You want to collect log or transaction data and you want to analyze and mine this data to look for trends, statistics, summarizations, or anomalies. In this case, you can use Logstash (part of the Elasticsearch/Logstash/Kibana stack) to collect, aggregate, and parse your data, and then have Logstash feed this data into Elasticsearch.

<https://speakerdeck.com/javanna/whats-new-in-percolator>

 use Kibana (part of the Elasticsearch/Logstash/Kibana stack) to build custom dashboards that can visualize aspects of your data that are important to you

An index is a collection of documents that have somewhat similar characteristics. For example, you can have an index for customer data, another index for a product catalog, and yet another index for order data

Within an index, you can define one or more types

Type

define a type for user data, another type for blog data, and yet another type for comments data.

Ater the index is created, you may change the number of replicas dynamically anytime but you cannot change the number shards after-the-fact.

./elasticsearch --cluster.name my\_cluster\_name --node.name my\_node\_name

curl 'localhost:9200/\_cat/health?v'

curl 'localhost:9200/\_cat/nodes?v'

curl 'localhost:9200/\_cat/indices?v'

curl -XPUT 'localhost:9200/customer?pretty'  
{  
  "acknowledged" : true  
}  
  
curl 'localhost:9200/\_cat/indices?v'

curl -XPUT 'localhost:9200/customer/external/1?pretty' -d '  
{  
  "name": "John Doe"  
}'

curl -XDELETE 'localhost:9200/customer?pretty'  
{  
  "acknowledged" : true  
}  
curl 'localhost:9200/\_cat/indices?v'

curl -<REST Verb> <Node>:<Port>/<Index>/<Type>/<ID>

<http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/_updating_documents.html>

{

"script" : "ctx.\_source.age += 5",

"lang":"groovy"

}

<http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/_batch_processing.html>

<http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/_the_search_api.html>

<http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/_introducing_the_query_language.html>

<http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/_executing_searches.html>

bool must clause specifies all the queries that must be true for a document to be considered a match.

the bool should clause specifies a list of queries either of which must be true for a document to be considered a match.

<http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/_executing_filters.html>

Under \*nix system, the command will start the process in the foreground. To run it in the background, add the -d switch to it:

$ bin/elasticsearch -d

**mlockall**

The third option on Linux/Unix systems only, is to use [mlockall](http://opengroup.org/onlinepubs/007908799/xsh/mlockall.html" \t "_top) to try to lock the process address space into RAM, preventing any Elasticsearch memory from being swapped out. This can be done, by adding this line to the config/elasticsearch.yml file:

bootstrap.mlockall: true

After starting Elasticsearch, you can see whether this setting was applied successfully by checking the value of mlockall in the output from this request:

curl http://localhost:9200/\_nodes/process?pretty

System configuration

file descriptors

To set this value permanently, update the vm.max\_map\_count setting in /etc/sysctl.conf.

The third option on Linux/Unix systems only, is to use [mlockall](http://opengroup.org/onlinepubs/007908799/xsh/mlockall.html" \t "_top) to try to lock the process address space into RAM, preventing any Elasticsearch memory from being swapped out. This can be done, by adding this line to the config/elasticsearch.yml file:

bootstrap.mlockall: true

need to edit the /etc/fstab file and comment out any lines that contain the word swap.