索引管理 总-细分

1. 常用cat api
2. 创建索引
   1. 参数：
      1. **aliases：索引别名**
      2. **mappings：索引映射**
         1. **字段类型：略**
         2. **参数：**
            1. [analyzer](https://www.elastic.co/guide/en/elasticsearch/reference/7.6/analyzer.html)：分词器
            2. search\_analyser：搜索词分词器
            3. doc\_values：正排索引选项
            4. [enabled](https://www.elastic.co/guide/en/elasticsearch/reference/7.6/enabled.html)：是否创建倒排索引
            5. index：未是否对创建对当前字段创建索引，默认true，如果不创建索引，该字段不会通过索引被搜索到,但是仍然会在source元数据中展示
            6. [fielddata](https://www.elastic.co/guide/en/elasticsearch/reference/7.6/fielddata.html)：搜索时正排索引选项
            7. [fields](https://www.elastic.co/guide/en/elasticsearch/reference/7.6/multi-fields.html)：字段
            8. properties：字段属性
            9. [format](https://www.elastic.co/guide/en/elasticsearch/reference/7.6/mapping-date-format.html)：常用语时间类型字段的格式化
            10. [ignore\_above](https://www.elastic.co/guide/en/elasticsearch/reference/7.6/ignore-above.html)：字段保留的长度
            11. meta：金针菇
            12. normalizer：processon 07分词器 图
            13. norms：计算评分用的，如果你确定当前字段将来不需要计算评分，设置false可以节省大量的磁盘空间，有助于提升性能。常见的比如filter和agg字段，都可以设为关闭。
            14. null\_value：为null值设置默认值
            15. **similarity：相关度评分算法**

**BM25：概率相关性模型（5.0之后（包括）默认算法）**

**classic：空间向量模型（向量空间模型），TF-IDF（5.0之前）**

**boolean：**

* + - * 1. **store：开辟另一块存储空间，可以节省带宽**

**注意：\_sourse：设置为false，则不存储元数据，可以节省磁盘，并且不影响搜索。但是禁用\_source必须三思而后行：**

[update](https://www.elastic.co/guide/en/elasticsearch/reference/7.9/docs-update.html)，[update\_by\_query](https://www.elastic.co/guide/en/elasticsearch/reference/7.9/docs-update-by-query.html)和[reindex](https://www.elastic.co/guide/en/elasticsearch/reference/7.9/docs-reindex.html)不可用。

高亮失效

reindex失效，原本可以修改的mapping部分参数将无法修改，并且无法升级索引

无法查看元数据和聚合搜索

影响索引的容灾能力。

如果是出于节省磁盘空间的目的，可以考虑压缩元数据（LZ4压缩）。但是会损失性能。

* + 1. **settings：索引的配置选项**
       1. **index.number\_of\_shards：**主分片个数（7.x以后默认是1，以前默认是5），每个索引分片数量上限：1024，通过以下方式修改。

export ES\_JAVA\_OPTS="-Des.index.max\_number\_of\_shards=128"

* + - 1. number\_of\_replicas：为每个主分片分配的副本个数，默认是1。
      2. **index.shard.check\_on\_startup**：检查分片是否损坏（不要轻易修改）
         1. false：不检查（默认）
         2. checksum：检查物理损坏
         3. true：检查物理损坏和逻辑损坏（消耗大量CPU和内存资源）。
      3. **index.codec：**best\_compression（压缩元数据，会降低性能）
      4. index.auto\_expand\_replicas**：**根据集群中数据节点的数量自动扩展副本的数量。设置为以短划线分隔的上下限（例如0-5）或all 用于上限（例如0-all）。默认为false（即禁用）。
      5. index.max\_result\_window：最大值页容量 默认1W
      6. **index.blocks.read\_only：**设置为true使索引和元数据只读
      7. **index.blocks.read：true：禁止索引读操作**
      8. **index.blocks.write：true：禁止写索引**
      9. **index.blocks.metadata：禁用元数据读写**
      10. **index.max\_terms\_count：查询语句最大term数**
      11. **index.max\_regex\_length：正则最大长度，默认1000**
          1. **index.routing.allocation.enable：控制索引的分片分配**all （默认）-允许为所有分片分配分片。
          2. primaries -仅允许为主要分片分配分片。
          3. new\_primaries -仅允许为新创建的主分片分配分片。
          4. none -不允许分片分配。
      12. **index.routing.rebalance.enable：分片自动均衡**
          1. all （默认）-允许所有分片重新平衡。
          2. primaries -仅允许对主要分片进行分片重新平衡。
          3. replicas -仅允许对副本分片进行分片重新平衡。
          4. none -不允许分片重新平衡

1. 判断索引是否存在：HEAD /my\_target\_index
2. 删除索引：
3. 查询索引信息：
4. 打开/关闭索引：在进行一些运维操作的时候，需要关闭索引，完成之后再打开索引。
5. 索引压缩：实际上是压缩的分片，并非在原有索引上压缩，而是生成了一个新的索引，由于使用了hash路由算法以及索引不可变的特性，target index分片数量必须为source index的约数。比如source index p shard：12,那么target index p shard只能是6 4 3 2 1，如果比如source index p shard是质数，那target index p shard只能是1。
   1. 前提条件：
      1. 要压缩的索引必须是只读
      2. target index的所有p shard必须位于同一节点。
      3. 索引的健康状态必须为green
      4. target index不能已存在
      5. 索引的doc数量不能超过2 147 483 519个，因为单个分片最大支持这么多个doc。
      6. 目标节点所在服务器必须有足够大的磁盘空间。
      7. target index name必须满足一下条件
         1. 仅小写
         2. 不能包括\，/，\*，?，"，<，>，|，``（空格字符）， ,，#
         3. 7.0之前的索引可能包含冒号（:），但已过时，并且在7.0+中不支持
         4. 无法下手-，\_，+
         5. 不能为.或..
         6. 不能超过255个字节（请注意它是字节，因此多字节字符将更快地计入255个限制）
         7. 不建议使用以.开头的名称，但[隐藏索引](https://www.elastic.co/guide/en/elasticsearch/reference/7.9/index-modules.html#index-hidden)和由插件管理的内部索引 除外
   2. 为了简化操作，可以备份数据后，删除副本，然后压缩，完成之后在创建副本。
      1. 备份索引
      2. 设置禁止写入
      3. 删除副本
      4. 迁移主分片
   3. 工作原理和过程
      1. 创建一个新的目标索引，其定义与源索引相同，但主碎片数量较少。
      2. 将段从源索引硬链接到目标索引。如果文件系统不支持硬链接，则将所有segment file都复制到新索引中，复制过程很耗时。
      3. shard recovery操作，恢复目标索引。
   4. 监控过程：GET \_cat/recovery?v
6. refresh：刷新索引。
7. rollover index：
8. 克隆索引：\_clone，必须先rollover
   1. 和压缩索引过程基本相同，不同的是，克隆操作，source和target索引完全相同
9. index template：
   1. 索引模板: 就是把已经创建好的某个索引的参数设置(settings)和索引映射(mapping)保存下来作为模板, 在创建新索引时, 指定要使用的模板名, 就可以直接重用已经定义好的模板中的设置和映射。
   2. 适用场景：适用于需要频繁创建重复索引的场景，比如：网站的行为日志使用rollover index创建索引，写好索引模板就不用了每次在创建索引的时候手动写mapper、setting了，索引模板一般和索引别名一起使用。