定义的RequestOptions (COMMON_OPTIONS)。
IndexResponse indexResponse = highClient.index(indexRequest, COMMON_OPTIONS);
```

更多RequestOptions的用法请参见[官方文档](#)。

示例

单击下载[完整示例代码](#)。

以下代码使用Index API创建索引，使用Delete API删除该索引，并演示了在JVM内存分配比较有限的客户端环境中，通过调整ResponseConsumer配置，限制异步响应所占用的缓存的大小。

```
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.elasticsearch.action.delete.DeleteRequest;
import org.elasticsearch.action.delete.DeleteResponse;
import org.elasticsearch.action.index.IndexRequest;
import org.elasticsearch.action.index.IndexResponse;
import org.elasticsearch.client.*;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
public class RestClientTest67 {
    private static final RequestOptions COMMON_OPTIONS;
    static {
        RequestOptions.Builder builder = RequestOptions.DEFAULT.toBuilder();
        // 默认缓存限制为100MB，此处修改为30MB。
        builder.setHttpAsyncResponseConsumerFactory(
            new HttpAsyncResponseConsumerFactory
                .HeapBufferedResponseConsumerFactory(30 * 1024 * 1024));
        COMMON_OPTIONS = builder.build();
    }
    public static void main(String[] args) {
        // 阿里云Elasticsearch集群需要basic auth验证。
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        //访问用户名和密码为您创建阿里云Elasticsearch实例时设置的用户名和密码，也是Kibana控制台的登录用户名和密码。
        credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials("{访问用户名}",
            "{访问密码}"));
        // 通过builder创建rest client，配置http client的HttpClientConfigCallback。
        // 单击所创建的Elasticsearch实例ID，在基本信息页面获取公网地址，即为ES集群地址。
        RestClientBuilder builder = RestClient.builder(new HttpHost("{ES集群地址}", 9200))
            .setHttpClientConfigCallback(new RestClientBuilder.HttpClientConfigCallback() {
                @Override
```

```
public HttpAsyncClientBuilder customizeHttpClient(HttpAsyncClientBuilder httpClientBuilder) {
    return httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
}
});
// RestHighLevelClient实例通过REST low-level client builder进行构造。
RestHighLevelClient highClient = new RestHighLevelClient(builder);
try {
    // 创建request。
    Map<String, Object> jsonMap = new HashMap<>();
    // field_01、field_02为字段名，value_01、value_02为对应的值。
    jsonMap.put("{field_01}", "{value_01}");
    jsonMap.put("{field_02}", "{value_02}");
    //index_name为索引名称；type_name为类型名称；doc_id为文档的id。
    IndexRequest indexRequest = new IndexRequest("{index_name}", "{type_name}", "{doc_id}").source(jsonMap);
    // 同步执行，并使用自定义RequestOptions（COMMON_OPTIONS）。
    IndexResponse indexResponse = highClient.index(indexRequest, COMMON_OPTIONS);
    long version = indexResponse.getVersion();
    System.out.println("Index document successfully! " + version);
    //index_name为索引名称；type_name为类型名称；doc_id为文档的id。与以上创建索引的名称和id相同。
    DeleteRequest request = new DeleteRequest("{index_name}", "{type_name}", "{doc_id}");
    DeleteResponse deleteResponse = highClient.delete(request, COMMON_OPTIONS);
    System.out.println("Delete document successfully! \n" + deleteResponse.toString() + "\n" + deleteResponse.status());
    highClient.close();
} catch (IOException ioException) {
    // 异常处理。
}
}
```

以上示例代码中带 `{}` 的参数需要替换为您具体业务的参数，详情请参见代码注释。

更多Java High Level REST Client的使用特性，请参见[Java High Level REST Client官方文档](#)。

2.4. High Level REST Client (6.3.x)

本文基于Java High Level REST Client 6.3.x版本，为您介绍Elasticsearch Java API的用法。


准备工作

- 安装Java，要求JDK版本为1.8及以上。

安装方法请参见[安装JDK](#)。

- 创建阿里云Elasticsearch实例，版本要求大于等于Java High Level REST Client的版本。

本文创建一个6.3.2版本的实例，创建方法请参见[创建阿里云Elasticsearch实例](#)。

 **注意** High Level Client能够向上兼容，例如6.3.2版本的Java High Level REST Client能确保与大于等于6.3.2版本的Elasticsearch集群通信。为了保证最大程度地使用最新版客户端的特性，推荐High Level Client版本与集群版本一致。

- 开启阿里云Elasticsearch实例的自动创建索引功能。

具体操作步骤请参见[开启自动创建索引](#)。

如果未开启会提示如下报错。

```
IndexNotFoundException[no such index and [action.auto_create_index] contains [-*] which forbids automatic creation of the index]
```

- 配置阿里云Elasticsearch实例的白名单，确保网络互通。
 - 如果运行Java代码的服务器在公网环境下，可通过阿里云Elasticsearch实例的公网地址进行连通。连通前，需要开启阿里云Elasticsearch实例的公网地址，并修改公网地址访问白名单，将服务器的公网IP地址加入白名单中。具体操作步骤请参见[配置ES公网或私网访问白名单](#)。

 **注意**

- 如果您使用的是WIFI、宽带等网络，需要将公网出口的跳板机IP地址配置进去。建议您通过[淘宝IP地址库](#)查询。
- 您也可以将白名单配置为0.0.0.0/0，允许所有IPv4地址访问阿里云Elasticsearch实例。此配置会导致实例完全暴露在公网中，增加安全风险，配置前请确认您是否可以接受这个风险。

- 如果运行Java代码的服务器与阿里云Elasticsearch实例在同一专有网络VPC（Virtual Private Cloud）中，可通过阿里云Elasticsearch实例的内网地址进行连通。连通前，需要确保VPC私网访问白名单（默认为0.0.0.0/0）中已添加了服务器的内网IP地址。
- 创建Java Maven工程，并将如下的pom依赖添加到Java工程的pom.xml文件中。

pom依赖


```
<dependency>
  <groupId>org.elasticsearch</groupId>
  <artifactId>elasticsearch</artifactId>
  <version>6.3.2</version>
</dependency>
<dependency>
  <groupId>org.elasticsearch.client</groupId>
  <artifactId>elasticsearch-rest-high-level-client</artifactId>
  <version>6.3.2</version>
</dependency>
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-core</artifactId>
  <version>2.7</version>
</dependency>
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-api</artifactId>
  <version>2.7</version>
</dependency>
```

示例

单击下载[完整示例代码](#)。

以下示例使用Index API创建索引，然后使用Delete API删除该索引。

```
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.elasticsearch.action.delete.DeleteRequest;
import org.elasticsearch.action.delete.DeleteResponse;
import org.elasticsearch.action.index.IndexRequest;
import org.elasticsearch.action.index.IndexResponse;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import java.io.IOException;
import java.util.HashMap;
```

```
import java.util.Map;
public class RestClientTest63 {
    public static void main(String[] args) {
        // 阿里云Elasticsearch集群需要basic auth验证。
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        //访问用户名和密码为您创建阿里云Elasticsearch实例时设置的用户名和密码，也是Kibana控制台的登录用户名和密码。
        credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials("{访问用户名}",
        "{访问密码}"));
        // 通过builder创建rest client，配置http client的HttpClientConfigCallback。
        // 单击所创建的Elasticsearch实例ID，在基本信息页面获取公网地址，即为ES集群地址。
        RestClientBuilder builder = RestClient.builder(new HttpHost("{ES集群地址}", 9200))
            .setHttpClientConfigCallback(new RestClientBuilder.HttpClientConfigCallback() {
                @Override
                public HttpAsyncClientBuilder customizeHttpClient(HttpAsyncClientBuilder httpClientBuilder) {
                    return httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
                }
            });
        // RestHighLevelClient实例通过REST low-level client builder进行构造。
        RestHighLevelClient highClient = new RestHighLevelClient(builder);
        try {
            // 创建request。
            Map<String, Object> jsonMap = new HashMap<>();
            // field_01、field_02为字段名，value_01、value_02为对应的值。
            jsonMap.put("{field_01}", "{value_01}");
            jsonMap.put("{field_02}", "{value_02}");
            //index_name为索引名称；type_name为类型名称；doc_id为文档的id。
            IndexRequest indexRequest = new IndexRequest("{index_name}", "{type_name}", "{doc_id}").source(jsonMap);
            // 同步执行。
            IndexResponse indexResponse = highClient.index(indexRequest);
            long version = indexResponse.getVersion();
            System.out.println("Index document successfully! " + version);
            //index_name为索引名称；type_name为类型名称；doc_id为文档的id。与以上创建索引的名称和id相同。
            DeleteRequest request = new DeleteRequest("{index_name}", "{type_name}", "{doc_id}");
            DeleteResponse deleteResponse = highClient.delete(request);
            System.out.println("Delete document successfully!");
            highClient.close();
        } catch (IOException ioException) {
            // 异常处理。
        }
    }
}
```

```
}  
}
```

以上示例代码中带 `{}` 的参数需要替换为您具体业务的参数，详情请参见代码注释。

更多Java High Level REST Client的使用特性，请参见[Java High Level REST Client官方文档](#)。

2.5. Low Level REST Client (5.x)

本文基于Java Low Level REST Client 5.x版本，为您介绍Elasticsearch Java API的用法。

注意事项

- 本文提供的Low Level REST Client示例主要适用于阿里云Elasticsearch 5.5.3版本，不适用于6.3.2版本。如果您的Elasticsearch实例是6.3.2版本，可参见Elasticsearch [Java REST Client 6.3.2](#)官方文档进行配置。
- Java REST Client版本需要与Elasticsearch实例版本保持一致。

准备工作

- 安装Java，要求JDK版本为1.8及以上。
安装方法请参见[安装JDK](#)。
- 创建阿里云Elasticsearch实例，版本为5.5.3。
创建方法请参见[创建阿里云Elasticsearch实例](#)。
- 开启阿里云Elasticsearch实例的自动创建索引功能。
具体操作步骤请参见[开启自动创建索引](#)。

如果未开启会提示如下报错。

```
IndexNotFoundException[no such index and [action.auto_create_index] contains [-*] which forbids automatic creation of the index]
```

- 配置阿里云Elasticsearch实例的白名单，确保网络互通。
 - 如果运行Java代码的服务器在公网环境下，可通过阿里云Elasticsearch实例的公网地址进行连通。连通前，需要开启阿里云Elasticsearch实例的公网地址，并修改公网地址访问白名单，将服务器的公网IP地址加入白名单中。具体操作步骤请参见[配置ES公网或私网访问白名单](#)。

注意

- 如果您使用的是WIFI、宽带等网络，需要将公网出口的跳板机IP地址配置进去。建议您通过[淘宝IP地址库查询](#)。
- 您也可以将白名单配置为0.0.0.0/0，允许所有IPv4地址访问阿里云Elasticsearch实例。此配置会导致实例完全暴露在公网中，增加安全风险，配置前请确认您是否可以接受这个风险。

- 如果运行Java代码的服务器与阿里云Elasticsearch实例在同一专有网络VPC（Virtual Private Cloud）中，可通过阿里云Elasticsearch实例的内网地址进行连通。连通前，需要确保VPC私网访问白名单（默认为0.0.0.0/0）中已添加了服务器的内网IP地址。
- 创建Java Maven工程，并将下文的添加到Java工程的pom.xml文件中。

pom依赖

```
<dependency>
  <groupId>org.elasticsearch.client</groupId>
  <artifactId>rest</artifactId>
  <version>5.5.3</version>
</dependency>
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-core</artifactId>
  <version>2.7</version>
</dependency>
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-api</artifactId>
  <version>2.7</version>
</dependency>
```

示例

单击下载[完整示例代码](#)。

通过Java REST Client 访问阿里云Elasticsearch的9200端口进行测试，示例代码如下。

```
import org.apache.http.HttpEntity;
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.entity.ContentType;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.entity.NStringEntity;
import org.apache.http.util.EntityUtils;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import java.io.IOException;
import java.util.Collections;

public class RestClientTest55 {
    public static void main(String[] args){
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY,
            new UsernamePasswordCredentials("USER NAME", "PASSWORD"));
    }
}
```

```
// 单击所创建的Elasticsearch实例ID，在基本信息页面获取公网地址，即为HOST。
RestClient restClient = RestClient.builder(new HttpHost("HOST", 9200))
    .setHttpClientConfigCallback(new RestClientBuilder.HttpClientConfigCallback() {
        @Override
        public HttpAsyncClientBuilder customizeHttpClient(HttpAsyncClientBuilder httpClientBuilder) {
            return httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
        }
    }).build();
try {
    //field_01、field_02为字段名，value_01、value_02为对应的值。
    HttpEntity entity = new NStringEntity("{\n\"field_01\": \"value_01\"\n,\n\"field_02\": \"value_02\"\n}",
    ContentType.APPLICATION_JSON);
    //index_name为索引名称；type_name为类型名称；doc_id为文档的id。
    Response indexResponse = restClient.performRequest(
        "PUT",
        "/index_name/type_name/doc_id",
        Collections.<String, String>emptyMap(),
        entity);
    //index_name为索引名称；type_name为类型名称；doc_id为文档的id。与以上创建索引的名称和id相同。
    Response response = restClient.performRequest("GET", "/index_name/type_name/doc_id",
        Collections.singletonMap("pretty", "true"));
    System.out.println(EntityUtils.toString(response.getEntity()));
} catch (IOException e) {
    e.printStackTrace();
}
}
```

参数	说明
USER NAME	替换为访问阿里云Elasticsearch实例的用户名。
PASSWORD	替换为访问阿里云Elasticsearch实例的密码。
HOST	替换为阿里云Elasticsearch实例的内网或外网地址。可在实例的基本信息页面获取，获取方法请参见 查看实例的基本信息 。

2.6. Transport Client (5.x)

Transport Client 访问es

您可以通过Transport Client 5.3.3访问阿里云Elasticsearch 5.x版本的实例。本文介绍相关配置说明。

注意事项

- 建议您使用5.3.3版本的Transport Client来访问Elasticsearch集群。使用5.5或5.6版本访问Elasticsearch集群会提示NoNodeAvailableException的错误。
- 由于Transport Client通过TCP与Elasticsearch进行通信，因此当客户端与不同版本的Elasticsearch通信时，会存在兼容性问题，所以官方在高版本集群中已弃用Transport Client。建议优先使用[Java Low Level REST Client](#)来访问Elasticsearch集群，以保障版本的兼容性。

准备工作

- 安装Java，要求JDK版本为1.8及以上。
安装方法请参见[安装JDK](#)。
 - 创建阿里云Elasticsearch实例，版本为5.5.3。
创建方法请参见[创建阿里云Elasticsearch实例](#)。
 - 开启阿里云Elasticsearch实例的自动创建索引功能。
具体操作步骤请参见[开启自动创建索引](#)。
- 如果未开启会提示如下报错。

```
IndexNotFoundException[no such index and [action.auto_create_index] contains [-*] which forbids automatic creation of the index]
```

- 配置阿里云Elasticsearch实例的白名单，确保网络互通。
 - 如果运行Java代码的服务器在公网环境下，可通过阿里云Elasticsearch实例的公网地址进行连通。连通前，需要开启阿里云Elasticsearch实例的公网地址，并修改公网地址访问白名单，将服务器的公网IP地址加入白名单中。具体操作步骤请参见[配置ES公网或私网访问白名单](#)。

注意

- 如果您使用的是WIFI、宽带等网络，需要将公网出口的跳板机IP地址配置进去。建议您通过[淘宝IP地址库](#)查询。
- 您也可以将白名单配置为0.0.0.0/0，允许所有IPv4地址访问阿里云Elasticsearch实例。此配置会导致实例完全暴露在公网中，增加安全风险，配置前请确认您是否可以接受这个风险。

- 如果运行Java代码的服务器与阿里云Elasticsearch实例在同一专有网络VPC（Virtual Private Cloud）中，可通过阿里云Elasticsearch实例的内网地址进行连通。连通前，需要确保VPC私网访问白名单（默认为0.0.0.0/0）中已添加了服务器的内网IP地址。
- 创建Java Maven工程，并将下文的pom依赖添加到Java工程的pom.xml文件中。

pom依赖

```
<repositories>
  <!-- add the elasticsearch repo -->
  <repository>
    <id>elasticsearch-releases</id>
    <url>https://artifacts.elastic.co/maven</url>
    <releases>
      <enabled>>true</enabled>
    </releases>
    <snapshots>
      <enabled>>false</enabled>
    </snapshots>
  </repository>
</repositories>
<dependencies>
  <dependency>
    <groupId>org.elasticsearch.client</groupId>
    <artifactId>x-pack-transport</artifactId>
    <version>5.3.3</version>
  </dependency>
  <dependency>
    <groupId>org.elasticsearch</groupId>
    <artifactId>elasticsearch</artifactId>
    <version>5.3.3</version>
  </dependency>
  <dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-api</artifactId>
    <version>2.7</version>
  </dependency>
  <dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-core</artifactId>
    <version>2.7</version>
  </dependency>
</dependencies>
```

示例

单击下载[完整示例代码](#)。

```
import org.elasticsearch.action.get.GetResponse;
import org.elasticsearch.action.index.IndexResponse;
import org.elasticsearch.client.transport.TransportClient;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.transport.InetSocketTransportAddress;
import org.elasticsearch.xpack.client.PreBuiltXPackTransportClient;
import static org.elasticsearch.common.xcontent.XContentFactory.*;
import java.net.InetAddress;
public class TransportClientDemo {
    public static void main(String[] args) {
        try {
            TransportClient client = new PreBuiltXPackTransportClient(Settings.builder()
                .put("cluster.name", "es-cn-n6w1rux8i000w****") //替换为对应阿里云ES实例的ID。
                .put("xpack.security.user", "elastic:es_password") //阿里云ES实例的用户名、密码。
                .put("client.transport.sniff", false) //设置sniff为false。
                .build()
                .addTransportAddress(new InetSocketTransportAddress(InetAddress.getByName("es-cn-n6w1rux
                8i000w****.public.elasticsearch.aliyuncs.com"), 9300)); //指定域名及端口，替换为对应阿里云ES实例的域名。
            //下面这段代码可以根据实际情况进行修改，指定index，type及文档id。
            IndexResponse idxResp3 = client.prepareIndex("test_index", "test_type", "333")
                .setSource(jsonBuilder()
                    .startObject()
                    .field("user_id", "333")
                    .field("email", "test@aliyun.com")
                    .endObject()
                )
                .get();
            System.out.println(idxResp3.toString());
            GetResponse getResp =
                client.prepareGet().setIndex("test_index").setType("test_type").setId("333").execute().get();
            System.out.println(getResp.getSourceAsString());
            client.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```


参数	描述
<code>cluster.name</code>	阿里云Elasticsearch实例的ID, 可在实例的基本信息页面获取, 获取方法请参见 查看实例的基本信息 。
<code>client.transport.sniff</code>	嗅探, 请设置为false。
<code>xpack.security.user</code>	阿里云Elasticsearch实例的用户名和密码。
<code>InetAddress.getByName()</code>	<p>请在此方法的参数中指定访问阿里云Elasticsearch实例的域名及端口号。</p> <p> 注意 域名不能加http。</p>

3.通过客户端访问阿里云Elasticsearch

本文介绍使用PHP、Python和Java客户端访问阿里云Elasticsearch的方法，并提供了示例代码和注意事项供您参考。

准备工作

在使用客户端访问阿里云Elasticsearch实例前，您需要先完成以下准备工作：

- 安装Java，要求JDK版本为1.8及以上。
安装方法请参见[安装JDK](#)。
- 创建阿里云Elasticsearch实例。
具体操作步骤请参见[创建阿里云Elasticsearch实例](#)。
- 开启阿里云Elasticsearch实例的自动创建索引功能。
具体操作步骤请参见[开启自动创建索引](#)。
- 配置阿里云Elasticsearch实例的白名单，确保网络互通。
 - 如果运行代码的服务器在公网环境下，可通过阿里云Elasticsearch实例的公网地址进行连通。连通前，需要开启阿里云Elasticsearch实例的公网地址，并修改公网地址访问白名单，将服务器的公网IP地址加入白名单中。具体操作步骤请参见[配置ES公网或私网访问白名单](#)。

注意

- 如果您使用的是WIFI、宽带等网络，需要将公网出口的跳板机IP地址配置进去。建议您通过[淘宝IP地址库](#)查询。
- 您也可以将白名单配置为0.0.0.0/0，允许所有IPv4地址访问阿里云Elasticsearch实例。此配置会导致实例完全暴露在公网中，增加安全风险，配置前请确认您是否可以接受这个风险。

- 如果运行代码的服务器与阿里云Elasticsearch实例在同一专有网络VPC（Virtual Private Cloud）中，可通过阿里云Elasticsearch实例的内网地址进行连通。连通前，需要确保VPC私网访问白名单（默认为0.0.0.0/0）中已添加了服务器的内网IP地址。

PHP语言

警告

- Elasticsearch的PHP客户端提供的默认连接池并不适合云上环境。阿里云Elasticsearch提供了负载均衡的域名服务，因此PHP客户端访问程序必须使用SimpleConnectionPool作为连接池，否则在触发阿里云Elasticsearch重启操作时会出现访问连接异常的问题。
- PHP客户端访问程序必须具备访问连接失败重连的机制。因为在PHP客户端访问程序已使用SimpleConnectionPool作为连接池后，不排除在触发阿里云Elasticsearch重启操作时还会出现访问连接异常问题（例如提示No enabled connection），因此PHP客户端需要具备访问连接失败重连的机制。

通过PHP客户端访问阿里云Elasticsearch的9200端口进行测试，示例代码如下，详情请参见[elasticsearch-php](#)。

```
<?php
require 'vendor/autoload.php';
```

```
use Elasticsearch\ClientBuilder;
$client = ClientBuilder::create()->setHosts([
    [
        'host' => '<HOST>',
        'port' => '9200',
        'scheme' => 'http',
        'user' => '<USER NAME>',
        'pass' => '<PASSWORD>'
    ]
])->setConnectionPool('\Elasticsearch\ConnectionPool\SimpleConnectionPool', [])
->setRetries(10)->build();
$indexParams = [
    'index' => 'my_index',
    'type' => 'my_type',
    'id' => '1',
    'body' => ['testField' => 'abc'],
    'client' => [
        'timeout' => 10,
        'connect_timeout' => 10
    ]
];
$indexResponse = $client->index($indexParams);
print_r($indexResponse);
$searchParams = [
    'index' => 'my_index',
    'type' => 'my_type',
    'body' => [
        'query' => [
            'match' => [
                'testField' => 'abc'
            ]
        ]
    ],
    'client' => [
        'timeout' => 10,
        'connect_timeout' => 10
    ]
];
$searchResponse = $client->search($searchParams);
print_r($searchResponse);
?>
```

参数	说明
host	阿里云Elasticsearch实例 基本信息 页面中的内网或外网地址。
user	访问阿里云Elasticsearch实例的用户名，默认为elastic。
pass	访问阿里云Elasticsearch实例的密码。elastic用户的密码在创建实例时设定，如果忘记可进行重置，重置密码的注意事项和操作步骤请参见 重置实例访问密码 。

Python语言


通过Python客户端访问阿里云Elasticsearch的9200端口进行测试，示例代码如下，详情请参见[elasticsearch-py](#)。

```
from elasticsearch import Elasticsearch, RequestsHttpConnection
import certifi
es = Elasticsearch(
    ['<HOST>'],
    http_auth=('username', 'password'),
    port=9200,
    use_ssl=False
)
res = es.index(index="my_index", doc_type="my_type", id=1, body={"title": "One", "tags": ["ruby"]})
res = es.get(index="my_index", doc_type="my_type", id=1)
print(res['_source'])
```

参数	说明
<HOST>	阿里云Elasticsearch实例 基本信息 页面中的内网或外网地址。
username	访问阿里云Elasticsearch实例的用户名，默认为elastic。
password	访问阿里云Elasticsearch实例的密码。elastic用户的密码在创建实例时设定，如果忘记可进行重置，重置密码的注意事项和操作步骤请参见 重置实例访问密码 。

Java语言

Java客户端包括3种类型：Transport Client、Low Level REST Client和High Level REST Client，各类型的代码示例请参见[Java API](#)章节。

 **注意** 由于Java Transport Client通过TCP与Elasticsearch进行通信，因此当客户端与不同版本的Elasticsearch通信时，会存在兼容性问题，所以官方在高版本集群中已弃用Transport Client。如果您已经创建了5.5或5.6版本的Elasticsearch集群，分别使用Transport Client 5.5或5.6版本与Elasticsearch建立连接时会提示NoNodeAvailableException的错误。推荐使用Transport Client 5.3.3或[Java Low Level REST Client](#)来访问Elasticsearch集群，以保障版本的兼容性。如果使用Transport Client，需要在代码中将 `client.transport.sniff` 设置为false。

Go语言

通过Go语言访问阿里云Elasticsearch的9200端口进行测试，示例代码如下。

使用Go语言连接阿里云Elasticsearch前，您需要先安装Go编译环境，详情请参见[官方文档](#)。以下示例代码使用Go 1.13.4版本。

```
package main
import (
    "context"
    "gopkg.in/olivere/elastic.v5"
    "gopkg.in/olivere/elastic.v5/config"
    "log"
)
const (
    url = "http://es-cn-xxxxxx.public.elasticsearch.aliyuncs.com:9200" //<1>
    username = "elastic" //<2>
    password = "password" //<3>
)
func main() {
    var sniff = false //<4>
    cfg := &config.Config{
        URL: url,
        Username: username,
        Password: password,
    }
    cfg.Sniff = &sniff
    var client, err = elastic.NewClientFromConfig(cfg)
    if err != nil {
        log.Println(err)
        return
    }
    exists, err := client.IndexExists("index_test").Do(context.Background()) //<5>
    if err != nil {
        log.Println(err)
    }
    log.Println(exists)
}
```

参数	说明
<1> : url	阿里云Elasticsearch的访问域名，可在实例的 基本信息 页面获取。
<2> : username	阿里云Elasticsearch的访问账号，默认为elastic。
<3> : password	阿里云Elasticsearch的访问密码。elastic用户的密码在创建实例时设定，如果忘记可进行重置，重置密码的注意事项和操作步骤请参见 重置实例访问密码 。

参数	说明
<4>: sniff	必须设置为false, 防止二次探测, 导致连接失败。
<5>: exists, err	在阿里云Elasticsearch上新建一个索引index_test (需要替换为您的索引名称), 并根据连接状态返回日志信息。

连接成功后, 返回如下结果。

```
2019/11/29 10:07:37 true
```


更多语言示例请参见[HTTP/REST Clients and Security](#)。

4.通过API操作阿里云Elasticsearch

开源Elasticsearch提供了一系列Restful风格的API，您可以通过curl命令使用，也可以在Kibana中使用。本文介绍如何通过curl命令，调用API与阿里云Elasticsearch实例进行交互，并完成查看集群信息、创建索引和文档、搜索文档等操作。

前提条件

创建一个与阿里云Elasticsearch在同一专有网络VPC（Virtual Private Cloud）下的云服务器ECS实例。具体操作，请参见[使用向导创建实例](#)。


 **注意** 您也可以使用已创建的ECS实例，但需确保与Elasticsearch实例在相同VPC下。如果ECS处于经典网络下，期望访问VPC内的Elasticsearch，需要先参考[通过经典网络访问ES常见问题](#)进行相关配置。

背景信息

本文介绍通过curl命令使用API的方法，Kibana方式请参见[登录Kibana控制台](#)。

访问实例

1. 连接ECS实例。具体操作，请参见[步骤三：连接ECS实例](#)。
2. 使用如下命令，访问阿里云Elasticsearch实例。

 **说明** 如果系统提示 `curl command not found`，请先使用 `yum install curl` 命令，在ECS中安装curl。

```
curl -u <user>:<password> http://<host>:<port>
```

变量名	说明
<code><user></code>	阿里云Elasticsearch实例的访问用户名。建议通过非elastic账号访问。 <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  注意 <ul style="list-style-type: none"> 支持通过elastic账号访问，但因为在修改elastic账号的密码后需要一些时间来生效，在密码生效期间会影响服务访问，因此不建议通过elastic账号来访问。建议在Kibana控制台中创建一个符合预期的Role角色用户进行访问，具体操作请参见创建角色和创建用户。 如果您创建的阿里云Elasticsearch实例的版本中包含with_X-Pack信息，则访问该实例时，必须指定用户名和密码。 </div>
<code><password></code>	对应账号的密码。elastic账号的密码在创建实例时设定，如果忘记可重置。重置密码的注意事项和具体操作，请参见 重置实例访问密码 。
<code><host></code>	实例的内网地址。可在实例的 基本信息 页面获取。

变量名	说明
<port>	实例的访问端口。一般为9200，可在实例的 基本信息 页面获取。

访问示例如下。

```
curl -u <user>:<password> http://es-cn-vxxxxx****.elasticsearch.aliyuncs.com:9200
```

访问成功后，返回结果如下。

```
root@i57z5Z ~]# curl -u elastic: [REDACTED] http://es-cn-v [REDACTED] p.elasticsearch.aliyuncs.com:9200
{
  "name": "[REDACTED]",
  "cluster_name": "es-cn-v [REDACTED] p",
  "cluster_uuid": "X [REDACTED] w",
  "version": {
    "number": "6.7.0",
    "build_flavor": "default",
    "build_type": "tar",
    "build_hash": "8453f77",
    "build_date": "2019-03-21T15:32:29.844721Z",
    "build_snapshot": false,
    "lucene_version": "7.7.0",
    "minimum_wire_compatibility_version": "5.6.0",
    "minimum_index_compatibility_version": "5.0.0"
  },
  "tagline": "You Know, for Search"
}
```

查看集群信息

- 查看集群健康状况

```
curl -u <user>:<password> -XGET 'http://es-cn-vxxxxx****.elasticsearch.aliyuncs.com:9200/_cat/health?v'
```

执行成功后，返回如下结果。

```
root@izl ~]# curl -u elastic: [REDACTED] -XGET 'http://es-cn-45 [REDACTED] p5.elasticsearch.aliyuncs.com:9200/_cat/health?v'
epoch timestamp cluster status node.total node.data shards pri relo init unassign pending_tasks max_task_wait_time active_shards_percent
1583490872 10:34:32 es-cn-45 green 3 3 30 15 0 0 0 0 0 0 100.0%
```

- 查看集群中包含的索引信息

```
curl -u <user>:<password> -XGET 'http://es-cn-vxxxxx****.elasticsearch.aliyuncs.com:9200/_cat/indices?v'
```

执行成功后，返回如下结果。

```
[root@izl ~]# curl -u elastic: [REDACTED] -XGET 'http://es-cn-45 [REDACTED] .elasticsearch.aliyuncs.com:9200/_cat/indices?v'
health status index uuid pri rep docs.count docs.deleted store.size pri.store.size
green open .monitoring-beats-6-2020.03.05 Qj6g07mn 1 1 6990 0 6.7mb 3.3mb
green open .security-6 OM8j5guF 1 1 4 0 59.6kb 29.8kb
green open .monitoring-kibana-6-2020.03.06 wXauleIa 1 1 3827 0 2.4mb 1.2mb
green open .monitoring-kibana-6-2020.03.05 7PTG4VW- 1 1 8640 0 5mb 2.5mb
green open .monitoring-beats-6-2020.03.04 2b1rORi7 1 1 154 0 504.5kb 231.3kb
green open heartbeat-6.8.5-2020.03.05 J5Op1iQ2 1 1 12421 0 2.4mb 1.2mb
green open heartbeat-6.8.5-2020.03.06 LHP37xXu 1 1 7654 0 1.4mb 755kb
green open heartbeat-6.8.5-2020.03.04 XfP2PPUA 1 1 394 0 381.5kb 182.3kb
green open .monitoring-es-6-2020.03.05 gU5FC1BN 1 1 445865 198 244.8mb 122.5mb
green open .kibana_task_manager -rU0pt6e 1 1 2 0 25.8kb 12.9kb
green open .monitoring-kibana-6-2020.03.04 eVtqi5W9 1 1 5077 0 3.2mb 1.6mb
green open .monitoring-beats-6-2020.03.06 w-17k1WB 1 1 4463 0 4.2mb 2.1mb
green open .monitoring-es-6-2020.03.06 h1PRfwZY 1 1 267918 180 143.6mb 70.2mb
green open .monitoring-es-6-2020.03.04 Z4xQ3t_b 1 1 191991 70 110.1mb 55.2mb
green open .kibana_1 -HizoYCj 1 1 12 1 97.4kb 48.7kb
```

创建索引和文档

- 创建索引

```
curl -u <user>:<password> -XPUT 'http://es-cn-vxxxxx****.elasticsearch.aliyuncs.com:9200/product_info'
```

以上示例创建了一个名称为 `product_info` 的索引。

执行成功后，返回如下结果。

```
[root@iz... ~]# curl -u elastic: -XPUT 'http://es-cn-45...elasticsearch.aliyuncs.com:9200/product_info'
{"acknowledged":true,"shards_acknowledged":true,"index":"product_info"}[root@iz... ~]#
```

- 为索引设置mapping

```
curl -u <user>:<password> -XPUT 'http://es-cn-vxxxxx****.elasticsearch.aliyuncs.com:9200/product_info/_doc/_mapping' -H 'Content-Type: application/json' -d '{
  "_doc":{
    "properties":{
      "productName":{"type":"text","analyzer":"ik_smart"},
      "annual_rate":{"type":"keyword"},
      "describe":{"type":"text","analyzer":"ik_smart"}
    }
  }
}'
```

执行成功后，返回如下结果。

```
[root@iz... ~]# curl -u elastic: -XPUT 'http://es-cn-4...elasticsearch.aliyuncs.com:9200/product_info/_doc/_mapping' -H 'Content-type: application/json' -d '{
  "_doc":{
    "properties": {
      "productName": {"type": "text", "analyzer": "ik_smart"},
      "annual_rate": {"type": "keyword"},
      "describe": {"type": "text", "analyzer": "ik_smart"}
    }
  }
}'
{"acknowledged":true}
```

以上示例设置 `product_info` 索引的类型为 `_doc`，包含了 `productName`、`annual_rate` 和 `describe` 字段，并定义了各字段的类型所使用的分词器。

- 创建文档并插入数据

o 创建单个文档

```
curl -u <user>:<password> -XPOST 'http://es-cn-vxxxxx****.elasticsearch.aliyuncs.com:9200/product_info/_doc/1?pretty' -H 'Content-Type: application/json' -d '{
  "productName":"testpro",
  "annual_rate":"3.22%",
  "describe":"testpro"
}'
```

执行成功后，返回结果如下。

```
[root@i2i ~]# curl -u elastic: -XPOST 'http://es-cn-45 .elasticsearch.aliyuncs.com:9200/product_info/_doc/1?pretty' -H 'Content-type: application/json' -d '{
  "productName":"testpro",
  "annual_rate":"3.22%",
  "describe":"testpro"
}'
{
  "_index": "product_info",
  "_type": "_doc",
  "_id": "1",
  "_version": 2,
  "result": "updated",
  "_shards": {
    "total": 2,
    "successful": 2,
    "failed": 0
  },
  "_seq_no": 1,
  "_primary_term": 1
}
```

以上示例在类型为 `_doc` 的 `product_info` 索引中，创建了一个名称为 `1` 的文档，并向文档中插入了一条数据。

o 创建多个文档

```
curl -u <user>:<password> -XPOST http://es-cn-vxxxxx****.elasticsearch.aliyuncs.com:9200/_bulk -H 'Content-Type: application/json' -d '{
  "index": { "_index": "product_info", "_type": "_doc", "_id": "1" } }
  {"productName":"testpro","annual_rate":"3.22%","describe":"testpro"}
  {"index": { "_index": "product_info", "_type": "_doc", "_id": "2" } }
  {"productName":"testpro1","annual_rate":"3.26%","describe":"testpro"}
```

执行成功后，返回结果如下。

```
[root@VM1 ~]# curl -u -XPOST http://es-cn-m6u1o1xw .elasticsearch.aliyuncs.com:9200/_bulk -H 'Content-type: application/json' -d '{
  "index": { "_index": "product_info", "_type": "_doc", "_id": "1" } }
  {"productName":"testpro","annual_rate":"3.22%","describe":"testpro"}
  {"index": { "_index": "product_info", "_type": "_doc", "_id": "2" } }
  {"productName":"testpro1","annual_rate":"3.26%","describe":"testpro"}
}'
{"took":117,"errors":false,"items":[{"_index":{"_index":"product_info","_type":"_doc","_id":"1","_version":1,"result":"created","_shards":{"total":2,"successful":2,"failed":0},"_seq_no":0,"_primary_term":1,"status":201},"_index":{"_index":"product_info","_type":"_doc","_id":"2","_version":1,"result":"created","_shards":{"total":2,"successful":2,"failed":0},"_seq_no":1,"_primary_term":1,"status":201}]}][root@VM1 ~]#
```

以上示例在类型为 `_doc` 的 `product_info` 索引中，创建了一个名称为 `1` 和 `2` 的文档，并分别向文档中插入了一条数据。

搜索文档

```
curl -u <user>:<password> -XGET 'http://es-cn-vxxxxx****.elasticsearch.aliyuncs.com:9200/product_info/_doc/1?pretty'
```

执行成功后，返回结果如下。

```
[root@iz1-xxxxxxx:~]# curl -u elastic:elastic -XGET 'http://es-cn-45-xxxxxxx.elasticsearch.aliyuncs.com:9200/product_info/_doc/1?pretty'  
{  
  "index" : "product_info",  
  "type" : "_doc",  
  "_id" : "1",  
  "version" : 2,  
  "seq_no" : 1,  
  "primary_term" : 1,  
  "found" : true,  
  "source" : {  
    "productName" : "testpro",  
    "annual_rate" : "3.22%",  
    "describe" : "testpro"  
  }  
}
```

以上示例搜索名称为 `1` 的文档。

删除索引

```
curl -u <user>:<password> -XDELETE 'http://es-cn-vxxxxx****.elasticsearch.aliyuncs.com:9200/product_info'
```

执行成功后，返回结果如下。

```
[root@iz1-xxxxxxx:~]# curl -u elastic:elastic -XDELETE 'http://es-cn-45-xxxxxxx.elasticsearch.aliyuncs.com:9200/product_info'  
{"acknowledged":true}
```

以上示例删除了名称为 `product_info` 的索引。

 **说明** 更多命令，请参见[Elasticsearch官方文档](#)。

5.Java API FAQ


本文介绍阿里云Elasticsearch（简称ES）Java API方面的常见问题。

使用Transport Client访问阿里云ES，其中cluster.name怎么获取？

cluster.name就是实例ID，可在实例的基本信息页面获取，详情请参见[查看实例的基本信息](#)。

使用Transport Client连接阿里云ES时，报错NoNodeAvailableException如何处理？

使用5.5或5.6版本的Transport Client与ES建立连接时会提示NoNodeAvailableException的错误，推荐您使用5.3.3版本。使用Transport Client，需要购买5.5或5.6版本的阿里云ES实例（6.x及以上版本不支持），并且需要在代码中将 `client.transport.sniff` 设置为false，详情请参见[Transport Client \(5.x\)](#)。

 **注意** ES 7.0中已经弃用Transport Client，在8.0中完全移除它。因此在实际开发中建议您使用Java REST Client，详情请参见[High Level REST Client \(6.3.x\)](#)、[High Level REST Client \(6.7.x\)](#)、[Low Level REST Client \(5.x\)](#)。