

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

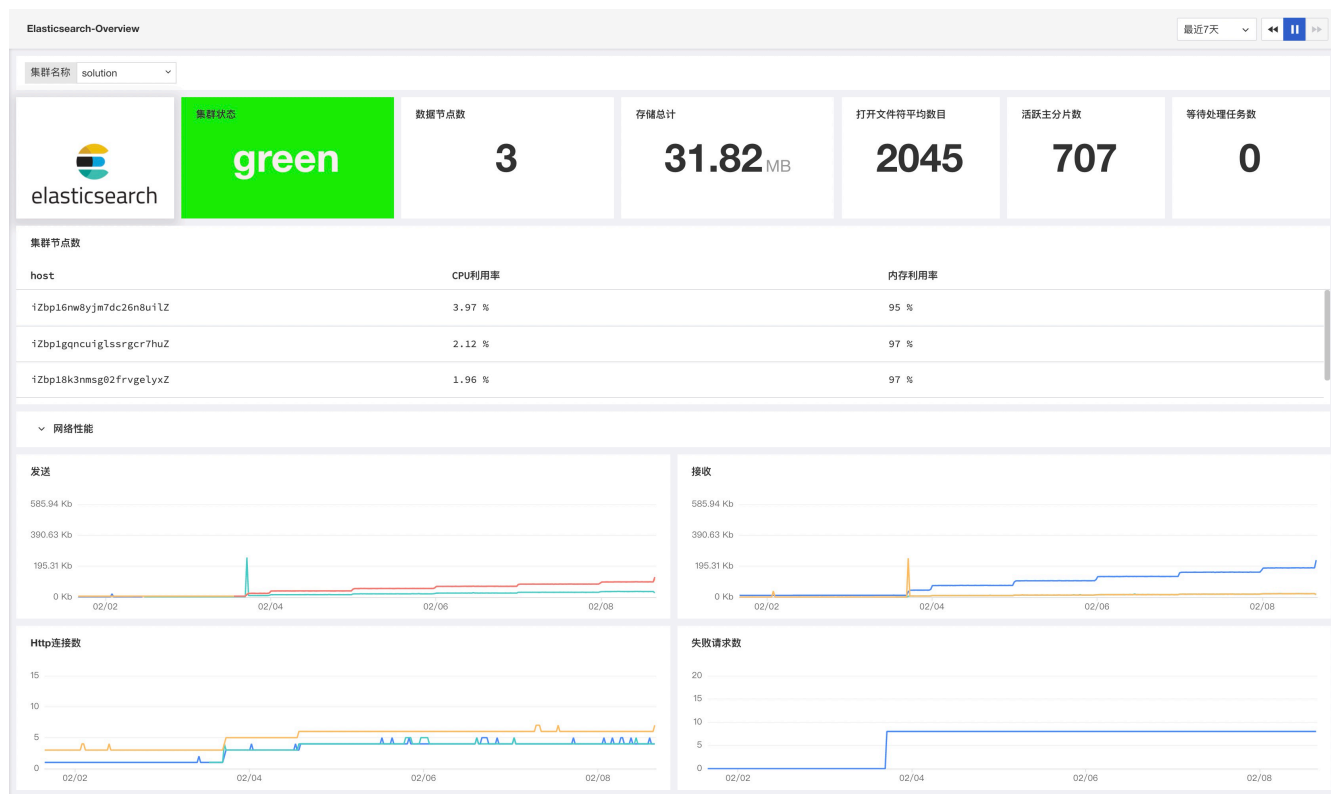
tags: IT运维,指标采集

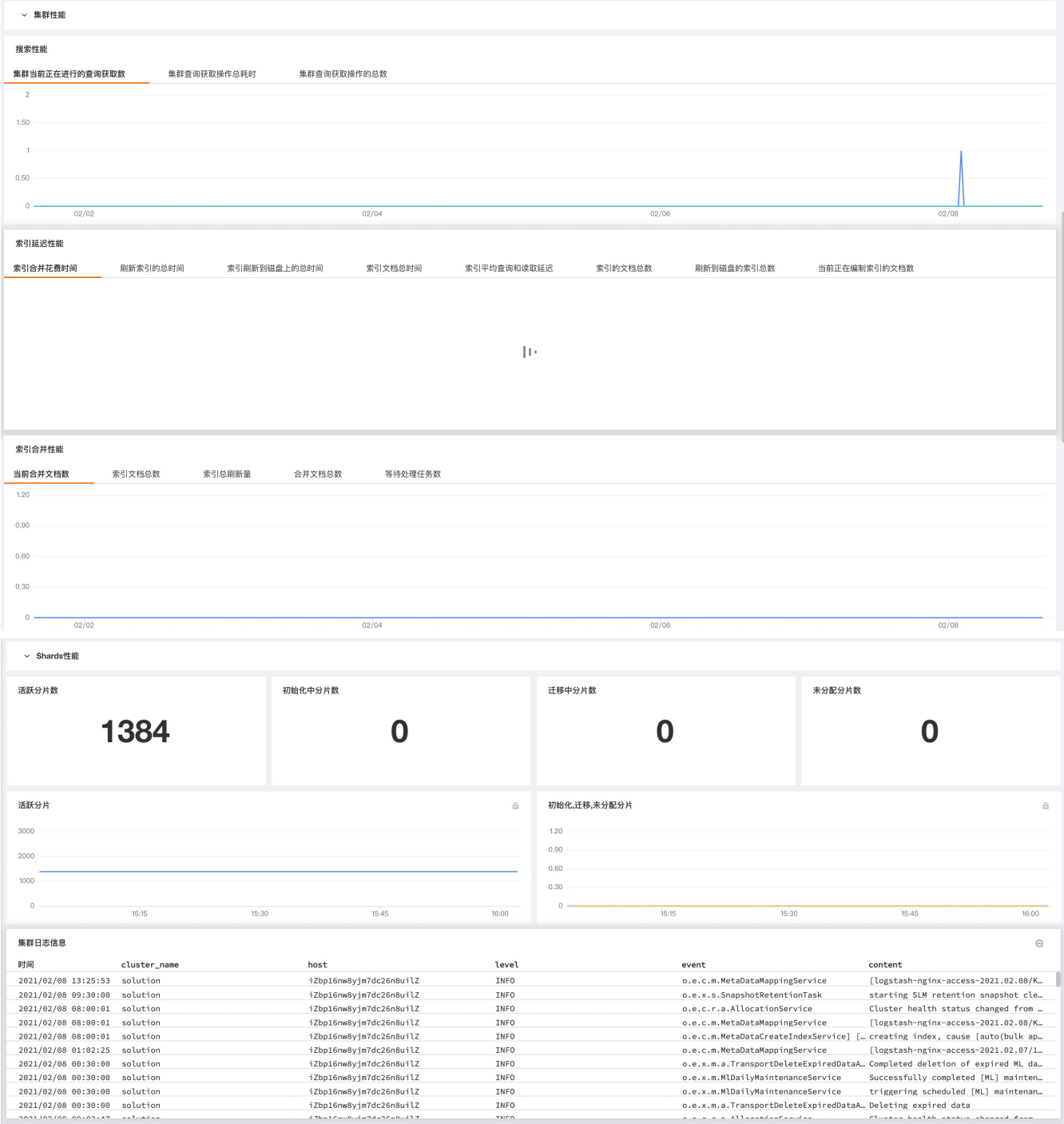
## 简介

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

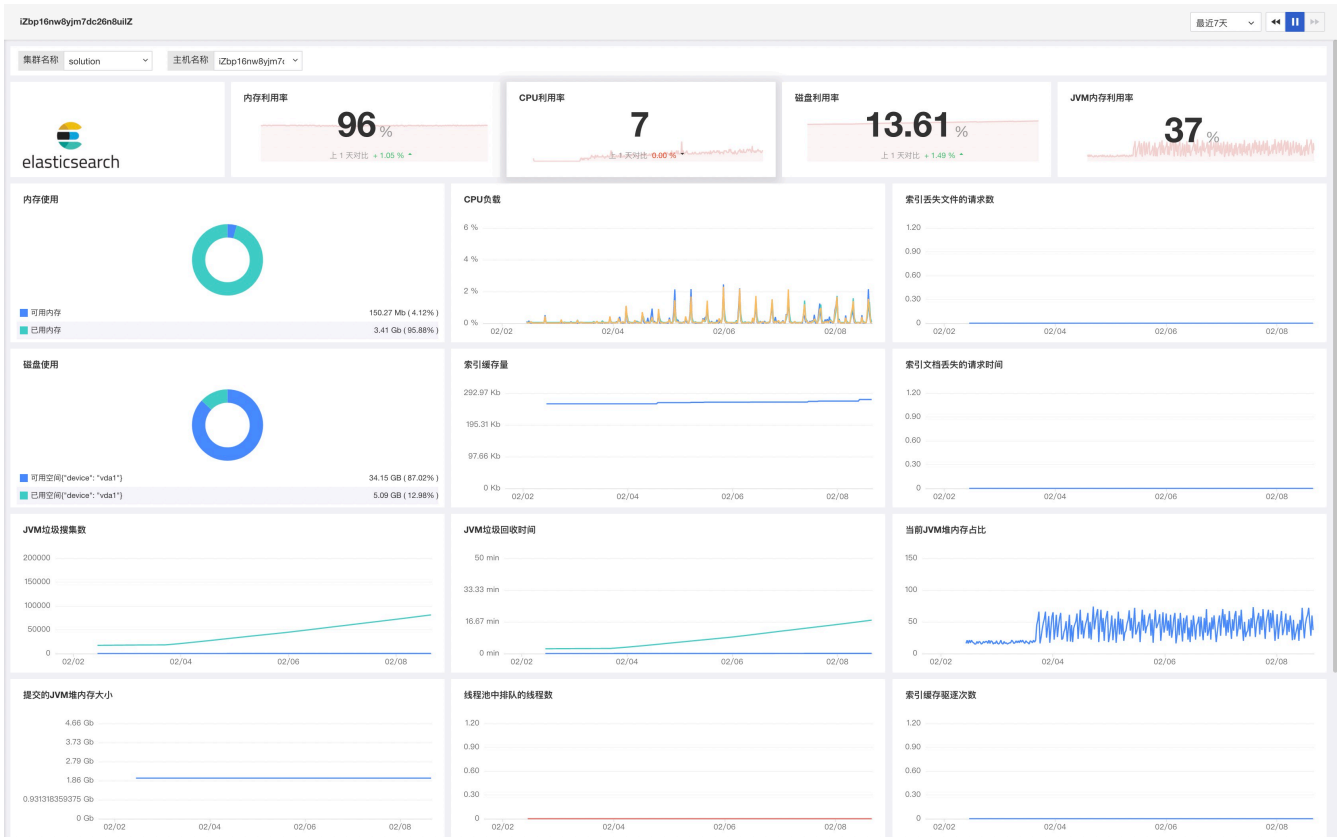
## 场景视图






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 = true
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改的数量	integer	-
<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
<code>name</code>	
<code>index</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>initializing_shards</code>	正在初始化的分片数	integer	-
<code>number_of_replicas</code>	复制中的分片数量	integer	-
<code>number_of_shards</code>	分区数量	integer	-
<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>unassigned_shards</code>	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
<code>cluster_name</code>	
<code>node_name</code>	
<code>status</code>	

指标

指标	描述	类型	单位
<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

query_cache_miss_count	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	-
query_cache_total_count	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	-
segments_count	分配给选定节点的所有分片上的分段总数	float	-
segments_doc_values_memory_in_bytes	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
segments_fixed_bit_set_memory_in_bytes	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
segments_index_writer_memory_in_bytes	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
segments_max_unsafe_auto_id_timestamp	最近重试的索引请求的Unix时间戳	float	ms
segments_memory_in_bytes	用于分配给选定节点的所有分片上的段的内存总量	float	byte
segments_norms_memory_in_bytes	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
shards_index primaries_avg	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	-
shards_index primaries_max	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	-
shards_index primaries_min	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	-
shards_index_replication_avg	索引中的平均复制因子，仅计算分配给选定节点的分片	float	-
shards_index_replication_max	索引中的最大复制因子，仅计算分配给所选节点的分片	float	-
shards_index_replication_min	索引中的最小复制因子，仅计算分配给选定节点的分片	float	-
shards_index_shards_avg	索引中的最小分片数，仅计算分配给选定节点的分片	float	-
shards_index_shards_max	索引中最大的分片数，仅计算分配给选定节点的分片	float	-
shards_index_shards_min	索引中的最小分片数，仅计算分配给选定节点的分片	float	-
shards primaries	分配给所选节点的主分片数	float	-
shards_replication	所有选定节点上的副本分片与主分片的比率	float	-
shards_total	分配给所选节点的分片总数	float	-
store_size_in_bytes	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
cluster_name	
node_name	
status	

指标

指标	描述	类型	单位
count_coordinating_only	没有角色的选定节点数。这些节点仅被视为协调节点	float	-
count_data	数据节点数据量	float	-



count_ingest	查询节点数量	float	-
count_master	master节点数量	float	-
count_total	所选节点的总数	float	-
fs_available_in_bytes	所有选定节点上文件存储中可用于JVM的字节总数	float	byte
fs_free_in_bytes	所有选定节点上文件存储中未分配字节的总数	float	byte
fs_total_in_bytes	所有选定节点中所有文件存储的总大小	float	byte
jvm_max_uptime_in_millis	自JVM上次启动以来的正常运行时间	float	ms
jvm_mem_heap_max_in_bytes	可供所有选定节点上的堆使用的最大内存量	float	byte
jvm_mem_heap_used_in_bytes	所有选定节点上的堆当前正在使用的内存	float	byte
jvm_threads	JVM在所有选定节点上使用的活动线程数	float	-
jvm_versions_0_count	使用JVM的选定节点总数	float	-
jvm_versions_0_version	一个或多个选定节点使用的JVM版本	string	-
jvm_versions_0_vm_name	JVM的名称	string	-
jvm_versions_0_vm_vendor	JVM的供应商	string	-
jvm_versions_0_vm_version	JVM的完整版本号	string	-
network_types_http_types_security4	包含有关所选节点使用的HTTP网络类型的统计信息	float	-
network_types_transport_types_security4	包含有关所选节点使用的传输网络类型的统计信息	float	-
os_allocated_processors	用于计算所有选定节点上的线程池大小的处理器数量	float	-
os_available_processors	所有选定节点上可用于JVM的处理器数	float	-
os_mem_free_in_bytes	可用物理内存量	float	byte
os_mem_free_percent	可用内存的百分比	float	-
os_mem_total_in_bytes	物理内存总量	float	byte
os_mem_used_in_bytes	已使用的物理内存量	float	- byte
os_names_0_count	使用操作系统的所选节点数	float	-
os_names_0_name	一个或多个选定节点使用的操作系统的名称	float	-
os_pretty_names_0_count	使用操作系统的所选节点数	float	-
os_pretty_names_0_pretty_name	一个或多个选定节点使用的操作系统的可读名称	float	-
process_cpu_percent	使用操作系统的所选节点数	float	-
process_open_file_descriptors_avg	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	-
process_open_file_descriptors_max	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	-
process_open_file_descriptors_min	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	-
versions_0	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>	内部集群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
<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
accounting_estimated_size_in_bytes	-	float	byte
accounting_limit_size_in_bytes	-	float	byte
accounting_overhead	-	float	-
accounting_tripped	-	float	-
fielddata_estimated_size_in_bytes	-	float	byte
fielddata_limit_size_in_bytes	-	float	byte
fielddata_overhead	-	float	-
fielddata_tripped	-	float	-
in_flight_requests_estimated_size_in_bytes	-	float	byte
in_flight_requests_limit_size_in_bytes	-	float	byte
in_flight_requests_overhead	-	float	-
in_flight_requests_tripped	-	float	-
parent_estimated_size_in_bytes	-	float	byte
parent_limit_size_in_bytes	-	float	byte
parent_overhead	-	float	-
parent_tripped	-	float	-
request_estimated_size_in_bytes	-	float	byte
accounting_estimated_size_in_bytes	-	float	byte
request_limit_size_in_bytes	-	float	byte
request_overhead	-	float	-
request_tripped	-	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使用的所有设备上的读写操作总数	float	-
<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		

docs_deleted	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	byte
flush_periodic	刷新定期操作的数量	float	-
flush_total	刷新操作的数量	float	-
flush_total_time_in_millis	执行刷新操作所花费的总时间	float	ms
get_current	当前正在运行的get操作的数量	float	-
get_exists_time_in_millis	执行成功的get操作所花费的时间	float	ms
get_exists_total	成功获取操作的总数	float	-
get_missing_time_in_millis	执行失败的get操作所花费的时间	float	ms
get_time_in_millis	执行获取操作所花费的时间	float	ms
get_total	get操作的总数	float	-
indexing_delete_current	当前正在运行的删除操作的数量	float	-
indexing_delete_time_in_millis	执行删除操作所花费的时间	float	ms
indexing_delete_total	删除操作的总数	float	-
indexing_index_current	当前正在运行的索引操作数	float	-
indexing_index_failed	索引操作失败的次数	float	-
indexing_index_time_in_millis	执行索引操作所花费的总时间	float	ms
indexing_index_total	索引操作的总数	float	-
indexing_noop_update_total	noop操作总数	float	-
indexing_throttle_time_in_millis	节流操作花费的总时间	float	ms
merges_current	当前正在运行的合并操作的数量	float	-
merges_current_docs	当前正在运行的文档合并数	float	-
merges_current_size_in_bytes	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	byte
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	-
query_cache_evictions	查询缓存逐出的数量	float	-
query_cache_hit_count	查询缓存命中数	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_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_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	-
search_query_time_in_millis	执行查询操作所花费的时间	float	ms
search_query_total	查询操作的总数	float	-
search_scroll_current	当前正在运行的滚动操作的数量	float	-
search_scroll_time_in_millis	执行滚动操作所花费的时间	float	ms
search_scroll_total	滚动操作的总数	float	-
search_suggest_current	当前正在运行的建议操作的数量	float	-
search_suggest_time_in_millis	执行建议操作所花费的时间	float	ms
search_suggest_total	建议操作总数	float	-
segments_count	段数	float	-
segments_doc_values_memory_in_bytes	用于分配给该节点的所有分片上的doc值的内存总量	float	byte
segments_fixed_bit_set_memory_in_bytes	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
segments_index_writer_memory_in_bytes	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
segments_max_unsafe_auto_id_timestamp	最近重试的索引请求的时间	float	ms
segments_memory_in_bytes	用于分配给节点的所有分片上的段的内存总量	float	byte
segments_norms_memory_in_bytes	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
segments_points_memory_in_bytes	用于分配给节点的所有分片上的点的内存总量	float	byte
segments_stored_fields_memory_in_bytes	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
segments_term_vectors_memory_in_bytes	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

segments_terms_memory_in_bytes	用于分配给该节点的所有分片上的术语的内存总量	float	byte
segments_version_map_memory_in_bytes	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
store_size_in_bytes	分配给节点的所有分片的总大小	float	byte
translog_earliest_last_modified_age	事务日志的最早最近修改年龄	float	-
translog_operations	事务日志操作数	float	-
translog_size_in_bytes	事务日志的大小	float	byte
translog_uncommitted_operations	未提交的事务日志操作数	float	-
translog_uncommitted_size_in_bytes	未提交的事务日志操作的大小	float	byte
warmer_current	活动索引加热器的数量	float	-
warmer_total	指数加热器的总数	float	-
warmer_total_time_in_millis	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
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
buffer_pools_direct_count	直接缓冲池的数量	float	-
buffer_pools_direct_total_capacity_in_bytes	直接缓冲池的总容量	float	byte
buffer_pools_direct_used_in_bytes	直接缓冲池的大小	float	byte
buffer_pools_mapped_count	映射的缓冲池数	float	-
buffer_pools_mapped_total_capacity_in_bytes	映射的缓冲池的总容量	float	byte
buffer_pools_mapped_used_in_bytes	映射的缓冲池的大小	float	byte
classes_current_loaded_count	JVM当前加载的类的数量	float	-
classes_total_loaded_count	自JVM启动以来已加载的类总数	float	-
classes_total_unloaded_count	自JVM启动以来已卸载的类的总数	float	-
gc_collectors_old_collection_count	收集老年代对象的JVM垃圾收集器	float	-



	的数量		
<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
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>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，以字节为单位	float	byte
<code>swap_used_in_bytes</code>	已使用的交换空间量	float	byte
<code>timestamp</code>	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	-
warmer_queue	-	float	-
warmer_rejected	-	float	-
warmer_threads	-	float	-
watcher_active	-	float	-
watcher_completed	-	float	-
watcher_queue	-	float	-
watcher_rejected	-	float	-
watcher_threads	-	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



get_total	get操作的总数	float	-
indexing_delete_current	当前正在运行的删除操作的数量	float	-
indexing_delete_time_in_millis	执行删除操作所花费的时间	float	-
indexing_delete_total	删除操作的总数	float	-
indexing_index_current	当前正在运行的索引操作数	float	-
indexing_index_failed	索引操作失败的次数	float	-
indexing_index_time_in_millis	执行索引操作所花费的总时间	float	-
indexing_index_total	索引操作的总数	float	-
indexing_is_throttled	节流操作次数	float	-
indexing_noop_update_total	noop操作总数	float	-
indexing_throttle_time_in_millis	节流操作花费的总时间	float	ms
merges_current	当前正在运行的合并操作的数量	float	-
merges_current_docs	当前正在运行的文档合并数	float	-
merges_current_size_in_bytes	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	-
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	-
search_query_time_in_millis	执行查询操作所花费的时间	float	-
search_query_total	查询操作的总数	float	-
search_scroll_current	当前正在运行的滚动操作的数量	float	-
search_scroll_time_in_millis	执行滚动操作所花费的时间	float	-
search_scroll_total	滚动操作的总数	float	-
search_suggest_current	当前正在运行的建议操作的数量	float	-
search_suggest_time_in_millis	执行建议操作所花费的时间	float	-
search_suggest_total	建议操作总数	float	-
segments_count	段数	float	-
segments_doc_values_memory_in_bytes	用于分配给该节点的所有分片上的doc值的内存总量	float	byte
segments_fixed_bit_set_memory_in_bytes	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
segments_index_writer_memory_in_bytes	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
segments_max_unsafe_auto_id_timestamp	最近重试的索引请求的时间	float	ms
segments_memory_in_bytes	用于分配给节点的所有分片上的段的内存总量	float	byte
segments_norms_memory_in_bytes	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
segments_points_memory_in_bytes	用于分配给节点的所有分片上的点的内存总量	float	byte
segments_stored_fields_memory_in_bytes	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
segments_term_vectors_memory_in_bytes	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
segments_terms_memory_in_bytes	用于分配给该节点的所有分片上的术语的内存总量	float	byte
segments_version_map_memory_in_bytes	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
store_size_in_bytes	分配给节点的所有分片的总大小	float	byte
translog_earliest_last_modified_age	事务日志的最早最近修改年龄	float	-
translog_operations	事务日志操作数	float	-
translog_size_in_bytes	事务日志的大小	float	-
translog_uncommitted_operations	未提交的事务日志操作数	float	-
translog_uncommitted_size_in_bytes	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
<code>index_name</code>	
<code>index_name</code>	
<code>node_name</code>	
<code>shard_name</code>	
<code>type</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	-

flush_total	刷新操作的数量	float	-
flush_total_time_in_millis	执行刷新操作所花费的总时间	float	ms
get_current	当前正在运行的get操作的数量	float	-
get_exists_time_in_millis	执行成功的get操作所花费的时间	float	ms
get_exists_total	成功获取操作的总数	float	-
get_missing_time_in_millis	执行失败的get操作所花费的时间	float	ms
get_missing_total	执行失败的get操作的总数	float	-
get_time_in_millis	执行获取操作所花费的时间	float	ms
get_total	get操作的总数	float	-
indexing_delete_current	当前正在运行的删除操作的数量	float	-
indexing_delete_time_in_millis	执行删除操作所花费的时间	float	-
indexing_delete_total	删除操作的总数	float	-
indexing_index_current	当前正在运行的索引操作数	float	-
indexing_index_failed	索引操作失败的次数	float	-
indexing_index_time_in_millis	执行索引操作所花费的总时间	float	-
indexing_index_total	索引操作的总数	float	-
indexing_is_throttled	节流操作次数	float	-
indexing_noop_update_total	noop操作总数	float	-
indexing_throttle_time_in_millis	节流操作花费的总时间	float	ms
merges_current	当前正在运行的合并操作的数量	float	-
merges_current_docs	当前正在运行的文档合并数	float	-
merges_current_size_in_bytes	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	-
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	-
search_query_time_in_millis	执行查询操作所花费的时间	float	-
search_query_total	查询操作的总数	float	-
search_scroll_current	当前正在运行的滚动操作的数量	float	-
search_scroll_time_in_millis	执行滚动操作所花费的时间	float	-
search_scroll_total	滚动操作的总数	float	-
search_suggest_current	当前正在运行的建议操作的数量	float	-
search_suggest_time_in_millis	执行建议操作所花费的时间	float	-
search_suggest_total	建议操作总数	float	-
segments_count	段数	float	-
segments_doc_values_memory_in_bytes	用于分配给该节点的所有分片上的doc值的内存总量	float	byte
segments_fixed_bit_set_memory_in_bytes	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
segments_index_writer_memory_in_bytes	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
segments_max_unsafe_auto_id_timestamp	最近重试的索引请求的时间	float	ms
segments_memory_in_bytes	用于分配给节点的所有分片上的段的内存总量	float	byte
segments_norms_memory_in_bytes	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
segments_points_memory_in_bytes	用于分配给节点的所有分片上的点的内存总量	float	byte
segments_stored_fields_memory_in_bytes	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
segments_term_vectors_memory_in_bytes	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
segments_terms_memory_in_bytes	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>clusterhealth.numberofdatanodes</code>	可用性
初始化分片数	<code>clusterhealth.initializingshards</code>	可用性
未分配分片数	<code>clusterhealth.unassignedshards</code>	可用性
活跃分片数	<code>clusterhealth.activeshards</code>	可用性
迁移中的分片数	<code>clusterhealth.relocatingshards</code>	可用性

#### 集群运行状况和节点可用性的要点

- 集群状态:** 如果群集状态为黄色，表示至少一个副本分片未分配或丢失但是搜索结果仍将是完整的，但是如果更多碎片丢失，则可能会丢失数据。集群状态为红色时表明至少有一个主分片丢失，表示丢失了数据，这意味着搜索将返回部分结果。将无法索引到该分片。考虑设置警报以在状态为黄色超过5分钟或状态在过去一分钟内为红色时触发。
- 初始化和未分配的分片:** 首次创建索引或重新启动节点时，其分片将短暂进入“初始化”状态，然后过渡到“已启动”或“未分配”状态，因为主节点会尝试将分片分配给集群中的其他的节点。如果您看到碎片保持初始化或未分配状态的时间过长，则可能是群集不稳定的警告信号。

### 2. 主机的网络 and 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.memusedpercent</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.txsizeinbytes</code> <code>transport.rxsizeinbytes</code>	资源利用率
打开文件描述符	<code>clusterstatsnodes.processopenfiledescriptors_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 and 系统的要点

- **磁盘空间:** 如果Elasticsearch集群是大量写入操作，则此指标特别重要。我们肯定不希望磁盘空间用完，因为这样将无法插入或更新任何内容，并且该节点将挂掉。如果某个节点上的可用磁盘空间不足20%，则可能需要使用Curator之类的工具来删除该节点上驻留的某些索引，这些索引占用了过多的磁盘空间。如果无法删除索引，则另一种选择是添加更多节点，并让主节点负责在新节点之间自动重新分配分片（注意：这会为繁忙的主节点创建额外的工作）。此外，具有分析字段（需要进行文本分析的字段（需要加标记，删除标点符号等）的文档）所占用的磁盘空间比具有未分析字段（准确值）的文档要多得多。
- **节点上的CPU利用率:** 在每个节点的内置视图中，可视化CPU使用情况可能会有所帮助。以查看一种类型的节点相对于另一种类型是否活动过多。如果看到CPU使用率增加，通常是由于繁重的搜索或索引工作量引起的。设置一条通知，以了解节点的CPU使用率是否一直在增加，并在需要时添加更多节点以重新分配负载。
- **网络字节发送/接收:** 节点之间的通信是平衡群集的关键组成部分。需要监视网络，以确保其运行状况良好，并且能够满足群集的需求（例如：在节点之间复制碎片或重新平衡碎片时）。Elasticsearch提供了有关集群通信的传输指标，但您也可以查看发送和接收的字节率，以查看您的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到节点的通信，客户端连接和文件操作。如果该数目达到系统的最大容量，则在关闭旧连接和文件之前，将无法进行新的连接和文件操作。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数量。大多数Linux系统附带每个进程只允许1,024个文件描述符。在生产环境中使用Elasticsearch时，应将OS文件描述符计数重置为更大的值，例如64,000。
- **HTTP链接:** 以Java以外的任何语言发送的请求将使用restful API通过HTTP与Elasticsearch进行通信。如果打开的HTTP连接的总数不断增加，则表明您的HTTP客户端无法正确建立持久连接。重新建立连接将使您的请求响应时间增加毫秒甚至几秒钟。确保已正确配置客户端，以避免对性能造成负面影响，或使用已正确配置HTTP连接的官方Elasticsearch客户端。

### 3. 搜索性能指标

搜索请求是Elasticsearch中的两个主要请求类型之一，另一个是索引请求。这些请求有时类似于传统数据库系统中的读写请求。Elasticsearch提供与搜索过程的两个主要阶段（查询和获取）相对应的度量。如果使用Elasticsearch主要用于搜索，或者如果搜索是面向客户的功能。应该监视查询延迟和设定阈值。监控关于查询和提取的相关指标很重要，可以帮助确定搜索随时间的变化。例如，业务上可能希望跟踪查询请求的尖峰和长期增长，以便可以做好准备。

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:

- **查询负载:** 监视当前正在进行的查询数量可以使您大致了解群集在任何特定时间正在处理多少个请求。考虑对可能指示潜在问题的异常峰值或下降发出警报。
- **查询延迟:** 尽管Elasticsearch并未明确提供此指标，但是监视工具可以通过定期查询总数和经过的总时间来帮助使用可用的指标来计算平均查询延迟。如果延迟超过阈值，则设置警报，如果触发，则查找潜在的资源瓶颈，或者调查是否需要优化查询。
- **提取延迟:** 搜索过程的第二部分，即获取阶段，通常应比查询阶段花费更少的时间。如果发现此指标持续增加，则可能表示磁盘速度慢，文档过多（突出显示搜索结果中的相关文本等）或请求太多结果的问题。

### 4. 索引性能指标

索引请求类似于传统数据库系统中的写入请求，如果es的写入工作量很重，那么监控和分析您能够如何有效地使用新数据更新索引非常重要。在了解指标之前，让我们来探索Elasticsearch更新索引的过程，在新数据被添加进索引、更新或删除已有数据，索引中的每个shard都有两个过程：refresh 和 flush Index fresh 新索引的文档不能立马被搜索的。首先，它们被写入一个内存中的缓冲区（in-memory buffer），等待下一次索引刷新，默认情况下每秒一次。刷新是以in-memory buffer为基础创建in-memory segment的过程。这样索引进的文档才能是可被搜索的，创建完segment后，清空buffer。

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>	工作表现
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:

- **索引延迟:** Elasticsearch不会直接公开此特定指标，但是监控工具可以帮助您从可用的`indextotal`和`indextimeinmillis`指标计算平均索引延迟。如果注意到延迟增加，可能会一次尝试索引太多的文档（Elasticsearch的文档建议从5到15兆字节的批量索引大小开始，并缓慢增加）。如果您计划索引大量文档，并且不需要立即可用于搜索。则可以通过减少刷新频率来优化。索引设置API使能够暂时禁用刷间隔搜索，则可以通过减少刷新频率直到完成索引，来针对搜索性能优化索引性能。该指数设置API，可以暂时禁用的刷新间隔：

```
bash curl -XPUT <nameofhost>:9200/<name_of_index>/_settings -d '{ "index" : { "refresh_interval" : "-1" } }'
```

完成索引后，您可以恢复为默认值“1s”

- **刷新延迟:** 在flush完成之前，数据不会被固化到磁盘中。因此追踪flush latency很有用。比如我们看到这个指标稳步增长，表明磁盘性能不好。这个问题将最终导致无法向索引添加新的数据。可以尝试降低`index.translog.flushthresholdsize`。这个设置决定translog的最大值（在flush被触发前）

## 5.内存使用和GC指标

在运行Elasticsearch时，内存是要密切监控的关键资源之一。Elasticsearch和Lucene以两种方式利用节点上的所有可用RAM：JVM heap和文件系统缓存。Elasticsearch运行在Java虚拟机（JVM）中，这意味着JVM垃圾回收的持续时间和频率将成为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.collectorsyoungcollectioncount</code>	-
老年代垃圾收集数	<code>jvm.gc.collectorsoldcollectioncount</code>	-
年轻代垃圾收集总时间	<code>jvm.gc.collectorsyoungcollectiontimeinmillis</code>	-
老年代垃圾收集总时间	<code>jvm.gc.collectorsold.collectiontimein_millis</code>	-
当前JVM堆内存的百分比	<code>jvm.mem.heapused_percent</code>	资源利用率
提交的JVM堆内存大小	<code>jvm.mem.heapcommittedinbytes</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:** 当JVM heap 使用率达到75%时，es启动GC。可以监控node的JVM heap，并且设置一个警报，确认哪个节点是否一直超过%85。如果一直超过，则表明垃圾的收集已经跟不上垃圾的产生。此时可以通过增加heap（需要满足建议法则不超过32G），或者通过增加节点来扩展集群，分散压力。
- **JVM堆内存的使用量和提交的JVM堆内存大小:** 与提堆提交的内存（保证可用的数量）相比，了解当前正在使用多少JVM heap的情况可能会有所帮助。heap memory的图一般是个锯齿图，在垃圾收集的时候heap上升，当收集完成后heap下降。如果这个锯齿图向上偏移，说明垃圾的收集速度低于rate of object creation，这可能会导致GC时间放缓，最终OutOfMemoryErrors。
- **垃圾收集持续时间和频率:** 主节点每30秒会去检查其他节点的状态，如果任何节点的垃圾回收时间超过30秒，则会导致主节点任务该节点



脱离集群。

- **内存使用情况:** 如上所述，es非常会利用除了分配给JVM heap的任何RAM。像Kafka一样，es被设计为依赖操作系统的文件系统缓存来快速可靠地提供请求。许多变量决定了Elasticsearch是否成功读取文件系统缓存，如果segment file最近由es写入到磁盘，它已经in the cache。然而如果节点被关闭并重新启动，首次查询某个segment的时候，数据很可能是必须从磁盘中读取，这是确保您的群集保持稳定并且节点不会崩溃的重要原因之一。总的来说，监控节点上的内存使用情况非常重要，并且尽可能多给es分配RAM。

## 6.资源饱和度和错误

Elasticsearch节点使用线程池来管理线程如何消耗内存和CPU。由于线程池设置是根据处理器数量自动配置的，因此通常没有必要对其进行调整。但是，最好注意线程池的队列和拒绝，以了解集群节点性能是不是满足要求；如果是这样，可能想要添加更多节点以处理所有并发请求。

- **线程池队列和拒绝:** 每个节点维护许多类型的线程池; 您要监视的确切位置将取决于您对es的具体用途，一般来说，监控的最重要的是搜索，索引，merge和bulk，它们与请求类型（搜索，索引，合并和批量操作）相对应。线程池队列的大小反应了当前等待的请求数。队列允许节点跟踪并最终服务这些请求，而不是丢弃它们。一旦超过线程池的最大队列大小，线程池的拒绝就会发生。

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:** 大队列是不理想的，因为它们耗尽资源，并且如果节点关闭，还会增加丢失请求的风险。如果看到线程池rejected稳步增加，我们可能希望尝试减慢请求速率（如果可能），增加节点上的处理器数量或增加群集中的节点数量。查询负载峰值与搜索线程池队列大小的峰值相关，因为节点会试图跟上查询请求的速度。
- **批量拒绝和批量排队:** 批量操作是一次发送许多请求的更有效的方式。通常，如果要执行许多操作（创建索引或添加，更新或删除文档），则应尝试以批量操作发送请求，而不是发送许多单独的请求。bulk rejections 通常与在一个批量请求中尝试索引太多文档有关。根据Elasticsearch的文档，批量rejections并不是很需要担心的事。然而，应该尝试实现线性或指数后退策略，以有效地处理批量拒绝。
- **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！
- **字段数据缓存:** 在对字段进行排序或聚合时，会使用fielddata缓存，该过程基本上必须 反转倒排索引以文档顺序创建每个字段的每个字段值的数组。