# Getting started

ElasticSearch(以下简称es)是一个具有高扩展性的开源全文检索和分析引擎。它可以帮助你快速、近实时的存储、搜索和分析大批量数据。那些有复杂搜索特性和需求的应用程序通常将es作为底层技术引擎来提供动力。

以下是一些es使用的简单示范：

* 假如你在运营一家电商平台，你需要让用户在上面搜索商品。在这个场景下，你可以将所有商品的目录和库存都存在es中，并且为它们提供搜索和自动完成的建议。
* 如果你想要搜集日志和事务数据，并分析和挖掘它们来寻找其中隐藏的趋势、统计、概要和异常。在这个场景里，可以使用Logstash(ElasticSearch/Logstash/Kibana，ELK技术栈的一部分)来搜集、聚合并分析这些数据，然后用Logstash将它们投入到Es里，一旦es有了这些数据，你就可以通过搜索和聚合来挖掘你感兴趣的信息。
* 你在运营一个价格提醒平台，允许对价格敏感的用户定义一个规则，比如“我想买一个电子产品，如果在下个月内有任何卖家出售价格掉到X元以下，我希望能收到通知”。在这个场景中，你可以爬取卖家的价格，把它们推送到es里，并且使用反向搜索功能来匹配针对用户搜索的价格趋势，最重在匹配上后提醒用户。
* 你有分析/智能化业务的需求，并且希望快速调查、分析、将它们可视化，然后对可能高达数百万、数十亿记录的这些数据提出特别的问题。在这个场景下，你可以使用es来存储数据，并用kibana来构建定制化的仪表盘，以便对你重要数据的某些方面进行可视化。此外，你也可以对数据使用es的聚合功能来进行复杂的商务智能需求。

在教程的剩余部分，你将被引导完成Es的启动和运行，了解其中的内容，进行基本的类似索引[[1]](#footnote-0)、搜索、更新的操作。在完成教程后，你应该对es是什么，它如何工作有深刻理解，希望能见到你用它构建复杂的搜索程序，或是从数据中挖掘一些情报。

## Basic Concepts

下面是一些es的核心概念，在一开始就了解这些概念有助于简化学习过程。

### 近实时(Near Realtime NRT)

Es是一个近实时的搜索平台。这意味着从索引完一个文档到它可以被查询到，中间有一个短暂的时间间隔，通常是一秒。

### 集群(Cluster)

集群是由一个或多个节点（服务器）组成的集合体，它们一起保存你的整体数据，并联合对外提供索引和搜索功能。集群用一个唯一的名称作标识，默认是”elasticsearch”。这个名称很重要，因为节点只有通过设置集群的名字来加入集群，才能成为集群的一部分。

在不同的环境下，请确保不要重复使用相同的集群名称，否则可能导致节点加入错误的集群。比如你可以使用logging-dev, logging-stage和logging-prod来作为开发、稳定和生产环境的集群名。

请注意，只用一个节点作为集群也是可行的，并且也能很好的运行。此外也可以构建多个独立的集群，每个都有自己单独的名称。

### 节点(Node)

节点是单独一个服务器，作为集群的一部分存储数据、提供索引和搜索功能。它也像索引一样，有一个独特的名称标识，默认是一个随机的UUID，在启动时就已分配好。如果不想使用默认的名称也可以手动指定。这个名称对于管理来说很重要，它可以标识集群中服务器和节点的关系。

节点可以通过配置集群名来加入集群，默认每个节点都会加入名为elasticsearch的集群。这意味着如果在内网启动了一定数量的节点，并且使它们互相可见，它们将自动的加入一个名为elasticsearch的集群。

### 索引(index)

索引是一些有相似特征的文档的集合。举个例子，你可以用一个索引存储顾客信息，用另一个索引存储产品目录，还可以用一个索引存储订单信息。索引必须用英文小写字母作为名称标识，而且这个名称还可以用来指代索引，以便对其中的文档执行索引、查询、更新和删除操作。

在一个集群内，可以定义任意多个索引。

### 类型(type)

警告：在6.0.0版本已被标记为废弃，详情见”[Removal of mapping types](https://www.elastic.co/guide/en/elasticsearch/reference/current/removal-of-types.html" \l "removal-of-types)”

类型通常用于索引的逻辑分类/分区，以便在同一个索引中存储不同类型的文档，比如，一个类型是用户，另一个是博客文章。请注意，现版本中一个索引已经不可以创建多个类型，而且类型的概念也将在后续版本完全移除，详情见”[Removal of mapping types](https://www.elastic.co/guide/en/elasticsearch/reference/current/removal-of-types.html" \l "removal-of-types)”。

### 文档(document)

文档是可以被索引的基本信息单元。比如，你可以将某个顾客作为一个文档，将一个订单作为另一个。文档用JSON来标识，这是一种无处不在的的的网络数据交互格式。在一个索引内存储的文档数量不限。

### 分片&副本(Shards & Replicas)

一个索引有可能存储非常多的数据量，甚至超过单节点的硬件极限。比如，一个包含10亿文档的索引占用了1TB的硬盘空间，这时单个节点的磁盘空间可能有所不足，执行搜索请求时也可能过慢，导致超时等。

为了解决这个问题，es提供了将索引细分为多个片段的能力，这些片段称为分片。

当我们创建索引时，可以直接定义我们想要的分片数量。每个分片本身都是一个功能齐全且独立的“索引”，可以托管在集群中的任何节点上。

分片之所以重要，有以下2个主要原因：

* 可以水平切割/扩展内容分卷
* 可以跨分片分布和并行化操作，从而提升性能和吞吐量，即使分片可能在多个节点上。

分片如何分布、如何将分片中的文档聚合到搜索请求的机制完全由es管理，并且对用户透明。

在网络和云这种每时每刻都有可能出错的环境里，万一分片或节点因为某些理由离线或宕机，那么使用失效转移的机制来应对是非常有用的，也是强烈推荐的。为此，es将索引的分片复制一到多个，我们称之为副本分片，或者简称副本。

副本的重要性也有两个主要的原因：

* 提供高可用性以应对分片和节点故障。因此请注意，副本永远不会和复制他的原始/主分片分布在同一个节点上
* 由于搜索请求可以并行执行在所有副本上，因此可以扩展搜索量和吞吐量。

总之，每个索引都会切割为多个分片。一个索引可以没有副本，也可以有多个副本。一旦有了副本，索引就同时拥有主分片和副本分片，主分片指被复制的原始分片，副本分片指主分片的复制体。

每个索引在被创建的时候就可以指定分片和副本的数量。创建索引之后，我们也可以随时动态修改副本的数量。我们可以对已有的索引使用[\_shrink](https://www.elastic.co/guide/en/elasticsearch/reference/7.0/indices-shrink-index.html)和[\_split](https://www.elastic.co/guide/en/elasticsearch/reference/7.0/indices-split-index.html)等API来修改分片的数量，但这不是一个简单的任务，最好是一开始就计划好正确的分片数量。

默认，es中的每个索引分配1个主分片和一个副本，也就是说，如果在集群中至少有2个节点，那么每个索引将有一个主分片和另一个完整的副本分片，总共两个分片。

注意：每个es分片都是一个Lucene索引。单个Lucene索引有文档数量上限，在Lucene-5843中，该值为2,147,483,519(=Integer.MAX\_VALUE)。可以通过[\_cat/shards](https://www.elastic.co/guide/en/elasticsearch/reference/7.0/cat-shards.html) API来监控分片数量。

介绍完了概念，开始一些有趣的环节吧。

1. 在es中，索引(index) 既表示存储一个文档的行为（动词），又表示文档存储的逻辑集合点，类比数据库的表（名词），请注意根据语境区分。 [↑](#footnote-ref-0)