

<https://blog.csdn.net/u010824591/article/details/78614505>

## 集群健康：[http://ip:9200/\\_cluster/health?pretty](http://ip:9200/_cluster/health?pretty)

- 1 status: 集群状态，分为green、yellow和red。
- 2 number\_of\_nodes/number\_of\_data\_nodes: 集群的节点数和数据节点数。
- 3 active\_primary\_shards: 集群中所有活跃的主分片数。
- 4 active\_shards: 集群中所有活跃的分片数。
- 5 relocating\_shards: 当前节点迁往其他节点的分片数量，通常为0，当有节点加入或者退出时该值会增加。
- 6 initializing\_shards: 正在初始化的分片。
- 7 unassigned\_shards: 未分配的分片数，通常为0，当有某个节点的副本分片丢失该值就会增加。
- 8 number\_of\_pending\_tasks: 是指主节点创建索引并分配shards等任务，如果该指标数值一直未减小代表集群存在不稳定因素
- 9 active\_shards\_percent\_as\_number: 集群分片健康度，活跃分片数占总分片数比例。
- 10 number\_of\_pending\_tasks: pending task只能由主节点来进行处理，这些任务包括创建索引并将shards分配给节点。

## 集群状态：[http://ip:9200/\\_cluster/stats?pretty](http://ip:9200/_cluster/stats?pretty)

- 1 indices.count: 索引总数。
- 2 indices.shards.total: 分片总数。
- 3 indices.shards primaries: 主分片数量。
- 4 docs.count: 文档总数。
- 5 store.size\_in\_bytes: 数据总存储容量。
- 6 segments.count: 段总数。
- 7 nodes.count.total: 总节点数。
- 8 nodes.count.data: 数据节点数。
- 9 nodes.process.cpu.percent: 节点CPU使用率。
- 10 fs.total\_in\_bytes: 文件系统使用总容量。
- 11 fs.free\_in\_bytes: 文件系统剩余总容量。

## 节点监控：[http://ip:9200/\\_nodes/stats?pretty](http://ip:9200/_nodes/stats?pretty)

- 1 关键指标说明:
- 2 name: 节点名。
- 3 roles: 节点角色。
- 4 indices.docs.count: 索引文档数。
- 5 segments.count: 段总数。
- 6 jvm.heap\_used\_percent: 内存使用百分比。
- 7 thread\_pool.{bulk, index, get, search}.{active, queue, rejected}: 线程池的一些信息，包括bulk、index、get和search线程池，主要指标有active（激活）线程数，线

程queue（队列）数和rejected（拒绝）线程数量。

8

9 以下一些指标是一个累加值，当节点重启之后会清零。

10 indices.indexing.index\_total: 索引文档数。

11 indices.indexing.index\_time\_in\_millis: 索引总耗时。

12 indices.get.total: get请求数。

13 indices.get.time\_in\_millis: get请求总耗时。

14 indices.search.query\_total: search总请求数。

15 indices.search.query\_time\_in\_millis: search请求总耗时。indices.search.fetch\_h\_total: fetch操作总数量。

16 indices.search.fetch\_time\_in\_millis: fetch请求总耗时。

17 jvm.gc.collectors.young.collection\_count: 年轻代垃圾回收次数。

18 jvm.gc.collectors.young.collection\_time\_in\_millis: 年轻代垃圾回收总耗时。

19 jvm.gc.collectors.old.collection\_count: 老年代垃圾回收次数。

20 jvm.gc.collectors.old.collection\_time\_in\_millis: 老年代垃圾回收总耗时。

21

22 一些需要计算的指标:

23 节点监控的计算指标主要分为两类，分别为请求速率指标和请求处理延迟指标，下面作具体介绍。

24 index\_per\_min: 每分钟索引请求数量。计算公式如下:

25 索引请求率=(index\_total两次采集差值)/(系统时间差值 (ms))×60000 (公式1)

26 indexAverage\_per\_min: 索引请求处理延迟。计算公式如下:

27 索引延迟=(index\_time\_in\_millis两次采集差值)/(index\_total两次采集差值) (公式2)

28

29 get\_per\_min: 每分钟get请求数量，计算公式如(公式1)，更改相应参数。

30 getAverage\_per\_min: get请求处理延迟，计算公式如(公式2)，更改相应参数。

31 merge\_per\_min: 每分钟merge请求数量，计算公式如(公式1)，更改相应参数。

32 mergeAverage\_per\_min: merge请求处理延迟，计算公式如(公式2)，更改相应参数。

33 searchQuery\_per\_min: 每分钟query请求数量，计算公式如(公式1)，更改相应参数。

34 searchQueryAverage\_per\_min: query请求延迟，计算公式如(公式2)，更改相应参数。

35 searchFetch\_per\_min: 每分钟fetch请求数量，计算公式如(公式1)，更改相应参数。

36 searchFetchAverage\_per\_min: fetch请求延迟，计算公式如(公式2)，更改相应参数。

37 youngGc\_per\_min: 每分钟young gc数量，计算公式如(公式1)，更改相应参数。

38 youngGcAverage\_per\_min: young gc请求延迟，计算公式如(公式2)，更改相应参数。

39 oldGc\_per\_min: 每分钟old gc数量，计算公式如(公式1)，更改相应参数。

40 oldGcAverage\_per\_min: old gc请求延迟，计算公式如(公式2)，更改相应参数。

**索引监控：** [http://ip:9200/\\_stats?pretty](http://ip:9200/_stats?pretty)

```
1  关键指标说明（indexname泛指索引名称）：
2  indexname primaries.docs.count: 索引文档数量。
3
4  以下一些指标是一个累加值，当节点重启之后会清零。
5  indexname primaries.indexing.index_total: 索引文档数。
6  indexname primaries.indexing.index_time_in_millis: 索引总耗时。
7  indexname primaries.get.total: get请求数。
8  indexname primaries.get.time_in_millis: get请求总耗时。
9  indexname primaries.search.query_total: search总请求数。
10 indexname primaries.search.query_time_in_millis: search请求总耗时。
    indices.search.fetch_total: fetch操作总数量。
11 indexname primaries.search.fetch_time_in_millis: fetch请求总耗时。
12 indexname primaries.refresh.total: refresh请求总量。
13 indexname primaries.refresh.total_time_in_millis: refresh请求总耗时。
14 indexname primaries.flush.total: flush请求总量。
15 indexname primaries.flush.total_time_in_millis: flush请求总耗时。
```

## 其他指标说明：

在节点监控和索引监控时可以获取到一些操作数据，例如indexing、search、get等。对于这些指标参数有一些需要注意的地方：

### search.query

query\_total：查询总数，指的每个分片上的查询次数，一般如果有五个分片的话进行一次查询query\_total = 5。注意在统计指标的时候应该统计的是total分片的，不应该只统计主分片的值，因为查询请求不一定全部分发到主分片。

### get

total:get请求数，一般get都需要指定文档id。同样get也应该统计total分片的值，如果只统计主分片会造成数据变少，因为有时候的get请求不是从主分片取的数据。

### indexing

index\_total和index\_time\_in\_millis指标是根据索引的文档数来记录，同样需要统计total分片的数据，total分片包括副本的索引请求，一般1个副本的话index\_total会乘以2。

### search.fetch

一个查询分为两个阶段，一是query另一个是fetch，fetch的情况与get相似，fetch.total指fetch请求数，fetch有可能从主分片拉取数据，也有可能从副本拉取数据。如果只统计主分片的数据会丢失数据。

