

# ElasticSearch系列文章:客户端

### 有多个地方需要使用Java client:

- 1、在存在的集群中执行标准的index, get, delete和search
- 2、在集群中执行管理任务
- 3、当你要运行嵌套在你的应用程序中的Elasticsearch的时候或者当你要运行单元测试或者集合测试的时候,启动所有节点

### 获得一个Client是非常容易的,最通用的步骤如下所示:

- 1、创建一个嵌套的节点,充当集群的一个节点
- 2、从这个嵌套的节点请求一个Client
- 3、另外一种方式是创建一个TransportClient来连接集群。

#### 重要提示:

客户端和集群端推荐使用相同的版本,如果版本不同,可能会出现一些不兼容的问题。

## 节点客户端

实例化一个节点的客户端是获得客户端的最简单的方式。这个Client可以执行elasticsearch相关的操作。

import static org.elasticsearch.node.NodeBuilder.\*;

// on startup
Node node = nodeBuilder().node();
Client client = node.client();

// on shutdown
node.close();

当你启动一个node,它就加入了elasticsearch集群。你可以通过简单的设置cluster.name或者明确地使用clusterName方法拥有不同的集群。

你能够在你项目的/src/main/resources/elasticsearch.yml文件中定义cluster.name。只要elasticsearch.yml在classpath目录下面,你就能够用到它来启动你的节点。或者通过java代码设置:

Node node = nodeBuilder().clusterName("yourclustername").node(); Client client = node.client();



利用Client的好处是,操作可以自动地路由到这些操作被执行的节点,而不需要执行双跳(do uble hop)。例如,索引操作将会在该操作最终存在的分片上执行。

当你启动了一个节点,最重要的决定是是否它将保有数据。大多数情况下,我们仅仅需要用到clients,而不需要分片分配给它们。这可以通过设置node.data为false或者设置node.client为true来简单实现。

import static org.elasticsearch.node.NodeBuilder.\*;

// on startup
Node node = nodeBuilder().client(true).node();
Client client = node.client();

// on shutdown
node.close();

另外一个通用的用处就是启动node,然后利用Client进行单元/集成测试。在这种情况下,我们应该启动一个"本地"node。这只是启动node的一个简单的设置。注意,"本地"表示运行在本地 JVM上。运行在同一个JVM上的两个本地服务可以彼此发现并组成一个集群。

import static org.elasticsearch.node.NodeBuilder.\*;

// on startup
Node node = nodeBuilder().local(true).node();
Client client = node.client();
// on shutdown
node.close();

# 传输 (transport) 客户端

TransportClient利用transport模块远程连接一个elasticsearch集群。它并不加入到集群中,只是简单的获得一个或者多个初始化的transport地址,并以轮询的方式与这些地址进行通信。

// on startup
Client client = new TransportClient()



.addTransportAddress(new InetSocketTransportAddress("host1", 9300))
.addTransportAddress(new InetSocketTransportAddress("host2", 9300));

// on shutdown
client.close();

注意,如果你有一个与elasticsearch集群不同的集群,你可以设置机器的名字。

你也可以用elasticsearch.yml文件来设置。

这个客户端可以嗅到集群的其它部分,并将它们加入到机器列表。为了开启该功能,设置client.tr ansport.sniff为true。

## 其它的transport客户端设置有如下几个:

Parameter Description

client.transport.ignore\_cluster\_name true:忽略连接节点的集群名验证 client.transport.ping\_timeout ping一个节点的响应时间,默认是5s client.transport.nodes\_sampler\_interval sample/ping 节点的时间间隔,默认是5s

本博客文章除特别声明,全部都是原创!

原创文章版权归过往记忆大数据(<u>过往记忆</u>)所有,未经许可不得转载。 本文链接:【】()