

title: Elasticsearch 性能指标采集 catalog: SuperCloud IT 监控 thumbnail: attachments/elasticsearch.png

tags: IT运维,指标采集


简介

Elasticsearch提供了许多指标，可以帮助您检测故障迹象并在遇到不可靠的节点，内存不足的错误以及较长的垃圾收集时间等问题时采取措施。需要监视的几个关键领域是：

- 集群运行状况和节点可用性
- 主机的网络和系统
- 搜索性能指标
- 索引性能指标
- 内存使用和GC指标
- 资源饱和度和错误


场景视图



场景视图Json文件 

内置视图



内置视图Json文件 

前置条件

- 已安装 DataKit ([DataKit 安装文档](#))

配置

监控指标采集

进入 DataKit 安装目录下的 `conf.d/db` 目录，复制 `elasticsearch.conf.sample` 并命名为 `elasticsearch.conf`。示例如下：

```

[[inputs.elasticsearch]]

## specify a list of one or more Elasticsearch servers
# you can add username and password to your url to use basic authentication:
# servers = ["http://user:pass@localhost:9200"]
servers = ["http://localhost:9200"]

## Timeout for HTTP requests to the elastic search server(s)
http_timeout = "5s"

## When local is true (the default), the node will read only its own stats.
## Set local to false when you want to read the node stats from all nodes
## of the cluster.
local = true

## Set cluster_health to true when you want to also obtain cluster health stats
cluster_health = true

## Adjust cluster_health_level when you want to also obtain detailed health stats
## The options are
## - indices (default)
## - cluster
# cluster_health_level = "cluster"

## Set cluster_stats to true when you want to also obtain cluster stats.
cluster_stats = true

## Only gather cluster_stats from the master node. To work this require local = tr
cluster_stats_only_from_master = true

## Indices to collect; can be one or more indices names or _all
indices_include = ["_all"]

## One of "shards", "cluster", "indices"
indices_level = "shards"

## node_stats is a list of sub-stats that you want to have gathered. Valid options
## are "indices", "os", "process", "jvm", "thread_pool", "fs", "transport", "http"
## "breaker". Per default, all stats are gathered.
node_stats = ["jvm", "http", "indices", "os", "process", "thread_pool", "fs", "transport"]

## HTTP Basic Authentication username and password.
# username = ""
# password = ""

## Optional TLS Config
# tls_ca = "/etc/telegraf/ca.pem"
# tls_cert = "/etc/telegraf/cert.pem"
# tls_key = "/etc/telegraf/key.pem"
## Use TLS but skip chain & host verification
# insecure_skip_verify = false

```

重新启动datakit生效

```
systemctl restart datakit
```

日志采集

进入 DataKit 安装目录下的 `conf.d/log` 目录，复制 `tailf.conf.sample` 并命名为 `tailf.conf`。示例如下：

```
[[inputs.tailf]]
  # glob logfiles
  # required
  logfiles = ["/var/log/elasticsearch/solution.log"]

  # glob filter
  ignore = [""]

  # required
  source = "es_clusterlog"

  # grok pipeline script path
  pipeline = "elasticsearch_cluster_log.p"

  # read file from beginning
  # if from_begin was false, off auto discovery file
  from_beginning = true

  ## characters are replaced using the unicode replacement character
  ## When set to the empty string the data is not decoded to text.
  ## ex: character_encoding = "utf-8"
  ##   character_encoding = "utf-16le"
  ##   character_encoding = "utf-16le"
  ##   character_encoding = "gbk"
  ##   character_encoding = "gb18030"
  ##   character_encoding = ""
  # character_encoding = ""

  ## The pattern should be a regexp
  ## Note the use of '''XXX'''
  # match = '''^\d{4}-\d{2}-\d{2}'''

  #Add Tag for elasticsearch cluster
  [inputs.tailf.tags]
    cluster_name = "solution"
```

Elasticsearch集群信息日志切割grok脚本



重新启动datakit生效

```
systemctl restart datakit
```

采集指标

配置采集器后默认指标以行协议的方式收集

指标集 `elasticsearch_cluster_health`

前置条件： - `cluster_health = true`

标签

| 标签名 | 描述 |
|-------------------|----|
| <code>name</code> | |

指标

| 指标 | 描述 | 类型 | 单位 |
|--|-------------------|---------|----|
| <code>active_primary_shards</code> | 活动主分区的数量 | integer | - |
| <code>active_shards</code> | 活动主分区和副本分区的总数 | integer | - |
| <code>active_shards_percent_as_number</code> | 群集中活动碎片的比率，以百分比表示 | float | - |
| <code>delay_unassigned_shards</code> | 其分配因超时设置而延迟的分片数 | integer | - |
| <code>initializing_shards</code> | 正在初始化的分片数 | integer | - |
| <code>number_of_data_nodes</code> | 作为专用数据节点的节点数 | integer | - |
| <code>number_of_in_flight_fetch</code> | 未完成的访存数量 | integer | - |
| <code>number_of_nodes</code> | 集群中的节点数 | integer | - |
| <code>number_of_pending_tasks</code> | 尚未执行的集群级别更 | integer | - |

| | 改的数量 | | |
|---|------------------------------|---|----|
| <code>relocating_shards</code> | 正在重定位的分片的数量 | integer | - |
| <code>status</code> | 集群的运行状况，基于其主要和副本分片的状态 | enum("green","yellow","red") | - |
| <code>status_code</code> | 集群的运行状况，基于其主要和副本分片的状态 | integer, green = 1, yellow = 2, red = 3 | - |
| <code>task_max_waiting_in_queue_millis</code> | 自最早的初始化任务等待执行以来的时间 | integer | ms |
| <code>timed_out</code> | 如果false响应在timeout参数指定的时间段内返回 | boolean | - |
| <code>unassigned_shards</code> | 未分配的分片数 | boolean | - |

status状态说明

green所有分片均已分配。

yellow所有主分片均已分配，但一个或多个副本分片未分配。如果群集中的某个节点发生故障，则在修复该节点之前，某些数据可能不可用。

red 未分配一个或多个主分片，因此某些数据不可用。在集群启动期间，这可能会短暂发生，因为已分配了主要分片。

指标集 `elasticsearch_cluster_health_indices`

前置条件： - `cluster_health = true` - `cluster_health_level = "indices"`

标签

| 标签名 | 描述 |
|-------|----|
| name | |
| index | |

指标

| 指标 | 描述 | 类型 | 单位 |
|-----------------------|-----------------------|---|----|
| active_primary_shards | 活动主分区数量 | integer | - |
| active_shards | 活动主分区和副本分区的总数 | integer | - |
| initializing_shards | 正在初始化的分片数 | integer | - |
| number_of_replicas | 复制中的分片数量 | integer | - |
| number_of_shards | 分区数量 | integer | - |
| relocating_shards | 正在重定位的分片的数量 | integer | - |
| status | 集群的运行状况，基于其主要和副本分片的状态 | enum("green","yellow","red") | - |
| status_code | 集群的运行状况，基于其主要和副本分片的状态 | integer, green = 1, yellow = 2, red = 3 | - |
| unassigned_shards | 未分配的分片数 | integer | - |

指标集 `elasticsearch_clusterstats_indices`

前置条件：

- `cluster_stats = true`

标签

| 标签名 | 描述 |
|--------------|----|
| cluster_name | |
| node_name | |
| status | |

指标

| 指标 | 描述 | 类 | 单 |
|----|----|---|---|
|----|----|---|---|

| | | 型 | 位 |
|---|--|-------|------|
| <code>completion_size_in_bytes</code> | 用于完成分配给所选节点的所有分片的内存总量 | float | byte |
| <code>count</code> | 已将分片分配给选定节点的索引总数 | float | - |
| <code>docs_count</code> | 分配给选定节点的所有主分片中未删除文档的总数 | float | - |
| <code>docs_deleted</code> | 分配给所选节点的所有主要分片中已删除文档的总数 | float | - |
| <code>fielddata_evictions</code> | 在分配给选定节点的所有分片上，来自字段数据高速缓存的逐出总数 | float | - |
| <code>fielddata_memory_size_in_bytes</code> | 用于分配给选定节点的所有分片上的字段数据缓存的内存总量 | float | byte |
| <code>query_cache_cache_count</code> | 在分配给所选节点的所有分片上添加到查询缓存的条目总数。该数字包括当前和逐出的条目 | float | - |
| <code>query_cache_cache_size</code> | 当前分配给选定节点的所有分片上查询缓存中当前条目的总数 | float | - |
| <code>query_cache_evictions</code> | 分配给所选节点的所有分片上的查询缓存逐出的总数 | float | - |
| <code>query_cache_hit_count</code> | 分配给所选节点的所有分片上的查询缓存命中总数 | float | - |
| <code>query_cache_memory_size_in_bytes</code> | 分配给所选节点的所有分片上用于查询缓存的内存总量 | float | byte |
| <code>query_cache_miss_count</code> | 分配给所选节点的所有分片上的查询高速缓存未命中总数 | float | - |
| <code>query_cache_total_count</code> | 查询高速缓存中分配给选定节点的所有分片上的命中总数和未命中总数 | float | - |
| <code>segments_count</code> | 分配给选定节点的所有分 | float | - |

| | | | |
|---|------------------------------|-------|------|
| | 片上的分段总数 | | |
| <code>segments_doc_values_memory_in_bytes</code> | 用于分配给所选节点的所有分片上的doc值的内存总量 | float | byte |
| <code>segments_fixed_bit_set_memory_in_bytes</code> | 由分配给选定节点的所有分片上的固定位集使用的内存总量 | float | byte |
| <code>segments_index_writer_memory_in_bytes</code> | 所有索引写程序在分配给选定节点的所有分片上使用的内存总量 | float | byte |
| <code>segments_max_unsafe_auto_id_timestamp</code> | 最近重试的索引请求的Unix时间戳 | float | ms |
| <code>segments_memory_in_bytes</code> | 用于分配给选定节点的所有分片上的段的内存总量 | float | byte |
| <code>segments_norms_memory_in_bytes</code> | 用于分配给所选节点的所有分片上的归一化因子的内存总量 | float | byte |
| <code>shards_index primaries_avg</code> | 索引中主碎片的平均数量，仅计算分配给选定节点的碎片 | float | - |
| <code>shards_index primaries_max</code> | 索引中主要分片的最大数量，仅计算分配给选定节点的分片 | float | - |
| <code>shards_index primaries_min</code> | 索引中主分片的最小数量，仅计算分配给选定节点的分片 | float | - |
| <code>shards_index_replication_avg</code> | 索引中的平均复制因子，仅计算分配给选定节点的分片 | float | - |
| <code>shards_index_replication_max</code> | 索引中的最大复制因子，仅计算分配给所选节点的分片 | float | - |
| <code>shards_index_replication_min</code> | 索引中的最小复制因子，仅计算分配给选定节点的分片 | float | - |
| <code>shards_index_shards_avg</code> | 索引中的最小分片数，仅计算分配给选定节点的分片 | float | - |
| | 索引中最大的分片数，仅 | | |

| | | | |
|--------------------------------------|-------------------------|-------|------|
| <code>shards_index_shards_max</code> | 计算分配给选定节点的分片 | float | - |
| <code>shards_index_shards_min</code> | 索引中的最小分片数，仅计算分配给选定节点的分片 | float | - |
| <code>shards primaries</code> | 分配给所选节点的主分片数 | float | - |
| <code>shards_replication</code> | 所有选定节点上的副本分片与主分片的比率 | float | - |
| <code>shards_total</code> | 分配给所选节点的分片总数 | float | - |
| <code>store_size_in_bytes</code> | 分配给所选节点的所有分片的总大小 | float | byte |

指标集 `elasticsearch_clusterstats_nodes`

标签

| 标签名 | 描述 |
|---------------------------|----|
| <code>cluster_name</code> | |
| <code>node_name</code> | |
| <code>status</code> | |

指标

| 指标 | 描述 | 类型 | 单位 |
|--------------------------------------|-------------------------|-------|------|
| <code>count_coordinating_only</code> | 没有角色的选定节点数。这些节点仅被视为协调节点 | float | - |
| <code>count_data</code> | 数据节点数据量 | float | - |
| <code>count_ingest</code> | 查询节点数量 | float | - |
| <code>count_master</code> | master节点数量 | float | - |
| <code>count_total</code> | 所选节点的总数 | float | - |
| <code>fs_available_in_bytes</code> | 所有选定节点上文件存储中可用于JVM的字节总数 | float | byte |

| | | | |
|--|--------------------------|--------|-----------|
| <code>fs_free_in_bytes</code> | 所有选定节点上文件存储中未分配字节的总数 | float | byte |
| <code>fs_total_in_bytes</code> | 所有选定节点中所有文件存储的总大小 | float | byte |
| <code>jvm_max_uptime_in_millis</code> | 自JVM上次启动以来的正常运行时间 | float | ms |
| <code>jvm_mem_heap_max_in_bytes</code> | 可供所有选定节点上的堆使用的最大内存量 | float | byte |
| <code>jvm_mem_heap_used_in_bytes</code> | 所有选定节点上的堆当前正在使用的内存 | float | byte |
| <code>jvm_threads</code> | JVM在所有选定节点上使用的活动线程数 | float | - |
| <code>jvm_versions_0_count</code> | 使用JVM的选定节点总数 | float | - |
| <code>jvm_versions_0_version</code> | 一个或多个选定节点使用的JVM版本 | string | - |
| <code>jvm_versions_0_vm_name</code> | JVM的名称 | string | - |
| <code>jvm_versions_0_vm_vendor</code> | JVM的供应商 | string | - |
| <code>jvm_versions_0_vm_version</code> | JVM的完整版本号 | string | - |
| <code>network_types_http_types_security4</code> | 包含有关所选节点使用的HTTP网络类型的统计信息 | float | - |
| <code>network_types_transport_types_security4</code> | 包含有关所选节点使用的传输网络类型的统计信息 | float | - |
| <code>os_allocated_processors</code> | 用于计算所有选定节点上的线程池大小的处理器数量 | float | - |
| <code>os_available_processors</code> | 所有选定节点上可用于JVM的处理器数 | float | - |
| <code>os_mem_free_in_bytes</code> | 可用物理内存量 | float | byte |
| <code>os_mem_free_percent</code> | 可用内存的百分比 | float | - |
| <code>os_mem_total_in_bytes</code> | 物理内存总量 | float | byte |
| <code>os_mem_used_in_bytes</code> | 已使用的物理内存量 | float | - byte |
| | | | |

| | | | |
|--|--------------------------------------|--------|---|
| <code>os_names_0_count</code> | 使用操作系统的所选节点数 | float | - |
| <code>os_names_0_name</code> | 一个或多个选定节点使用的操作系统的名称 | float | - |
| <code>os_pretty_names_0_count</code> | 使用操作系统的所选节点数 | float | - |
| <code>os_pretty_names_0_pretty_name</code> | 一个或多个选定节点使用的操作系统的可读名称 | float | - |
| <code>process_cpu_percent</code> | 使用操作系统的所选节点数 | float | - |
| <code>process_open_file_descriptors_avg</code> | 同时打开的文件描述符的平均数量。-1如果不支持，则返回 | float | - |
| <code>process_open_file_descriptors_max</code> | 所有选定节点上允许的并发打开文件描述符的最大数量。-1如果不支持，则返回 | float | - |
| <code>process_open_file_descriptors_min</code> | 所有选定节点上允许的并发打开文件描述符的最大数量。-1如果不支持，则返回 | float | - |
| <code>versions_0</code> | 在选定节点上使用的Elasticsearch版本的数组 | string | - |

指标集 `elasticsearch_transport`

前置条件

- 开启 `node_stats` 设置

标签

| 标签名 | 描述 |
|---|----|
| <code>cluster_name</code> | |
| <code>node_attribute_ml.enabled</code> | |
| <code>node_attribute_ml.machine_memory</code> | |
| <code>node_attribute_ml.max_open_jobs</code> | |
| <code>node_attribute_xpack.installed</code> | |
| <code>node_host</code> | |
| <code>node_id</code> | |
| <code>node_name</code> | |

指标

| 指标 | 描述 | 类型 | 单位 |
|-------------------------------|-----------------------------|-------|------|
| <code>rx_count</code> | 在内部群集通信期间，节点接收到的RX（接收）数据包总数 | float | - |
| <code>rx_size_in_bytes</code> | 内部集群通信期间节点接收到的RX数据包的大小 | float | byte |
| <code>server_open</code> | 当前用于节点之间内部通信的入站TCP连接数 | float | - |
| <code>tx_count</code> | 在内部群集通信期间，节点发送的TX（传输）数据包总数 | float | - |
| <code>tx_size_in_bytes</code> | 内部群集通信期间节点发送的TX数据包的大小 | float | byte |

指标集 `elasticsearch_breakers`

标签

| 标签名 | 描述 |
|----------------------------------|----|
| cluster_name | |
| node_attribute_ml.enabled | |
| node_attribute_ml.machine_memory | |
| node_attribute_ml.max_open_jobs | |
| node_attribute_xpack.installed | |
| node_host | |
| node_id | |
| node_name | |

指标

| 指标 | 描述 | 类型 | 单位 |
|---|----|-------|------|
| <code>accounting_estimated_size_in_bytes</code> | - | float | byte |
| <code>accounting_limit_size_in_bytes</code> | - | float | byte |
| <code>accounting_overhead</code> | - | float | - |
| <code>accounting_tripped</code> | - | float | - |
| <code>fielddata_estimated_size_in_bytes</code> | - | float | byte |
| <code>fielddata_limit_size_in_bytes</code> | - | float | byte |
| <code>fielddata_overhead</code> | - | float | - |
| <code>fielddata_tripped</code> | - | float | - |
| <code>in_flight_requests_estimated_size_in_bytes</code> | - | float | byte |
| <code>in_flight_requests_limit_size_in_bytes</code> | - | float | byte |
| <code>in_flight_requests_overhead</code> | - | float | - |
| <code>in_flight_requests_tripped</code> | - | float | - |
| <code>parent_estimated_size_in_bytes</code> | - | float | byte |
| <code>parent_limit_size_in_bytes</code> | - | float | byte |
| <code>parent_overhead</code> | - | float | - |
| <code>parent_tripped</code> | - | float | - |
| <code>request_estimated_size_in_bytes</code> | - | float | byte |
| <code>accounting_estimated_size_in_bytes</code> | - | float | byte |
| <code>request_limit_size_in_bytes</code> | - | float | byte |
| <code>request_overhead</code> | - | float | - |
| <code>request_tripped</code> | - | float | - |

包含断路器的统计信息

`limitsizeinbytes`: 断路器的内存限制（以字节为单位）。`limitsize`: 断路器的内存限制。

`estimatedsizeinbytes`: 用于操作的估计内存（以字节为单位）。`estimatedsize`: 用于该操作的估计内存。`overhead`: 一个常数，断路器的所有估计值都将与该常数相乘以计算最终估计值。`tripped`: 已触发断路器并防止发生内存不足错误的总次数

指标集 `elasticsearch_fs`

标签

| 标签名 | 描述 |
|---|----|
| <code>cluster_name</code> | |
| <code>node_attribute_ml.enabled</code> | |
| <code>node_attribute_ml.machine_memory</code> | |
| <code>node_attribute_ml.max_open_jobs</code> | |
| <code>node_attribute_xpack.installed</code> | |
| <code>node_host</code> | |
| <code>node_id</code> | |
| <code>node_name</code> | |

指标

| 指标 | 描述 | 类型 | 单位 |
|--|---|-------|------|
| <code>data_0_available_in_bytes</code> | 此文件存储上可用于此Java虚拟机的字节总数 | float | - |
| <code>data_0_free_in_bytes</code> | 文件存储中未分配的字节总数 | float | byte |
| <code>data_0_total_in_bytes</code> | 文件存储的总大小 | float | byte |
| <code>io_stats_devices_0_operations</code> | 自启动Elasticsearch以来，Elasticsearch使用的所有设备上的读写操作总数 | float | - |
| <code>io_stats_devices_0_read_kilobytes</code> | 自启动Elasticsearch以来设备读取的总千字节数 | float | kb |
| <code>io_stats_devices_0_read_operations</code> | 自启动Elasticsearch以来已完成的设备读取操作的总数 | float | - |
| <code>io_stats_devices_0_write_kilobytes</code> | 自启动Elasticsearch以来为设备写入的总千字节数 | float | kb |
| <code>io_stats_total_write_operations</code> | 自启动Elasticsearch以来，在Elasticsearch使用的所有设备上完成的写操作总数 | float | - |
| <code>io_stats_devices_0_write_operations</code> | 自启动Elasticsearch以来已完成的设备写入操作总数 | float | - |
| <code>io_stats_total_operations</code> | 自启动Elasticsearch以来，Elasticsearch使用的所有设备 | float | - |

| | | | |
|---|---|-------|------|
| | 上的读写操作总数 | | |
| <code>io_stats_total_read_kilobytes</code> | 自启动Elasticsearch以来在Elasticsearch使用的所有设备上读取的总千字节数 | float | kb |
| <code>io_stats_total_read_operations</code> | 自启动Elasticsearch以来，Elasticsearch使用的所有设备上的读取操作总数 | float | - |
| <code>io_stats_total_write_kilobytes</code> | 自启动Elasticsearch以来在Elasticsearch使用的所有设备上写入的总千字节数 | float | kb |
| <code>timestamp</code> | 上次刷新文件存储统计信息 | | |
| float | ms | | |
| <code>total_available_in_bytes</code> | 所有文件存储上可用于此Java虚拟机的字节总数。根据操作系统或进程级别的限制，该值可能小于 <code>free_in_bytes</code> 。这是Elasticsearch节点可以利用的实际可用磁盘空间量 | float | byte |
| <code>total_free_in_bytes</code> | 所有文件存储中未分配的字节总数 | float | byte |
| <code>total_total_in_bytes</code> | 所有文件存储的总大小 | float | byte |

指标集 `elasticsearch_http`

标签

| 标签名 | 描述 |
|---|----|
| <code>cluster_name</code> | |
| <code>node_attribute_ml.enabled</code> | |
| <code>node_attribute_ml.machine_memory</code> | |
| <code>node_attribute_ml.max_open_jobs</code> | |
| <code>node_attribute_xpack.installed</code> | |
| <code>node_host</code> | |
| <code>node_id</code> | |
| <code>node_name</code> | |

指标

| 指标 | 描述 | 类型 | 单位 |
|---------------------------|-----------------|-------|----|
| <code>current_open</code> | 节点当前打开的HTTP连接数 | float | - |
| <code>total_opened</code> | 为该节点打开的HTTP连接总数 | float | - |

指标集 `elasticsearch_indices`

标签

| 标签名 | 描述 |
|---|----|
| <code>cluster_name</code> | |
| <code>node_attribute_ml.enabled</code> | |
| <code>node_attribute_ml.machine_memory</code> | |
| <code>node_attribute_ml.max_open_jobs</code> | |
| <code>node_attribute_xpack.installed</code> | |
| <code>node_host</code> | |
| <code>node_id</code> | |
| <code>node_name</code> | |

指标

| 指标 | 描述 | 类型 | 单位 |
|---------------------------------------|---|-------|------|
| <code>completion_size_in_bytes</code> | 用于完成分配给该节点的所有分片的内存总量 | float | byte |
| <code>docs_count</code> | Lucene报告的文档数。这 不包括已删除的文档，并 且将所有嵌套文档与其父 文档分开计数。它还不包 括最近被索引但尚未属于 段的文档 | float | - |
| | Lucene报告的已删除文档 的数量，它可能高于或低 于您执行的删除操作的数 量。该数字不包括最近执 行但尚未属于段的删除。 如果合理的话，将通过自 | | |

| | | | |
|---|--|-------|------|
| <code>docs_deleted</code> | 动合并过程清除已删除的文档。此外，Elasticsearch会创建额外的已删除文档，以在内部跟踪分片上最近的操作历史记录 | float | - |
| <code>fielddata_evictions</code> | 字段数据逐出的数量 | float | - |
| <code>fielddata_memory_size_in_bytes</code> | 用于分配给该节点的所有分片上的字段数据缓存的内存总量 | float | byte |
| <code>flush_periodic</code> | 刷新定期操作的数量 | float | - |
| <code>flush_total</code> | 刷新操作的数量 | float | - |
| <code>flush_total_time_in_millis</code> | 执行刷新操作所花费的总时间 | float | ms |
| <code>get_current</code> | 当前正在运行的get操作的数量 | float | - |
| <code>get_exists_time_in_millis</code> | 执行成功的get操作所花费的时间 | float | ms |
| <code>get_exists_total</code> | 成功获取操作的总数 | float | - |
| <code>get_missing_time_in_millis</code> | 执行失败的get操作所花费的时间 | float | ms |
| <code>get_time_in_millis</code> | 执行获取操作所花费的时间 | float | ms |
| <code>get_total</code> | get操作的总数 | float | - |
| <code>indexing_delete_current</code> | 当前正在运行的删除操作的数量 | float | - |
| <code>indexing_delete_time_in_millis</code> | 执行删除操作所花费的时间 | float | ms |
| <code>indexing_delete_total</code> | 删除操作的总数 | float | - |
| <code>indexing_index_current</code> | 当前正在运行的索引操作数 | float | - |
| <code>indexing_index_failed</code> | 索引操作失败的次数 | float | - |
| <code>indexing_index_time_in_millis</code> | 执行索引操作所花费的总时间 | float | ms |
| <code>indexing_index_total</code> | 索引操作的总数 | float | - |

| | | | |
|--|--------------------------|-------|------|
| <code>indexing_noop_update_total</code> | noop操作总数 | float | - |
| <code>indexing_throttle_time_in_millis</code> | 节流操作花费的总时间 | float | ms |
| <code>merges_current</code> | 当前正在运行的合并操作的数量 | float | - |
| <code>merges_current_docs</code> | 当前正在运行的文档合并数 | float | - |
| <code>merges_current_size_in_bytes</code> | 用于执行当前文档合并的内存 | float | byte |
| <code>merges_total</code> | 合并操作的总数 | float | - |
| <code>merges_total_auto_throttle_in_bytes</code> | 自动限制的合并操作的大小 | float | byte |
| <code>merges_total_docs</code> | 合并文档的总数 | float | - |
| <code>merges_total_size_in_bytes</code> | 文档合并的总大小 | float | byte |
| <code>merges_total_stopped_time_in_millis</code> | 停止合并操作所花费的总时间 | float | ms |
| <code>merges_total_throttled_time_in_millis</code> | 限制合并操作花费的总时间 | float | ms |
| <code>merges_total_time_in_millis</code> | 执行合并操作所花费的总时间 | float | ms |
| <code>query_cache_cache_count</code> | 查询缓存中的查询计数 | float | - |
| <code>query_cache_cache_size</code> | 查询缓存的大小 | float | - |
| <code>query_cache_evictions</code> | 查询缓存逐出的数量 | float | - |
| <code>query_cache_hit_count</code> | 查询缓存命中数 | float | - |
| <code>query_cache_memory_size_in_bytes</code> | 用于分配给该节点的所有分片上的查询缓存的内存总量 | float | byte |
| <code>query_cache_miss_count</code> | 查询高速缓存未命中的数量 | float | - |
| <code>query_cache_total_count</code> | 查询缓存中的命中，未命中和缓存的查询的总数 | float | - |
| <code>recovery_current_as_source</code> | 恢复当前源的次数 | float | - |
| <code>recovery_current_as_target</code> | 恢复当前目标的次数 | float | - |
| <code>recovery_throttle_time_in_millis</code> | 恢复节流所花费的总时间 | float | ms |
| | | | |

| | | | |
|---|--------------------------|-------|------|
| <code>refresh_listeners</code> | 刷新侦听器的数量 | float | - |
| <code>refresh_total</code> | 刷新操作的总数 | float | - |
| <code>refresh_total_time_in_millis</code> | 执行刷新操作所花费的总时间 | float | ms |
| <code>request_cache_evictions</code> | 请求高速缓存的逐出数 | float | - |
| <code>request_cache_hit_count</code> | 请求缓存命中数 | float | - |
| <code>request_cache_memory_size_in_bytes</code> | 用于分配给该节点的所有分片上的请求缓存的内存总量 | float | byte |
| <code>request_cache_miss_count</code> | 请求高速缓存未命中的数量 | float | - |
| <code>search_fetch_current</code> | 当前正在运行的提取操作的数量 | float | - |
| <code>search_fetch_time_in_millis</code> | 执行提取操作所花费的时间 | float | ms |
| <code>search_fetch_total</code> | 提取操作的总数 | float | - |
| <code>search_open_contexts</code> | 开放搜索上下文的数量 | float | - |
| <code>search_query_current</code> | 当前正在运行的查询操作的数量 | float | - |
| <code>search_query_time_in_millis</code> | 执行查询操作所花费的时间 | float | ms |
| <code>search_query_total</code> | 查询操作的总数 | float | - |
| <code>search_scroll_current</code> | 当前正在运行的滚动操作的数量 | float | - |
| <code>search_scroll_time_in_millis</code> | 执行滚动操作所花费的时间 | float | ms |
| <code>search_scroll_total</code> | 滚动操作的总数 | float | - |
| <code>search_suggest_current</code> | 当前正在运行的建议操作的数量 | float | - |
| <code>search_suggest_time_in_millis</code> | 执行建议操作所花费的时间 | float | ms |
| <code>search_suggest_total</code> | 建议操作总数 | float | - |
| <code>segments_count</code> | 段数 | float | - |
| | | | |

| | | | |
|---|-----------------------------|-------|------|
| <code>segments_doc_values_memory_in_bytes</code> | 用于分配给该节点的所有分片上的doc值的内存总量 | float | byte |
| <code>segments_fixed_bit_set_memory_in_bytes</code> | 由分配给节点的所有分片上的固定位集使用的内存总量 | float | byte |
| <code>segments_index_writer_memory_in_bytes</code> | 所有索引写程序在分配给该节点的所有分片上使用的内存总量 | float | byte |
| <code>segments_max_unsafe_auto_id_timestamp</code> | 最近重试的索引请求的时间 | float | ms |
| <code>segments_memory_in_bytes</code> | 用于分配给节点的所有分片上的段的内存总量 | float | byte |
| <code>segments_norms_memory_in_bytes</code> | 用于分配给节点的所有分片上的标准化因子的内存总量 | float | byte |
| <code>segments_points_memory_in_bytes</code> | 用于分配给节点的所有分片上的点的内存总量 | float | byte |
| <code>segments_stored_fields_memory_in_bytes</code> | 用于分配给节点的所有分片上的存储字段的内存总量 | float | byte |
| <code>segments_term_vectors_memory_in_bytes</code> | 用于分配给该节点的所有分片上的术语向量的内存总量 | float | byte |
| <code>segments_terms_memory_in_bytes</code> | 用于分配给该节点的所有分片上的术语的内存总量 | float | byte |
| <code>segments_version_map_memory_in_bytes</code> | 所有版本映射在分配给节点的所有分片上使用的内存总量 | float | byte |
| <code>store_size_in_bytes</code> | 分配给节点的所有分片的总大小 | float | byte |
| <code>translog_earliest_last_modified_age</code> | 事务日志的最早最近修改年龄 | float | - |
| <code>translog_operations</code> | 事务日志操作数 | float | - |
| <code>translog_size_in_bytes</code> | 事务日志的大小 | float | byte |
| <code>translog_uncommitted_operations</code> | 未提交的事务日志操作数 | float | - |
| <code>translog_uncommitted_size_in_bytes</code> | 未提交的事务日志操作的 | float | byte |

| | 大小 | | |
|--|-----------------|-------|----|
| <code>warmer_current</code> | 活动索引加热器的数量 | float | - |
| <code>warmer_total</code> | 指数加热器的总数 | float | - |
| <code>warmer_total_time_in_millis</code> | 执行索引预热操作所花费的总时间 | float | ms |

指标集 `elasticsearch_jvm`

标签

| 标签名 | 描述 |
|---|----|
| <code>cluster_name</code> | |
| <code>node_attribute_ml.enabled</code> | |
| <code>node_attribute_ml.machine_memory</code> | |
| <code>node_attribute_ml.max_open_jobs</code> | |
| <code>node_attribute_xpack.installed</code> | |
| <code>node_host</code> | |
| <code>node_id</code> | |
| <code>node_name</code> | |

指标

| 指标 | 描述 | 类型 | 单位 |
|--|-----------------|-------|------|
| <code>buffer_pools_direct_count</code> | 直接缓冲池的数量 | float | - |
| <code>buffer_pools_direct_total_capacity_in_bytes</code> | 直接缓冲池的总容量 | float | byte |
| <code>buffer_pools_direct_used_in_bytes</code> | 直接缓冲池的大小 | float | byte |
| <code>buffer_pools_mapped_count</code> | 映射的缓冲池数 | float | - |
| <code>buffer_pools_mapped_total_capacity_in_bytes</code> | 映射的缓冲池的总容量 | float | byte |
| <code>buffer_pools_mapped_used_in_bytes</code> | 映射的缓冲池的大小 | float | byte |
| <code>classes_current_loaded_count</code> | JVM当前加载的类的数量 | float | - |
| <code>classes_total_loaded_count</code> | 自JVM启动以来已加载的类总数 | float | - |
| <code>classes_total_unloaded_count</code> | 自JVM启动以来已卸载的类的 | float | - |

| | | | |
|--|------------------------------------|-------------|-------|
| | 总数 | | |
| <code>gc_collectors_old_collection_count</code> | 收集老年代对象的JVM垃圾收集器的数量 | float | - |
| <code>gc_collectors_old_collection_time_in_millis</code> | JVM收集老年代对象所花费的总时间 | float | ms |
| <code>gc_collectors_young_collection_count</code> | 收集年轻代对象的JVM垃圾收集器的数量 | float | - |
| <code>gc_collectors_young_collection_time_in_millis</code> | JVM收集年轻代对象所花费的总时间 | float | ms |
| <code>mem_heap_committed_in_bytes</code> | 可供堆使用的内存量 | float | byte |
| | <code>mem_heap_max_in_bytes</code> | 可供堆使用的最大内存量 | float |
| <code>mem_heap_used_in_bytes</code> | 堆当前正在使用的内存 | float | byte |
| <code>mem_heap_used_percent</code> | 堆当前正在使用的内存百分比 | float | - |
| <code>mem_non_heap_committed_in_bytes</code> | 可用的非堆内存量 | float | byte |
| <code>mem_non_heap_used_in_bytes</code> | 使用的非堆内存 | float | byte |
| <code>mem_pools_old_max_in_bytes</code> | 可用于老年代的最大内存量 | float | byte |
| <code>mem_pools_old_peak_max_in_bytes</code> | 历史上可用于老年代的最高内存限制 | float | byte |
| <code>mem_pools_old_peak_used_in_bytes</code> | 历史上老年代使用的内存量 | float | byte |
| <code>mem_pools_old_used_in_bytes</code> | 老年代使用的内存 | float | byte |
| <code>mem_pools_survivor_max_in_bytes</code> | 可供生存空间使用的最大内存量 | float | byte |
| <code>mem_pools_survivor_peak_max_in_bytes</code> | 历史上幸存者空间使用的最大内存量 | float | byte |
| <code>mem_pools_survivor_peak_used_in_bytes</code> | 历史上幸存者空间使用的内存量 | float | byte |
| | | | |

| | | | |
|---|-----------------|-------|------|
| <code>mem_pools_survivor_used_in_bytes</code> | 幸存者空间使用的内存 | float | byte |
| <code>mem_pools_young_max_in_bytes</code> | 可供年轻一代堆使用的最大内存量 | float | byte |
| <code>mem_pools_young_peak_max_in_bytes</code> | 历史上年轻代堆使用的最大内存量 | float | byte |
| <code>mem_pools_young_peak_used_in_bytes</code> | 历史上年轻代堆使用的内存量 | float | byte |
| <code>mem_pools_young_used_in_bytes</code> | 年轻代使用的内存 | float | - |
| <code>threads_count</code> | JVM使用的活动线程数 | float | - |
| <code>threads_peak_count</code> | JVM使用的最大线程数 | float | - |
| <code>timestamp</code> | 上次刷新JVM统计信息的时间 | float | - |
| <code>uptime_in_millis</code> | JVM正常运行时间 | float | ms |

指标集 `elasticsearch_os`

标签

| 标签名 | 描述 |
|---|----|
| <code>cluster_name</code> | |
| <code>node_attribute_ml.enabled</code> | |
| <code>node_attribute_ml.machine_memory</code> | |
| <code>node_attribute_ml.max_open_jobs</code> | |
| <code>node_attribute_xpack.installed</code> | |
| <code>node_host</code> | |
| <code>node_id</code> | |
| <code>node_name</code> | |

指标

| 指标 | 描述 | 类型 | 单位 |
|---|--|-------|----|
| <code>cgroup_cpu_cfs_period_micros</code> | 时间段，该时间段表示与Elasticsearch进程相同的cgroup中的所有任务应该多长时间重新分配一次对CPU资源的访问 | float | us |

| | | | |
|--|--|-------|------|
| <code>cgroup_cpu_cfs_quota_micros</code> | 一个时间段内，与Elasticsearch进程在同一cgroup中的所有任务可以运行的总时间 | float | us |
| <code>cgroup_cpu_stat_number_of_elapsed_periods</code> | <code>cfsperiodmicros</code> 已过去的报告期数 | float | - |
| <code>cgroup_cpu_stat_number_of_times_throttled</code> | 已限制与Elasticsearch进程在同一cgroup中的所有任务的总时间 | float | ns |
| <code>cgroup_cpu_stat_time_throttled_nanos</code> | 与elasticsearch进程在同一cgroup中的所有任务已被限制的次数 | float | - |
| <code>cgroup_cpuacct_usage_nanos</code> | 与Elasticsearch进程在同一cgroup中的所有任务消耗的总CPU时间 | float | ns |
| <code>cpu_load_average_15m</code> | 系统上15分钟的平均负载（如果15分钟的平均负载不可用，则不存在该字段） | float | - |
| <code>cpu_load_average_1m</code> | 系统上的一分钟平均负载（如果没有一分钟平均负载，则不存在该字段） | float | - |
| <code>cpu_load_average_5m</code> | 系统上五分钟的平均负载（如果没有五分钟的平均负载，则不存在该字段） | float | - |
| <code>cpu_percent</code> | 整个系统的最近CPU使用情况，或者-1如果不支持，则为最近一次 | float | - |
| <code>mem_free_in_bytes</code> | 可用物理内存量 | float | byte |
| <code>mem_free_percent</code> | 可用内存的百分比 | float | - |
| <code>mem_total_in_bytes</code> | 物理内存总量 | float | byte |
| <code>mem_used_in_bytes</code> | 已使用的物理内存量 | float | byte |
| <code>mem_used_percent</code> | 已用内存的百分比 | float | - |
| <code>swap_free_in_bytes</code> | 可用交换空间量 | float | byte |
| <code>swap_total_in_bytes</code> | 交换空间的总量，以字 | float | byte |

| | | | |
|--------------------|--------------|-------|------|
| | 节为单位 | | |
| swap_used_in_bytes | 已使用的交换空间量 | float | byte |
| timestamp | 上次刷新操作系统统计信息 | float | ms |

指标集 `elasticsearch_process`

标签

| 标签名 | 描述 |
|----------------------------------|----|
| cluster_name | |
| node_attribute_ml.enabled | |
| node_attribute_ml.machine_memory | |
| node_attribute_ml.max_open_jobs | |
| node_attribute_xpack.installed | |
| node_host | |
| node_id | |
| node_name | |

指标

| 指标 | 描述 | 类型 | 单位 |
|----------------------------|----------------------------------|-------|----|
| cpu_percent | CPU使用率 | float | - |
| cpu_total_in_millis | 运行Java虚拟机的进程使用的CPU时间 | float | ms |
| max_file_descriptors | 系统上允许的最大文件描述符数量，或者-1如果不支持，则为最大数量 | float | - |
| mem_total_virtual_in_bytes | 保证正在运行的进程可用的虚拟内存大小 | float | - |
| open_file_descriptors | 与当前或-1不支持的文件关联的已打开文件描述符的数量 | float | - |
| timestamp | 上次刷新统计信息 | float | - |

指标集 `elasticsearch_thread_pool`

标签

| 标签名 | 描述 |
|----------------------------------|----|
| cluster_name | |
| node_attribute_ml.enabled | |
| node_attribute_ml.machine_memory | |
| node_attribute_ml.max_open_jobs | |
| node_attribute_xpack.installed | |
| node_host | |
| node_id | |
| node_name | |

指标名

| 指标 | 描述 | 类型 | 单位 |
|-------------------------------|----|-------|----|
| analyze_active | - | float | - |
| analyze_completed | - | float | - |
| analyze_largest | - | float | - |
| analyze_queue | - | float | - |
| analyze_rejected | - | float | - |
| analyze_threads | - | float | - |
| ccr_active | - | float | - |
| ccr_completed | - | float | - |
| ccr_largest | - | float | - |
| ccr_queue | - | float | - |
| ccr_rejected | - | float | - |
| ccr_threads | - | float | - |
| fetch_shard_started_active | - | float | - |
| fetch_shard_started_completed | - | float | - |
| fetch_shard_started_largest | - | float | - |
| force_merge_queue | - | float | - |
| force_merge_rejected | - | float | - |
| force_merge_threads | - | float | - |

| | | | |
|----------------------|---|-------|---|
| generic_active | - | float | - |
| generic_completed | - | float | - |
| generic_largest | - | float | - |
| generic_queue | - | float | - |
| generic_rejected | - | float | - |
| generic_threads | - | float | - |
| get_active | - | float | - |
| get_completed | - | float | - |
| get_largest | - | float | - |
| get_queue | - | float | - |
| get_rejected | - | float | - |
| get_threads | - | float | - |
| index_active | - | float | - |
| index_completed | - | float | - |
| index_largest | - | float | - |
| index_queue | - | float | - |
| index_rejected | - | float | - |
| index_threads | - | float | - |
| listener_active | - | float | - |
| listener_completed | - | float | - |
| listener_largest | - | float | - |
| listener_queue | - | float | - |
| listener_rejected | - | float | - |
| listener_threads | - | float | - |
| management_active | - | float | - |
| management_completed | - | float | - |
| management_largest | - | float | - |
| management_queue | - | float | - |
| management_rejected | - | float | - |
| | | | |

| | | | |
|---------------------------|---|-------|---|
| management_threads | - | float | - |
| ml_autodetect_active | - | float | - |
| ml_autodetect_completed | - | float | - |
| ml_autodetect_largest | - | float | - |
| ml_autodetect_queue | - | float | - |
| ml_autodetect_rejected | - | float | - |
| ml_autodetect_threads | - | float | - |
| ml_datafeed_active | - | float | - |
| ml_datafeed_completed | - | float | - |
| ml_datafeed_largest | - | float | - |
| ml_datafeed_queue | - | float | - |
| ml_datafeed_rejected | - | float | - |
| ml_datafeed_threads | - | float | - |
| ml_utility_active | - | float | - |
| ml_utility_completed | - | float | - |
| ml_utility_largest | - | float | - |
| ml_utility_queue | - | float | - |
| ml_utility_rejected | - | float | - |
| ml_utility_threads | - | float | - |
| refresh_active | - | float | - |
| refresh_completed | - | float | - |
| refresh_largest | - | float | - |
| refresh_queue | - | float | - |
| refresh_rejected | - | float | - |
| refresh_threads | - | float | - |
| rollup_indexing_active | - | float | - |
| rollup_indexing_completed | - | float | - |
| rollup_indexing_largest | - | float | - |
| rollup_indexing_queue | - | float | - |
| rollup_indexing_rejected | - | float | - |

| | | | |
|------------------------------|---|-------|---|
| rollup_indexing_threads | - | float | - |
| search_active | - | float | - |
| search_completed | - | float | - |
| search_largest | - | float | - |
| search_queue | - | float | - |
| search_rejected | - | float | - |
| search_threads | - | float | - |
| search_throttled_active | - | float | - |
| search_throttled_completed | - | float | - |
| search_throttled_largest | - | float | - |
| search_throttled_queue | - | float | - |
| rollup_indexing_completed | - | float | - |
| search_throttled_rejected | - | float | - |
| search_throttled_threads | - | float | - |
| security-token-key_active | - | float | - |
| security-token-key_completed | - | float | - |
| security-token-key_largest | - | float | - |
| security-token-key_queue | - | float | - |
| security-token-key_rejected | - | float | - |
| security-token-key_threads | - | float | - |
| snapshot_active | - | float | - |
| snapshot_completed | - | float | - |
| snapshot_largest | - | float | - |
| snapshot_queue | - | float | - |
| snapshot_rejected | - | float | - |
| snapshot_threads | - | float | - |
| warmer_active | - | float | - |
| warmer_completed | - | float | - |
| warmer_largest | - | float | - |
| | | | |

| | | | |
|--------------------------------|---|-------|---|
| <code>warmer_queue</code> | - | float | - |
| <code>warmer_rejected</code> | - | float | - |
| <code>warmer_threads</code> | - | float | - |
| <code>watcher_active</code> | - | float | - |
| <code>watcher_completed</code> | - | float | - |
| <code>watcher_queue</code> | - | float | - |
| <code>watcher_rejected</code> | - | float | - |
| <code>watcher_threads</code> | - | float | - |
| <code>write_active</code> | - | float | - |
| <code>write_completed</code> | - | float | - |
| <code>write_largest</code> | - | float | - |
| <code>write_queue</code> | - | float | - |
| <code>write_rejected</code> | - | float | - |
| <code>write_threads</code> | - | float | - |

包含节点的线程池统计信息

threads: 线程池中的线程数。 queue: 线程池中队列中的任务数。 active: 线程池中活动线程的数量。
rejected: 线程池执行程序拒绝的任务数。 largest: 线程池中活动线程的最高数量。 completed: 线程池执行程序完成的任务数。

指标集 `elasticsearch_indices_stats_(primaries|total)`

前置条件

- 开启 `indices_stats` 设置

标签

| 标签名 | 描述 |
|-------------------------|----|
| <code>index_name</code> | |

指标

| 指标 | 描述 | 类型 | 单位 |
|---------------------------------------|----------------------|-------|----|
| <code>completion_size_in_bytes</code> | 用于完成分配给该节点的所有分片的内存总量 | float | - |

| | | | |
|---|---|-------|----|
| <code>docs_count</code> | Lucene报告的文档数。这 不包括已删除的文档，并 且将所有嵌套文档与其父 文档分开计数。它还不包 括最近被索引但尚未属于 段的文档 | float | - |
| <code>docs_deleted</code> | Lucene报告的已删除文档 的数量，它可能高于或低 于您执行的删除操作的数 量。该数字不包括最近执 行但尚未属于段的删除。 如果合理的话，将通过自 动合并过程清除已删除的 文档。此外， Elasticsearch会创建额外的 已删除文档，以在内部跟 踪分片上最近的操作历史 记录 | float | - |
| <code>fielddata_evictions</code> | 字段数据逐出的数量 | float | - |
| <code>fielddata_memory_size_in_bytes</code> | 用于分配给该节点的所有 分片上的字段数据缓存的 内存总量 | float | - |
| <code>flush_periodic</code> | 刷新定期操作的数量 | float | - |
| <code>flush_total</code> | 刷新操作的数量 | float | - |
| <code>flush_total_time_in_millis</code> | 执行刷新操作所花费的总 时间 | float | ms |
| <code>get_current</code> | 当前正在运行的get操作的 数量 | float | - |
| <code>get_exists_time_in_millis</code> | 执行成功的get操作所花费 的时间 | float | ms |
| <code>get_exists_total</code> | 成功获取操作的总数 | float | - |
| <code>get_missing_time_in_millis</code> | 执行失败的get操作所花费 的时间 | float | ms |
| <code>get_missing_total</code> | 执行失败的get操作的总数 | float | - |
| <code>get_time_in_millis</code> | 执行获取操作所花费的时 间 | float | ms |
| <code>get_total</code> | get操作的总数 | float | - |
| <code>indexing_delete_current</code> | 当前正在运行的删除操作 | float | - |

| | | | |
|--|----------------|-------|------|
| | 的数量 | | |
| <code>indexing_delete_time_in_millis</code> | 执行删除操作所花费的时间 | float | - |
| <code>indexing_delete_total</code> | 删除操作的总数 | float | - |
| <code>indexing_index_current</code> | 当前正在运行的索引操作数 | float | - |
| <code>indexing_index_failed</code> | 索引操作失败的次数 | float | - |
| <code>indexing_index_time_in_millis</code> | 执行索引操作所花费的总时间 | float | - |
| <code>indexing_index_total</code> | 索引操作的总数 | float | - |
| <code>indexing_is_throttled</code> | 节流操作次数 | float | - |
| <code>indexing_noop_update_total</code> | noop操作总数 | float | - |
| <code>indexing_throttle_time_in_millis</code> | 节流操作花费的总时间 | float | ms |
| <code>merges_current</code> | 当前正在运行的合并操作的数量 | float | - |
| <code>merges_current_docs</code> | 当前正在运行的文档合并数 | float | - |
| <code>merges_current_size_in_bytes</code> | 用于执行当前文档合并的内存 | float | byte |
| <code>merges_total</code> | 合并操作的总数 | float | - |
| <code>merges_total_auto_throttle_in_bytes</code> | 自动限制的合并操作的大小 | float | byte |
| <code>merges_total_docs</code> | 合并文档的总数 | float | - |
| <code>merges_total_size_in_bytes</code> | 文档合并的总大小 | float | - |
| <code>merges_total_stopped_time_in_millis</code> | 停止合并操作所花费的总时间 | float | ms |
| <code>merges_total_throttled_time_in_millis</code> | 限制合并操作花费的总时间 | float | ms |
| <code>merges_total_time_in_millis</code> | 执行合并操作所花费的总时间 | float | ms |
| <code>query_cache_cache_count</code> | 查询缓存中的查询计数 | float | - |
| <code>query_cache_cache_size</code> | 查询缓存的大小 | float | - |
| | | | |

| | | | |
|---------------------------------------|--------------------------|-------|------|
| active_primary_shards | 活跃主分片数 | float | - |
| query_cache_memory_size_in_bytes | 用于分配给该节点的所有分片上的查询缓存的内存总量 | float | byte |
| query_cache_hit_count | 查询缓存命中数 | float | - |
| query_cache_miss_count | 查询高速缓存未命中的数量 | float | - |
| query_cache_total_count | 查询缓存中的命中，未命中和缓存的查询的总数 | float | - |
| recovery_current_as_source | 恢复当前源的次数 | float | - |
| recovery_current_as_target | 恢复当前目标的次数 | float | - |
| recovery_throttle_time_in_millis | 恢复节流所花费的总时间 | float | ms |
| refresh_external_total | 外部刷新操作的总数 | float | - |
| refresh_external_total_time_in_millis | 外部刷新操作的总时间 | float | ms |
| refresh_listeners | 刷新侦听器的数量 | float | - |
| refresh_total | 刷新操作的总数 | float | - |
| refresh_total_time_in_millis | 执行刷新操作所花费的总时间 | float | ms |
| request_cache_evictions | 请求高速缓存的逐出数 | float | - |
| request_cache_hit_count | 请求缓存命中数 | float | - |
| request_cache_memory_size_in_bytes | 用于分配给该节点的所有分片上的请求缓存的内存总量 | float | byte |
| request_cache_miss_count | 请求高速缓存未命中的数量 | float | - |
| search_fetch_current | 当前正在运行的提取操作的数量 | float | - |
| search_fetch_time_in_millis | 执行提取操作所花费的时间 | float | ms |
| search_fetch_total | 提取操作的总数 | float | - |
| search_open_contexts | 开放搜索上下文的数量 | float | - |
| search_query_current | 当前正在运行的查询操作的数量 | float | - |
| | 执行查询操作所花费的时 | | |

| | | | |
|---|-----------------------------|-------|------|
| <code>search_query_time_in_millis</code> | 间 | float | - |
| <code>search_query_total</code> | 查询操作的总数 | float | - |
| <code>search_scroll_current</code> | 当前正在运行的滚动操作的数量 | float | - |
| <code>search_scroll_time_in_millis</code> | 执行滚动操作所花费的时间 | float | - |
| <code>search_scroll_total</code> | 滚动操作的总数 | float | - |
| <code>search_suggest_current</code> | 当前正在运行的建议操作的数量 | float | - |
| <code>search_suggest_time_in_millis</code> | 执行建议操作所花费的时间 | float | - |
| <code>search_suggest_total</code> | 建议操作总数 | float | - |
| <code>segments_count</code> | 段数 | float | - |
| <code>segments_doc_values_memory_in_bytes</code> | 用于分配给该节点的所有分片上的doc值的内存总量 | float | byte |
| <code>segments_fixed_bit_set_memory_in_bytes</code> | 由分配给节点的所有分片上的固定位集使用的内存总量 | float | byte |
| <code>segments_index_writer_memory_in_bytes</code> | 所有索引写程序在分配给该节点的所有分片上使用的内存总量 | float | byte |
| <code>segments_max_unsafe_auto_id_timestamp</code> | 最近重试的索引请求的时间 | float | ms |
| <code>segments_memory_in_bytes</code> | 用于分配给节点的所有分片上的段的内存总量 | float | byte |
| <code>segments_norms_memory_in_bytes</code> | 用于分配给节点的所有分片上的标准化因子的内存总量 | float | byte |
| <code>segments_points_memory_in_bytes</code> | 用于分配给节点的所有分片上的点的内存总量 | float | byte |
| <code>segments_stored_fields_memory_in_bytes</code> | 用于分配给节点的所有分片上的存储字段的内存总量 | float | byte |
| <code>segments_term_vectors_memory_in_bytes</code> | 用于分配给该节点的所有分片上的术语向量的内存总量 | float | byte |

| | | | |
|---|---------------------------|-------|------|
| <code>segments_terms_memory_in_bytes</code> | 用于分配给该节点的所有分片上的术语的内存总量 | float | byte |
| <code>segments_version_map_memory_in_bytes</code> | 所有版本映射在分配给节点的所有分片上使用的内存总量 | float | byte |
| <code>store_size_in_bytes</code> | 分配给节点的所有分片的总大小 | float | byte |
| <code>translog_earliest_last_modified_age</code> | 事务日志的最早最近修改年龄 | float | - |
| <code>translog_operations</code> | 事务日志操作数 | float | - |
| <code>translog_size_in_bytes</code> | 事务日志的大小 | float | - |
| <code>translog_uncommitted_operations</code> | 未提交的事务日志操作数 | float | - |
| <code>translog_uncommitted_size_in_bytes</code> | 未提交的事务日志操作的大小 | float | byte |
| <code>warmer_current</code> | 活动索引加热器的数量 | float | - |
| <code>warmer_total</code> | 指数加热器的总数 | float | - |
| <code>warmer_total_time_in_millis</code> | 执行索引预热操作所花费的总时间 | float | ms |

指标集 `elasticsearch_indices_stats_shards_total`

前置条件

- 开启 `shards_stats` 设置

标签

无

指标

| 指标 | 描述 | 类型 | 单位 | Tag |
|-------------------------|----|-------|----|-----|
| <code>failed</code> | - | float | - | - |
| <code>successful</code> | - | float | - | - |
| <code>total</code> | - | float | - | - |

指标集 `elasticsearch_indices_stats_shards`

标签

| 标签名 | 描述 |
|------------|----|
| index_name | |
| index_name | |
| node_name | |
| shard_name | |
| type | |

指标

| 指标 | 描述 | 类型 | 单位 |
|--------------------------------|---|-------|----|
| completion_size_in_bytes | 用于完成分配给该节点的所有分片的内存总量 | float | - |
| docs_count | Lucene报告的文档数。这 不包括已删除的文档，并 且将所有嵌套文档与其父 文档分开计数。它还不包 括最近被索引但尚未属于 段的文档 | float | - |
| docs_deleted | Lucene报告的已删除文档 的数量，它可能高于或低 于您执行的删除操作的数 量。该数字不包括最近执 行但尚未属于段的删除。 如果合理的话，将通过自 动合并过程清除已删除的 文档。此外， Elasticsearch会创建额外的 已删除文档，以在内部跟 踪分片上最近的操作历史 记录 | float | - |
| fielddata_evictions | 字段数据逐出的数量 | float | - |
| fielddata_memory_size_in_bytes | 用于分配给该节点的所有 分片上的字段数据缓存的 内存总量 | float | - |
| flush_periodic | 刷新定期操作的数量 | float | - |
| flush_total | 刷新操作的数量 | float | - |

| | | | |
|---|------------------|-------|------|
| <code>flush_total_time_in_millis</code> | 执行刷新操作所花费的总时间 | float | ms |
| <code>get_current</code> | 当前正在运行的get操作的数量 | float | - |
| <code>get_exists_time_in_millis</code> | 执行成功的get操作所花费的时间 | float | ms |
| <code>get_exists_total</code> | 成功获取操作的总数 | float | - |
| <code>get_missing_time_in_millis</code> | 执行失败的get操作所花费的时间 | float | ms |
| <code>get_missing_total</code> | 执行失败的get操作的总数 | float | - |
| <code>get_time_in_millis</code> | 执行获取操作所花费的时间 | float | ms |
| <code>get_total</code> | get操作的总数 | float | - |
| <code>indexing_delete_current</code> | 当前正在运行的删除操作的数量 | float | - |
| <code>indexing_delete_time_in_millis</code> | 执行删除操作所花费的时间 | float | - |
| <code>indexing_delete_total</code> | 删除操作的总数 | float | - |
| <code>indexing_index_current</code> | 当前正在运行的索引操作数 | float | - |
| <code>indexing_index_failed</code> | 索引操作失败的次数 | float | - |
| <code>indexing_index_time_in_millis</code> | 执行索引操作所花费的总时间 | float | - |
| <code>indexing_index_total</code> | 索引操作的总数 | float | - |
| <code>indexing_is_throttled</code> | 节流操作次数 | float | - |
| <code>indexing_noop_update_total</code> | noop操作总数 | float | - |
| <code>indexing_throttle_time_in_millis</code> | 节流操作花费的总时间 | float | ms |
| <code>merges_current</code> | 当前正在运行的合并操作的数量 | float | - |
| <code>merges_current_docs</code> | 当前正在运行的文档合并数 | float | - |
| <code>merges_current_size_in_bytes</code> | 用于执行当前文档合并的内存 | float | byte |
| | | | |

| | | | |
|---------------------------------------|--------------------------|-------|------|
| merges_total | 合并操作的总数 | float | - |
| merges_total_auto_throttle_in_bytes | 自动限制的合并操作的大小 | float | byte |
| merges_total_docs | 合并文档的总数 | float | - |
| merges_total_size_in_bytes | 文档合并的总大小 | float | - |
| merges_total_stopped_time_in_millis | 停止合并操作所花费的总时间 | float | ms |
| merges_total_throttled_time_in_millis | 限制合并操作花费的总时间 | float | ms |
| merges_total_time_in_millis | 执行合并操作所花费的总时间 | float | ms |
| query_cache_cache_count | 查询缓存中的查询计数 | float | - |
| query_cache_cache_size | 查询缓存的大小 | float | - |
| active_primary_shards | 活跃主分片数 | float | - |
| query_cache_memory_size_in_bytes | 用于分配给该节点的所有分片上的查询缓存的内存总量 | float | byte |
| query_cache_hit_count | 查询缓存命中数 | float | - |
| query_cache_miss_count | 查询高速缓存未命中的数量 | float | - |
| query_cache_total_count | 查询缓存中的命中，未命中和缓存的查询的总数 | float | - |
| recovery_current_as_source | 恢复当前源的次数 | float | - |
| recovery_current_as_target | 恢复当前目标的次数 | float | - |
| recovery_throttle_time_in_millis | 恢复节流所花费的总时间 | float | ms |
| refresh_external_total | 外部刷新操作的总数 | float | - |
| refresh_external_total_time_in_millis | 外部刷新操作的总时间 | float | ms |
| refresh_listeners | 刷新侦听器的数量 | float | - |
| refresh_total | 刷新操作的总数 | float | - |
| refresh_total_time_in_millis | 执行刷新操作所花费的总时间 | float | ms |
| request_cache_evictions | 请求高速缓存的逐出数 | float | - |
| request_cache_hit_count | 请求缓存命中数 | float | - |

| | | | |
|---|-----------------------------|-------|------|
| <code>request_cache_memory_size_in_bytes</code> | 用于分配给该节点的所有分片上的请求缓存的内存总量 | float | byte |
| <code>request_cache_miss_count</code> | 请求高速缓存未命中的数量 | float | - |
| <code>search_fetch_current</code> | 当前正在运行的提取操作的数量 | float | - |
| <code>search_fetch_time_in_millis</code> | 执行提取操作所花费的时间 | float | ms |
| <code>search_fetch_total</code> | 提取操作的总数 | float | - |
| <code>search_open_contexts</code> | 开放搜索上下文的数量 | float | - |
| <code>search_query_current</code> | 当前正在运行的查询操作的数量 | float | - |
| <code>search_query_time_in_millis</code> | 执行查询操作所花费的时间 | float | - |
| <code>search_query_total</code> | 查询操作的总数 | float | - |
| <code>search_scroll_current</code> | 当前正在运行的滚动操作的数量 | float | - |
| <code>search_scroll_time_in_millis</code> | 执行滚动操作所花费的时间 | float | - |
| <code>search_scroll_total</code> | 滚动操作的总数 | float | - |
| <code>search_suggest_current</code> | 当前正在运行的建议操作的数量 | float | - |
| <code>search_suggest_time_in_millis</code> | 执行建议操作所花费的时间 | float | - |
| <code>search_suggest_total</code> | 建议操作总数 | float | - |
| <code>segments_count</code> | 段数 | float | - |
| <code>segments_doc_values_memory_in_bytes</code> | 用于分配给该节点的所有分片上的doc值的内存总量 | float | byte |
| <code>segments_fixed_bit_set_memory_in_bytes</code> | 由分配给节点的所有分片上的固定位集使用的内存总量 | float | byte |
| <code>segments_index_writer_memory_in_bytes</code> | 所有索引写程序在分配给该节点的所有分片上使用的内存总量 | float | byte |
| | | | |

| | | | |
|---|---------------------------|-------|------|
| <code>segments_max_unsafe_auto_id_timestamp</code> | 最近重试的索引请求的时间 | float | ms |
| <code>segments_memory_in_bytes</code> | 用于分配给节点的所有分片上的段的内存总量 | float | byte |
| <code>segments_norms_memory_in_bytes</code> | 用于分配给节点的所有分片上的标准化因子的内存总量 | float | byte |
| <code>segments_points_memory_in_bytes</code> | 用于分配给节点的所有分片上的点的内存总量 | float | byte |
| <code>segments_stored_fields_memory_in_bytes</code> | 用于分配给节点的所有分片上的存储字段的内存总量 | float | byte |
| <code>segments_term_vectors_memory_in_bytes</code> | 用于分配给该节点的所有分片上的术语向量的内存总量 | float | byte |
| <code>segments_terms_memory_in_bytes</code> | 用于分配给该节点的所有分片上的术语的内存总量 | float | byte |
| <code>segments_version_map_memory_in_bytes</code> | 所有版本映射在分配给节点的所有分片上使用的内存总量 | float | byte |
| <code>store_size_in_bytes</code> | 分配给节点的所有分片的总大小 | float | byte |
| <code>translog_earliest_last_modified_age</code> | 事务日志的最早最近修改年龄 | float | - |
| <code>translog_operations</code> | 事务日志操作数 | float | - |
| <code>translog_size_in_bytes</code> | 事务日志的大小 | float | - |
| <code>translog_uncommitted_operations</code> | 未提交的事务日志操作数 | float | - |
| <code>translog_uncommitted_size_in_bytes</code> | 未提交的事务日志操作的大小 | float | byte |
| <code>warmer_current</code> | 活动索引加热器的数量 | float | - |
| <code>warmer_total</code> | 指数加热器的总数 | float | - |
| <code>warmer_total_time_in_millis</code> | 执行索引预热操作所花费的总时间 | float | ms |

监控指标说明

1.集群运行状况和节点可用性

| 指标描述 | 名称 | 度量标准 |
|----------------|--|------|
| 集群状态（绿色，黄色，红色） | <code>cluster_health.status</code> | 其他 |
| 数据节点数 | <code>cluster_health.number_of_data_nodes</code> | 可用性 |
| 初始化分片数 | <code>cluster_health.initializing_shards</code> | 可用性 |
| 未分配分片数 | <code>cluster_health.unassigned_shards</code> | 可用性 |
| 活跃分片数 | <code>cluster_health.active_shards</code> | 可用性 |
| 迁移中的分片数 | <code>cluster_health.relocating_shards</code> | 可用性 |

集群运行状况和节点可用性的要点

- **集群状态:** 如果集群状态为YELLOW，说明至少有一个副本分片未分配或者丢失。尽管这个时候搜索结果仍然是完整的，但是如果更多的分片消失的话，有可能造成整个索引的数据丢失。如果集群状态为RED，则表示至少有一个主分片丢失，索引缺少数据，这意味着搜索将返回部分结果，而且新的文档数据也无法索引到该分片。这时可以考虑设置一个告警，如果状态为黄色超过 5 分钟，或者上次检测状态为红色，则触发警报。
- **初始化（initializing）和未分配（unassigned）状态的分片:** 当索引首次创建或者节点重新启动时，由于 Master节点试图将分片分配给Data节点，所以在转换为“started”或“unassigned”状态之前，该分片将暂时处于“initializing”状态。如果您看到分片处于初始化状态或未分配状态的时间过长，则可能是群集不稳定的信号。

2.主机的网络和系统

| 指标描述 | 名称 | 度量标准 |
|-----------|--|-------|
| 可用磁盘空间 | <code>disk.used</code> <code>disk.total</code> | 资源利用率 |
| 内存利用率 | <code>os.mem_used_percent</code> | 资源利用率 |
| CPU使用率 | <code>os.cpu_percent</code> | 资源利用率 |
| 网络字节发送/接收 | <code>transport.tx_size_in_bytes</code> <code>transport.rx_size_in_bytes</code> | 资源利用率 |
| 打开文件描述符 | <code>clusterstats_nodes.process_open_file_descriptors_avg</code> | 资源利用率 |
| HTTP链接数 | <code>http.current_open</code> | 资源利用率 |

主机的网络和系统的要点

- 磁盘空间:** 如果当 Elasticsearch 集群是写负载型的，那么这个指标将变得更加重要。因为一旦空间不足，将不能进行任何插入和更新操作，节点也会下线，这应该是业务上不允许的。如果节点的可用空间小于 20%，应该利用类似 Curator 这样的工具，删除一些占用空间较大的索引来释放部分空间。如果业务逻辑上不允许删除索引，那么另一种方案就是扩容更多的节点，让 Master 在新节点间重新分配分片（尽管这样会增加 Master 节点的负载）。另外需要记住一点，包含需要分析（analyzed）的字段的文档占用的磁盘空间比那些不需要分析（non-analyzed）的字段（精确值）的文档要多得多。

- **节点上的CPU利用率:** 利用图示来展示不同节点类型的 CPU 使用情况会很有帮助。例如，可以创建三个不同的图来表示群集中的不同节点类型（例如 Data 节点，Master 节点和 Client 节点）的 CPU 使用情况，通过对比图示来发现是否某一种类型的节点过载了。如果您看到 CPU 使用率增加，这通常是由于大量的搜索或索引工作造成的负载。设置一个通知，以确定您的节点的 CPU 使用率是否持续增长，并根据需要添加更多的节点来重新分配负载。
- **网络字节发送/接收:** 节点之间的通信是衡量群集是否平衡的关键组成部分。需要监视网络以确保网络正常运行，并且能够跟上群集的需求（例如，在节点间复制或分片的重新分配）。Elasticsearch 提供有关集群节点间通信的传输指标，但是也可以通过发送和接收的字节速率，来查看网络正在接收多少流量。
- **打开文件描述符:** 文件描述符用于节点间的通信、客户端连接和文件操作。如果打开的文件描述符达到系统的限制，那么新的连接和文件操作将不可用，直到有旧的被关闭。如果超过80%的可用文件描述符正在使用中，则可能需要增加系统的最大文件描述符计数。大多数 Linux 系统限制每个进程中只允许有 1024 个文件描述符。在生产中使用 Elasticsearch 时，应该将操作系统文件描述符数量设置为更大的值，例如 64,000。
- **HTTP链接:** 除了 Java Client 以外的任何语言发送的请求都将使用基于 HTTP 的 RESTful API 与 Elasticsearch 进行通信。如果打开的 HTTP 连接总数不断增加，则可能表明您的 HTTP 客户端没有正确建立持久化连接。重新建立连接会在请求响应时间内增加额外的时间开销。因此务必确保您的客户端配置正确，以避免对性能造成影响，或者使用已正确配置 HTTP 连接的官方 Elasticsearch 客户端之一。

3. 查询性能指标

如果您主要使用 Elasticsearch 进行查询，那么您应该关注查询延迟并在超出阈值时采取措施。监控 Query 和 Fetch 的相关指标可以帮助您跟踪查询在一段时间内的执行情况。例如，您可能需要跟踪查询曲线的峰值以及长期的查询请求增长趋势，以便可以优化配置来获得更好的性能和可靠性。

| 指标描述 | 名称 | 度量标准 |
|--------------|--|------|
| 集群查询操作的总数 | <code>indices.search_query_total</code> | 吞吐量 |
| 集群查询操作总耗时 | <code>indices.search_query_time_in_millis</code> | 性能表现 |
| 集群当前正在进行的查询数 | <code>indices.search_query_current</code> | 吞吐量 |
| 集群获取操作的总数 | <code>indices.search_fetch_total</code> | 吞吐量 |
| 集群获取操作总耗时 | <code>indices.search_fetch_time_in_millis</code> | 性能表现 |
| 集群当前正在进行的获取数 | <code>indices.search_fetch_current</code> | 吞吐量 |

搜索性能指标的要点:

- **查询（Query）负载:** 监控当前查询并发数可以大致了解集群在某个时刻处理的请求数量。对不寻常的峰值低谷的关注，也许能发现一些潜在的风险。可能还需要监控查询线程池队列的使用情况。
- **查询（Query）延迟:** 虽然 Elasticsearch 并没有直接提供这个指标，但是我们可以通过定期采样查询请求总数除以所耗费的时长来简单计算平均查询延迟时间。如果超过我们设定的某个阈值，就需要排查潜在的资源瓶颈或者优化查询语句。
- **获取（Fetch）延迟:** 查询（search）过程的第二阶段，即获取阶段，通常比查询（query）阶段花费更

少的时间。如果您注意到此度量指标持续增加，则可能表示磁盘速度慢、富文档化（比如文档高亮处理等）或请求的结果太多等问题。

4.索引性能指标

索引（Indexing）请求类似于传统数据库里面的写操作。如果您的 Elasticsearch 集群是写负载类型的，那么监控和分析索引（indices）更新的性能和效率就变得很重要。在讨论这些指标之前，我们先看一下 Elasticsearch 更新索引的过程。如果索引发生了变化（比如新增了数据、或者现有数据需要更新或者删除），索引的每一个相关分片都要经历如下两个过程来实现更新操作：refresh 和 flush。

| 指标描述 | 名称 | 度量标准 |
|--------------|--|------|
| 索引的文档总数 | <code>indices.indexing_index_total</code> | 吞吐量 |
| 索引文档花费的总时间 | <code>indices.indexing_index_time_in_millis</code> | 性能表现 |
| 索引平均获取延迟 | <code>indices.search_fetch_time_in_millis</code> | 性能表现 |
| 索引平均查询延迟 | <code>indices.search_query_time_in_millis</code> | 性能表现 |
| 当前正在编制索引的文档数 | <code>indices.indexing_index_current</code> | 吞吐量 |
| 索引刷新总数 | <code>indices.refresh_total</code> | 吞吐量 |
| 刷新索引所花费的总时间 | <code>indices.refresh_total_time_in_millis</code> | 性能表现 |
| 刷新到磁盘的索引总数 | <code>indices.flush_total</code> | 吞吐量 |

| | | |
|-----------------|--|------|
| 将索引刷新到磁盘上花费的总时间 | <code>indices.flush_total_time_in_millis</code> | 性能表现 |
| 索引合并文档数 | <code>indices.merges_current_docs</code> | 吞吐量 |
| 索引合并花费时间 | <code>indices.merges_total_stopped_time_in_millis</code> | 性能表现 |
| 等待处理任务数 | <code>indices.number_of_pending_tasks</code> | 吞吐量 |

索引性能指标的要点:

- **索引 (Indexing) 延迟:** Elasticsearch 并未直接提供这个指标，但是可以通过计算 `indextotal` 和 `indextimeinmillis` 来获取平均索引延迟。如果您发现这个指标不断攀升，可能是因为一次性 bulk 了太多的文档。Elasticsearch 推荐的单个 bulk 的文档大小在 5-15MB 之间，如果资源允许，可以从这个值慢慢加大到合适的值。
- **Flush 延迟:** 由于 Elasticsearch 是通过 flush 操作来将数据持久化到磁盘的，因此关注这个指标非常有用，可以在必要的时候采取一些措施。比如如果发现这个指标持续稳定增长，则可能说明磁盘 I/O 能力已经不足，如果持续下去最终将导致无法继续索引数据。此时您可以尝试适当调低 `index.translog.flushthresholdsize` 的值，来减小触发 flush 操作的 translog 大小。与此同时，如果您的集群是一个典型的 write-heavy 系统，您应该利用 iostat 这样的工具持续监控磁盘的 IO，必要的时候可以考虑升级磁盘类型。

5.内存使用和GC指标

在 Elasticsearch 运行过程中，内存是需要密切关注的关键资源之一。Elasticsearch 和 Lucene 以两种方式利用节点上的所有可用 RAM：JVM 堆和文件系统缓存。Elasticsearch 运行在 Java 虚拟机 (JVM) 上，这意味着 JVM 垃圾回收的持续时间和频率将是其他重要的监控领域。

| 指标描述 | 名称 | 度量标准 |
|--------------|--|-------|
| 青年代垃圾收集数 | <code>jvm.gc_collectors_young_collection_count</code> | - |
| 老年代垃圾收集数 | <code>jvm.gc_collectors_old_collection_count</code> | - |
| 年轻代垃圾收集总时间 | <code>jvm.gc_collectors_young_collection_time_in_millis</code> | - |
| 老年代垃圾收集总时间 | <code>jvm.gc_collectors_old_collection_time_in_millis</code> | - |
| 当前JVM堆内存的百分比 | <code>jvm.mem_heap_used_percent</code> | 资源利用率 |
| 提交的JVM堆内存大小 | <code>jvm.mem_heap_committed_in_bytes</code> | 资源利用率 |

内存使用和GC指标的要点:

- JVM堆内存的使用量:** Elasticsearch 默认当 JVM 堆栈使用率达到 75%的时候启动垃圾回收。因此监控节点堆栈使用情况并设置告警阈值来定位哪些节点的堆栈使用率持续维持在 85%变得非常有用，这表明垃圾回收的速度已经赶不上垃圾产生的速度了。要解决这个问题，可以增加堆栈大小，或者通过添加更多的节点来扩展群集。
- JVM堆内存的使用量和提交的JVM堆内存大小:** 和 JVM 堆栈分配了多少内存（committed）相比，监控 JVM 使用了多少内存（used）会更加有用。使用中的堆栈内存的曲线通常会呈锯齿状，在垃圾累积时逐渐上升在垃圾回收时又会下降。如果这个趋势随着时间的推移开始向上倾斜，这意味着垃圾回收的速度跟不上创建对象的速度，这可能导致垃圾回收时间变慢，最终导致 OutOfMemoryError。
- 垃圾收集持续时间和频率:** 为了收集无用的对象信息，JVM 会暂停执行所有任务，通常这种状态被称为“Stop the world”，不管是 young 还是 old 垃圾回收器都会经历这个阶段。由于 Master 节点每隔 30s 检测一下其他节点的存活状态，如果某个节点的垃圾回收时长超过这个时间，就极可能被 Master 认为该节点已经失联。
- 内存使用情况:** Elasticsearch 可以很好地使用任何尚未分配给 JVM 堆的 RAM。和 Kafka 一样，Elasticsearch 被设计为依靠操作系统的文件系统缓存来快速可靠地处理请求。如果某个 segment 最近由 Elasticsearch 写入磁盘，那么它已经在缓存中。但是，如果一个节点已被关闭并重启，则在第一次查询一个 segment 的时候，必须先从磁盘读取数据。所以这是确保群集保持稳定并且节点不会崩溃的重要原因之一。通常，监视节点上的内存使用情况非常重要，并尽可能为 Elasticsearch 提供更多的 RAM，以便在不溢出的情况下最大化利用文件系统缓存。

6.资源饱和度和错误

Elasticsearch 节点使用线程池来管理线程对内存和 CPU 使用。由于线程池是根据处理器的核数自动配置的，因此调整它们通常是没有意义的。但是通过请求队列和请求被拒绝的情况，来确定你的节点是否够用是一个不错的主意。如果出现了状况，您可能需要添加更多的节点来处理所有的并发请求。

- **线程池的请求队列（queues）和拒绝情况（rejections）**：每个 Elasticsearch 节点都维护着很多类型的线程池。具体应该监控哪些线程池，需要根据 Elasticsearch 的使用场景而定。一般来讲，最重要的几个线程池是搜索（search），索引（index），合并（merger）和批处理（bulk）。每个线程池队列的大小代表着当前节点有多少请求正在等待服务。队列的作用是允许节点追踪并最终处理这些请求，而不是直接丢弃它们。但是线程池队列不是无限扩展的（队列越大占用的内存越多），一旦线程池达到最大队列大小（不同类型的线程池的默认值不一样），后面的请求都会被线程池拒绝。

| 指标描述 | 名称 | 度量标准 |
|-------------|--|------|
| 线程池中排队的线程数 | <code>thread_pool.rollup_indexing_queue</code> <code>thread_pool.search_queue</code> <code>thread_pool.transform_indexing_queue</code> <code>thread_pool.force_merge_queue</code> | 饱和度 |
| 线程池中被拒绝的线程数 | <code>thread_pool.rollup_indexing_rejected</code> <code>thread_pool.transform_indexing_rejected</code> <code>thread_pool.search_rejected</code> <code>thread_pool.force_merge_rejected</code> | 错误 |

资源饱和度和错误的要点：

- **线程池队列**：线程池队列并不是越大越好，因为线程池越大占用的资源越多，并且增大了节点宕机时请求丢失的风险。如果您看到排队和拒绝的线程数量持续增加，则需要尝试减慢请求速率、增加节点上的处理器数量或增加集群中的节点数量。
- **批处理（bulk）的请求队列和请求拒绝**：批处理是同时发送多个请求的更有效方式。通常如果要执行许多操作（创建索引，或者添加，更新或删除文档），则应尝试将请求作为批处理发送，而不是多个单独的请求。批处理请求被拒绝通常与试图在一个批处理请求中索引太多文档有关。虽然据 Elasticsearch 的文档所说，出现批处理请求被拒绝的情况，不一定是必须要担心的事情，但是应该尝试实施一些退避策略，以有效处理这种情况。
- **缓存使用率指标**：每个查询请求都发送到索引中的每个分片，然后命中每个分片的每个段。Elasticsearch 逐段缓存查询，以加快响应时间。另一方面，如果您的缓存占用了太多的堆，它们可能会降低速度而不是加快速度！