主要是在于es的版本选择，以及主要会用到的es内容。

## 版本选择

es版本更新算是比较快速的，其中主要接触的是2.3.5版本的内容，最新的稳定版本则是5.x版本，对于1.x及以下的内容就不再做考虑，对6.x以上的测试版，暂时也不会进行对比。

es5.x在2.x上的变化内容较多，参考：

<http://cwiki.apachecn.org/pages/viewpage.action?pageId=4260618>

选型的结果是es2.x的版本，原因有两点：

一是对于es2.x版本的操作熟悉一点，es5.x的内容还不太熟悉，其次就是2.x的kopf监控用的比较习惯，而5.x的监控在x-pack中，x-pack的安装目前都有着问题。

二是业务中对关联查询的需求，实现方法父子文档、es-sql、siren-join等，目前偏向于使用siren-join来完成关联查询的需求，但是该插件并没有5.x的版本，所以选型中还是优先考虑2.x的版本内容。

## settings设置

主要的设置有两个：

number\_of\_shards：每个索引的主分片数，默认值是5.

number\_of\_replicas：每个主分片的副本数，默认值是1.

可以通过直接put中设置settings属性，也可以通过\_settings API来设置。

其中，分片数是需要在一开始就要预估好的，副本数则可以随时修改。

## Mappings设置

使用es建索引的时候，mapping的设置尤其重要，根据需求设置合理的mapping，可以节省大量资源。

首先就是\_source字段与\_all字段的问题。

### \_source

\_source是传输文档的json数据。

禁用\_source字段带来的问题是update、update\_by\_query（这两种方法主要是通过脚本来实现内容的更新），reindex API不再可用，修改mapping、修改分词、自动修复index顺坏等将不可用。

可以在mapping中禁用\_source，也可以设置source所包含的字段内容以及排除的内容。

### \_all

\_all字段是一个大字符串数组，包含了文档的所有term，其结构应用在倒排中。允许我们在不知道field的情况下，进行查询搜索。

禁用\_all字段带来的问题是无法在不知field的情况下进行一些value搜索等操作。

### \_parent field

### \_routing field

Mapping中的设置主要就是是否设置type的routing值，确认了的话，文档导入的时候必须指定routing值，否则的话，还是以默认的文档id来作为路由值。

### 映射参数

#### analyzer分析器

设置mapping的时候，对字段进行设置的时候，进行”analyzer” :”english”，这种来设置字段的分析器，这个在之后设置中文分析器的时候比较多用到。

#### Boost

自动提升某个字段的权重，使得查询的时候返回结果位于前列，如在查看某个漏洞的时候，希望等保或者监测平台的信息位于前面，就可以通过boost提升某个字段的权重。但是建议还是将提升字段权重放在查询的时候做，放在mapping里设置会导致查询变慢，同时降低了质量相关性的计算。

#### Cocere

用于清除脏数据，否则整型数字5可能会被写成字符串”5”或者浮点数5.0，

Coerce属性可以用来清除脏数据：

字符串会被强制转换为整数、浮点数会被强制转换为整数。

#### Copy\_to

相当于创建自定义的\_all字段，也就是可以将多个字段的值复制到group field（组字段），然后可以作为单个字段进行查询。

#### Doc\_values(文档值)

用作聚合以及排序，如果确认字段不需要这两个操作，可以禁用doc\_values节省空间。

#### Dynamic(动态设置)

这个设置级别与\_source同级，如果为false，mappings设置了之后，索引不可以动态增加字段

#### Enabled（开启字段）

这个属性是用来那些只存储，但不检索也不聚合排序的字段

“field1”:{

“enabled”: false

}这样进行设置

#### Fielddata（字段数据）

es2.x中，分词字段默认启用这个内容，但是一旦用到它，会消耗很多的堆空间，加载fielddata是一个非常耗费资源的过程。

#### Fotmat（日期格式）

#### Include\_in\_all（\_all查询包含字段）

默认都是开启的，在字段mapping中设置include\_in\_all为false，