[一、基础概念 1](#_Toc1100)

[二、搭建Elasticsearch开发环境 1](#_Toc27545)

[三、通过RESTful API与Elasticsearch交互 2](#_Toc15160)

[四、通过JAVA客户端与Elasticsearch交互 3](#_Toc17462)

[五、Elasticsearch的扩容机制 3](#_Toc26717)

### 一、基础概念

1、什么是Elasticsearch：一个实时的分布式搜索分析引擎，可用于文档的分布式存储、字段索引搜索、全文搜索、结构化搜索和分析，并能够支持上百服务节点的扩展和PB级的数据。Elasticsearch建立于全文搜索引擎Lucene之上，使用Lucene做索引与搜索，并提供一套RESTful API。

2、Elasticsearch是面向文档的：Elasticsearch可以直接存储整个对象或是文档，而不必将对象或文档扁平化。Elasticsearch使用JSON作为文档额序列化格式。有别于高度结构化的关系型数据库，单个对象散布于多个表格中，文档数据库可以一次存储整个对象，并且每个对象可与其他被存储的对象不同。

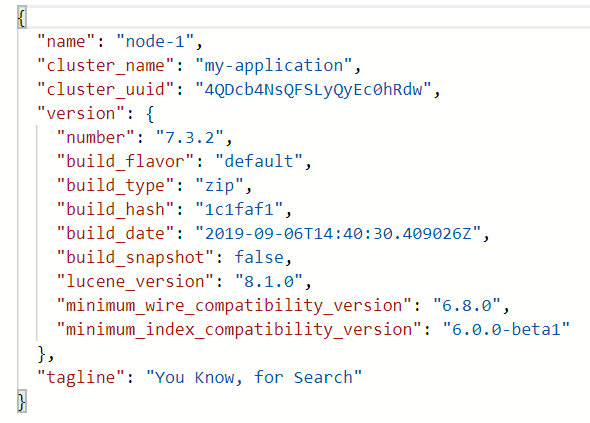
3、Elasticsearch的数据组织方式：Elasticsearch中的索引(Index)类似于关系型数据库中的一个数据库，使用倒排索引来达到提升检索速度的目的。Elasticsearch中的类型(Type)类似于关系型数据库中的表格。Elasticsearch中的文档(Document)类似于关系型数据库中的一个数据行，将其存入的行为被称作索引（动词）。Elasticsearch中的字段(Field)类似于关系型数据库中的列。

4、Elasticsearch的集群组织方式：在分布式环境下，Elasticsearch由集群(cluster)、节点(node)以及分片(shard)组成。一个运行中的实例被称为一个节点，一个或多个拥有相同cluster.name配置的节点组成集群。新加入节点或是删除节点均会导致集群重新分配所有数据。集群中的主节点负责索引和其他节点的增删操作，不会涉及文档级别的数据操作和搜索。

### 二、搭建Elasticsearch开发环境

1、执行bin/elasticsearch -d，后台运行Elasticsearch

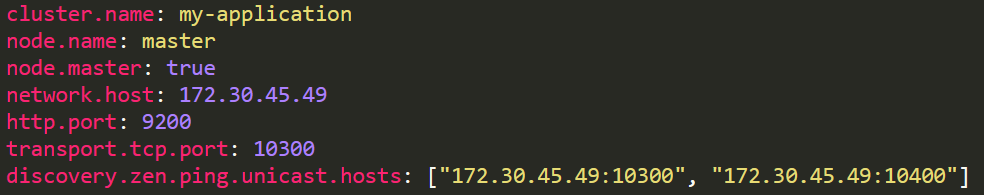
2、访问http://localhost:9200/?pretty，测试Elasticsearch是否启动成功，若成功启动，可以收到类似结构的信息：



可以在conf/elasticsearch.yml中配置相关信息。一个Elasticsearch集群中的节点拥有相同的cluster\_name。

1. 为集群添加故障转移。

在同一台机器上Elasticsearch默认使用组播的方式发现新节点，集群中使用单播方式发现节点。通过修改elasticsearch.yml中的discovery.zen.ping.unicast.hosts可以修改单播列表，新节点连接单播列表中的任一成员均可获取到集群的整体状态和节点信息，之后新节点会联系master节点，并加入集群。集群的最小配置字段如下：



同一集群内节点的cluster.name字段应该相同，Elasticsearch启动需要绑定两个端口（9200与9300），一个负责处理http请求，另一个负责处理来自客户端与集群其他节点的tcp连接。

### 三、通过RESTful API与Elasticsearch交互

1、Curl请求示例

默认使用9200端口与Elasticsearch通信，请求示例：

curl -XGET -H "Content-Type: application/json" http://localhost:9200/\_count?pretty -d '{ "query": {"match\_all":{} } }'

向Elasticsearch插入一条数据，请求示例：

curl -X PUT -H 'Content-Type: application/json' "localhost:9200/megacorp/employee/1?pretty" -d' { "first\_name" : "John", "last\_name" : "Smith", "age" : 25, "about" : "I love to go rock climbing", "interests": [ "sports", "music" ] } '

其中megacorp为索引名，employee为类型名称，1为文档ID。

根据文档ID请求一条数据，请求示例：

curl -X GET "localhost:9200/megacorp/employee/1?pretty"

通过更换请求method为DELETE，可以删除指定数据，请求示例：

curl -X DELETE "localhost:9200/megacorp/employee/1?pretty"

可以通过使用PUT或POST方法对已存在的文档进行更新操作，请求示例：

curl -X PUT -H 'Content-Type: application/json' "localhost:9200/megacorp/employee/1?pretty" -d '{ "first\_name" : "John"}'

Elasticsearch在执行更新操作时，会先将旧文档标记为删除状态，然后添加新文档。

通过在请求路径中使用\_search端点，并配合q参数可以实现具体到字段的轻量搜索，请求示例：

curl -X GET "localhost:9200/megacorp/employee/\_search?q=last\_name:Smith&pretty"

也可以使用JSON构造一个更复杂的查询请求：

curl -X GET -H "Content-Type: application/json" "localhost:9200/megacorp/employee/\_search" -d'

{"query": {"bool": {"must": [{ "match": { "last\_name": "Smith" }},{ "match": { "age": 25 }}]}}}'

curl -X GET "localhost:9200/megacorp/employee/\_search?pretty" -H 'Content-Type: application/json' -d'

{"query" : {"bool": {"must": {"match" : {"last\_name" : "smith"}},"filter": {"range" : {"age" : { "gt" : 30 }}}}}}'

2、

### 四、通过JAVA客户端与Elasticsearch交互

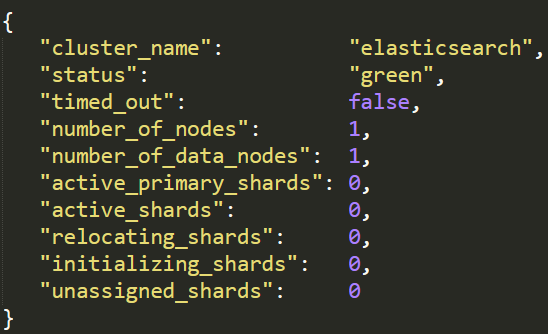
### 五、Elasticsearch的存储与扩容机制

1、查看集群的健康信息

请求示例：

curl -X GET "localhost:9200/\_cluster/health?pretty"

返回内容如下：

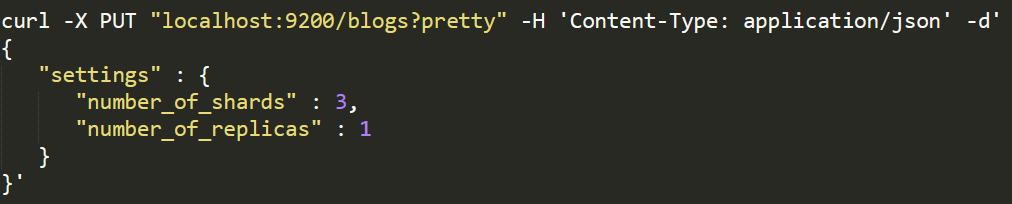


其中status字段表示集群的运行状态：green表示所有主分片和副本分片都正常运行，yellow表示所有主分片和部分副本分片正常运行，red表示部分主分片没有正常运行。

2、Elasticsearch的分片机制

分片是Elasticsearch的一个底层工作单元，仅保存一部分数据，一个分片即为一个完整的搜索引擎（一个Lucene实例）。当集群规模变化时，为了能够使数据均匀分布在集群里，Elasticsearch会迁移分片，利用分片将数据分发到集群各处。作为数据容器，当添加一个文档时，文档会被存储和索引到分片内。

分片可以被分为主分片和副本分片，副本分片只是主分片的拷贝，用于数据冗余备份以及提供数据读操作。在创建索引时，Elasticsearch会为索引分配多个主分片，每个主分片对应一个副本分片，可以通过如下方式在创建时指定索引的主分片数(number\_of\_shards)，以及每个主分片对应的副本分片数(number\_of\_replicas)：



同时也可以通过GET方法查看索引的状态信息。