本文记录一次Elasticsearch mapping field修改过程  
  
  
团队使用Elasticsearch做日志的分类检索分析服务,使用了类似如下的\_mapping

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | {      "settings" : {          "number\_of\_shards" : 20      },      "mappings" : {        "client" : {          "properties" : {            "ip" : {              "type" : "long"            },            "cost" : {              "type" : "long"            },  } |

现在问题来了,日志中输出的"127.0.0.1"这类的IP地址在Elasticsearch中是不能转化为long的(报错Java.lang.NumberFormatException)，所以我们必须将字段改为string型或者ip型(Elasticsearch支持， 数据类型可见[mapping-core-types](http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/mapping-core-types.html))才能达到理想的效果.  
  
目标明确了，就是改掉mapping的ip的field type即可.  
elasticsearch.org找了一圈 嘿嘿, update一下即可

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | curl -XPUT localhost:8301/store/client/\_mapping -d '  {      "client" : {          "properties" : {              "local\_ip" : {"type" : "string", "store" : "yes"}          }      }  } |

报错结果

|  |  |
| --- | --- |
| 1 | {"error":"MergeMappingException[Merge failed with failures {[mapper [local\_ip] of different type, current\_type [long], merged\_type [string]]}]","status":400} |

尼玛 真逗  我long想转一下string 居然失败（elasticsearch产品层面理应支持这种无损转化）  无果  
Google了一下类似的案例 ([案例](http://elasticsearch-users.115913.n3.nabble.com/Update-mapping-td3952162.html))  
在一个帖子中得到的elasticsearch开发人员的准确答复

　　"You can't change existing mapping type, you need to create a new index with the correct mapping and index the data again."  
  
想想 略坑啊 我不管是因为elasticsearch还是因为底层Lucene的原因，修改一个field需要对所有已有数据的所有field进行reindex，这本身就是一个逆天的思路，但是elasticsearch的研发人员还觉得这没有什么不合理的.  
  
在Elasticsearch上游逛了一圈，上面这样写到  
(http://www.elasticsearch.org/blog/changing-mapping-with-zero-downtime/)  
the problem — why you can’t change mappings  
  
You can only find that which is stored in your index. In order to make your data searchable, your database needs to know what type of data each field contains and how it should be indexed. If you switch a field type from e.g. a string to a date, all of the data for that field that you already have indexed becomes useless. One way or another, you need to reindex that field.  
  
...  
OK，这一段话很合理，我改了一个field的类型 需要对这个field进行reindex，如论哪种数据库都需要这么做，没错.  
我们再继续往下看看，reindexing your data, 尼玛一看，弱爆了，他的reindexing your data不是对修改的filed进行reindex，而是创建了一个新的index，对所有的filed进行reindexing, 太逆天了。  
  
吐槽归吐槽，这个事情逃不了，那我就按他的来吧.  
首先创建一个新的索引

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | curl -XPUT localhost:8305/store\_v2 -d '  {      "settings" : {          "number\_of\_shards" : 20      },      "mappings" : {        "client" : {          "properties" : {            "ip" : {              "type" : "string"            },            "cost" : {              "type" : "long"            },  } |

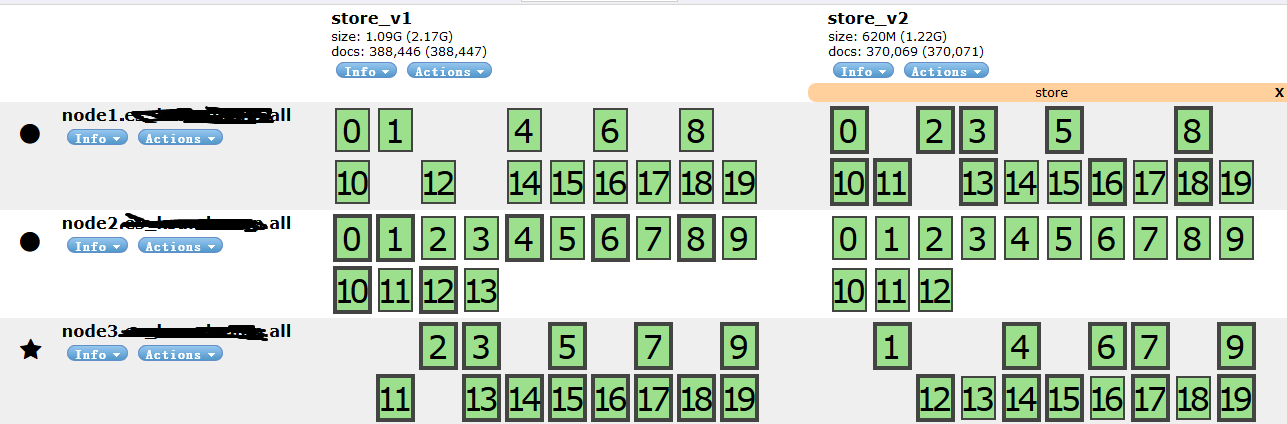
等等，我创建了新索引,client往Elasticsearch的代码不会需要修改吧，瞅了一眼，有解决方案，建立一个alias（别名，和C++引用差不多），通过alias来实现对后面索引数据的解耦合，看到这，舒了一口气。  
  
现在的问题是 这是一个线上服务，不能停服务，所以我需要一个倒数据到我的新索引的一个方案  
Elasticsearch官网写到  
　　pull the documents in from your old index, using a scrolled search and index them into the new index using the bulk API. Many of the client APIs provide a reindex() method which will do all of this for you. Once you are done, you can delete the old index.  
第一句，看起来很美好，找了一圈，尼玛无图无真相，Google都没有例子，你让我怎么导数据？  
第二句 client APIS, 看起来只有这个方法可搞了  
  
python用起来比较熟，所以我就直接选 pyes了，装了一大堆破依赖库之后，终于可以run起来了

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | import pyes  conn = pyes.es.ES("http://10.xx.xx.xx:8305/")  search = pyes.query.MatchAllQuery().search(bulk\_read=1000)  hits = conn.search(search, 'store\_v1', 'client', scan=True, scroll="30m", model=lambda \_,hit: hit)  for hit in hits:       #print hit       conn.index(hit['\_source'], 'store\_v2', 'client', hit['\_id'], bulk=True)  conn.flush() |

花了大概一个多小时，新的索引基本和老索引数据一致了，对于线上完成瞬间的增量，这里没心思关注了，数据准确性要求没那么高，得过且过。  
  
接下来修改alias别名的指向（如果你之前没有用alias来改mapping,纳尼就等着哭吧）

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | curl -XPOST localhost:8305/\_aliases -d '  {      "actions": [          { "remove": {              "alias": "store",              "index": "store\_v1"          }},          { "add": {              "alias": "store",              "index": "store\_v2"          }}      ]  }  ' |

啷啷锵锵，正在追数据中



等新索引的数据已经追上时

将老的索引删掉

|  |  |
| --- | --- |
| 1 | curl -XDELETE localhost:8303/store\_v1 |

至此完成！

一件如此简单的事情，Elasticsearch居然能让他变得如此复杂，真是牛逼啊...