ES核心概念和原理

1、什么是搜索：百度、垂直搜索（站内搜索）

搜索：通过一个**关键词**或一段描述，得到你想要的（相关度高）结果。

2、如何实现搜索功能?

关系型数据库：性能差、不可靠、结果不准确（相关度低）

3.倒排索引、Lucene和全文检索?

(1) 倒排索引的数据结构

数据结构：1、包含这个关键词的document list

2.关键词在每个doc中出现的次数 TF term frequency

3.关键词在整个索引中出现的次数 IDF inverse doc frequency

4.关键词在当前doc中出现的次数

5.每个doc的长度，越长相关度越低

6.包含这个关键词的所有doc的平均长度

(2) Lucene：jar包，帮我们创建倒排索引，提供了复杂的API

(3) 如果用Lucene做集群实现搜索，会有那些问题

1. 节点一旦宕机，节点数据丢失，后果不堪设想，可用性差。
2. 自己维护，麻烦（自己创建管理索引），单台节点的承载请求的能力是有限的，需要人工做负载（雨露均沾）。

4.Elasticsearch：**分布式、高性能、高可用、可伸缩、易维护** ES≠搜索引擎

(1) 分布式的**搜索，存储**和**数据分析**引擎：

(2) 优点：

1. 面向开发者友好，屏蔽了Lucene的复杂特性，集群自动发现（cluster discovery）
2. 自动维护数据在多个节点上的建立
3. 会帮我做搜索请求的负载均衡
4. 自动维护冗余副本，保证了部分节点宕机的情况下仍然不会有任何数据丢失
5. ES基于Lucene提供了很多高级功能：复合查询、聚合分析、基于地理位置等。
6. 对于大公司，可以构建几百台服务器的大型分布式集群，处理PB级别数据；对于小公司，开箱即用，门槛低上手简单。
7. 相遇传统数据库，提供了全文检索，同义词处理（美丽的cls>漂亮的cls），相关度排名。聚合分析以及海量数据的近实时（NTR）处理，这些传统数据库完全做不到。

(3) 应用领域：

1. 百度（全文检索、高亮、搜索推荐）
2. 各大网站的用户行为日志（用户点击、浏览、收藏、评论）
3. BI（Business Intelligence商业智能），数据分析：数据挖掘统

计。

4. Github：代码托管平台，几千亿行代码

5. ELK：Elasticsearch（数据存储）、Logstash（日志采集）、Kibana（可视化）

5.ES核心概念：

(1) cluster（集群）：每个集群至少包含两个节点.

(2) node：集群中的每个节点，一个节点不代表一台服务器

(3) field：一个数据字段，与index和type一起，可以定位一个doc

(4) document：ES最小的数据单元 Json

```
{
  "id": "1",
  "name": "小米",
  "price": {
    "标准版": 3999,
    "尊享版": 4999,
    "吴磊签名定制版": 19999
  }
}
```

Type：逻辑上的数据分类，es 7.x中删除了type的概念

Index：一类相同或者类似的doc，比如一个员工索引，商品索引。

Shard分片：

1：一个index包含多个Shard，默认5P，默认每个P分配一个R，P的数量在创建索引的时候设置，如果想修改，需要重建索引。

2：每个Shard都是一个Lucene实例，有完整的创建索引的处理请求能力。

3：ES会自动在nodes上为我们做shard 均衡。

4：一个doc是不可能同时存在于多个PShard中的，但是可以存在于多个RShard中。

5：P和对应的R不能同时存在于同一个节点，所以最低的可用配置是两个节点，互为主备。