# Ejercicio 1 - Instalación y configuración de ES

|  |  |
| --- | --- |
| Apellidos, Nombre: | Batista, Tania |
| Puntuación Final: |  |

[> kibana console interface (to run queries)](https://www.elastic.co/guide/en/kibana/current/targz.html)

./bin/kibana

[> terminal interface (to run jvm options)](https://www.elastic.co/guide/en/elasticsearch/reference/current/_installation.html)

cd elasticsearch-6.2.4/bin

./elasticsearch

## Pregunta 1 [Puntuación 2]

### Explica de forma detallada si recomendarías hacer uso de ElasticSearch como un sistema de base de datos tradicional

* No I would not recommend it! ElasticSearch should be used only for search-intensive needs because it is good at searching quickly and dynamically through large amounts of data.
* It is a good NoSQL tool, but it has a lot of limitations. For example, there is no transaction when it is indexing the data. Therefore it causes a lot of data to be lost.

### ¿Qué ventajas o qué inconvenientes puede tener este uso?

Advantages of ElasticSearch:

* Finding documents that contain specific terms and phrases (uppercase/lowercase, ending in s, etc.)
* Scoring and sorting documents according to relevance.
* Aggregating data (sum, average, count) which is useful in e-commerce for grouping results in categories
* Easy clustering structure: You can just dump information into ES. Everything in the search has a JSON interface.
* Querying for matches is extremely because you just need to use ‘match’: {}

Disadvantages

* You cannot return the original documents, because there is no transaction while indexing data. You can’t domake a search and then reassemble it into the original document from the data structure (indexing). So there will be data loss. Additionally, there is no foreign key or relation like in RDMS. With ES you have to copy the same data to make up for this absence, which causes an unnecessary amount of duplicated data.
* It’s kind of anti pattern. Your application must have a lot of ideas mixed together to do phrase matching.

## Pregunta 2 – [Puntuación 2]

### Explica las ventajas o inconvenientes que puede tener configurar los archivos de log y de datos en discos diferentes

[Advantages](https://www.morpheusdata.com/blog/2015-01-30-is-elasticsearch-the-right-solution-for-you):

* Indices can be divided into shards, with each shard able to have any number of replicas. So ES server disks will never be full of log or data. ES will crash directly if the disk gets full! Routing and rebalancing operations are done automatically when new documents are added. Data can safely be indexed into Elasticsearch without having to perform a low-level Lucene commit for every document.

[Disadvantages:](https://www.elastic.co/guide/en/elasticsearch/resiliency/current/index.html#_loss_of_documents_during_network_partition_status_ongoing)

* Repeated network partitions can cause cluster state updates to be lost. And when a node receives a search request, it needs to communicate that request to a copy (either primary or replica) of every shard in the index. So disk servers network latency will cause some slowness.

*Source:* [*https://www.elastic.co/blog/found-dive-into-elasticsearch-storage*](https://www.elastic.co/blog/found-dive-into-elasticsearch-storage)

## Pregunta 3- [Puntuación 0,5]

### ¿Cuál es el mínimo número de réplica para soportar la caída de dos servidores en un cluster de 5 nodos?

*What is the minimum replica number to support the fall of 2 servers in a cluster of 5 nodes?*

* There is no definitive answer, but at minimum, 2 replicas should be used. This way, each node will have a full copy of the data from the index. It will still be safe if 2 nodes are lost at the same time.
* A 5-node cluster looks like this:
  + 5-node cluster = [shard-1], [shard-2], [shard-3], [shard-4], [shard-5]
* A 5-node cluster with 2 replicas, and 2 fallen servers, looks like this:
  + ~~[shard1, shard5, shard4]~~
  + ~~[shard2, shard1, shard5]~~
  + [shard3, shard2, shard1]
  + [shard4, shard3, shard2]
  + [shard5, shard4, shard3]

*Source: http://45.77.47.12/elasticsearch-how-many-shards-replicas-should-an-index-have/*

## Pregunta 4- [Puntuación 0,5]

### Describe los pasos a seguir para configurar un nodo de ES para que tenga una memoria máxima de 4GB.

*If you want to configure an ES node so that it has a maximum memory of 4GB, what are the steps?*

* Configure HEAP\_SIZE to have maximum 2GB.
* Reason: The standard recommendation is to give 50% of the available memory to Elasticsearch heap, while leaving the other 50% free.

*Source:*

*https://www.elastic.co/guide/en/elasticsearch/guide/current/heap-sizing.html#\_give\_less\_than\_half\_your\_memory\_to\_lucene*

## Pregunta 5 – [Puntuación 0.5]

### ¿Cómo configurarías ES para reservar durante el arranque los 4GB de memoria reservada?

*If you want to reserve 4GB of reserved memory during booting, how do you configure ES?*

* We need to allocate 4GB of memory, but we don’t know the total memory capacity of the server.
* So we configure the parameters like this: ES\_JAVA\_OPTS="-Xms4g -Xmx4g".

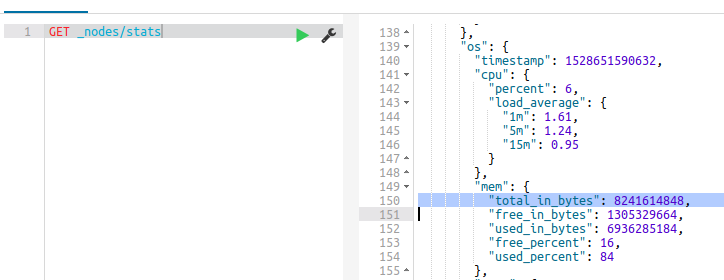
*source: https://www.elastic.co/guide/en/elasticsearch/reference/current/heap-size.html*

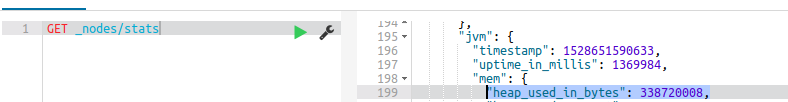
## Pregunta 6 – [Puntuación 0,5]

### Mostrar, mediante el uso del API REST de ElasticSearch, que la memoria reservada está siendo utilizada

*Show, by using the ElasticSearch REST API, that reserved memory is being used.*

* Run the request: GET \_nodes/stats
* Check nodes in: os.mem.total\_in\_bytes and jvm.mem.heap\_used\_in\_bytes

**

**

*Sources:*

*https://www.elastic.co/guide/en/elasticsearch/reference/current/logging.html*

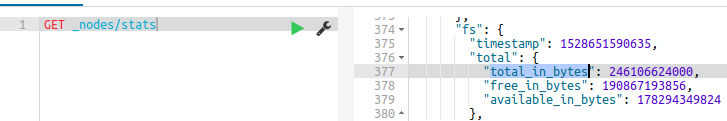
*https://www.elastic.co/guide/en/elasticsearch/reference/current/index-modules-slowlog.html*

## Pregunta 7 – [Puntuación 1]

### Mostrar, mediante el uso del API REST, el tamaño libre en disco para un nodo de ES

*Show, by using the REST API, the free disk size for an ES node*

* Run the request: GET \_nodes/stats
* Check nodes: fs.total.total\_in\_bytes

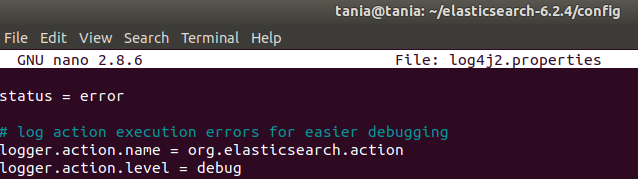


## Pregunta 8 – [Puntuación 1]

### Configura ES para que escriba información de depuración en los ficheros de log

*Set ES to write debugging information in the log files*

* Open file: log4j2.properties
* Look for parameter: logger.action.level
* Set to debugging → logger.action.level = debug



## Pregunta 9 – [Puntuación 1]

### Configura ES para que escriba cualquier consulta ejecutada contra el nodo en los ficheros de log

*Configure ES to write any query executed against the node in the log files*

* You have to set slowlogs threholds to 0:

PUT /\_settings

{

"index.indexing.slowlog.threshold.index.warn": "0ms",

"index.indexing.slowlog.threshold.index.info": "0ms",

"index.indexing.slowlog.threshold.index.debug": "0ms",

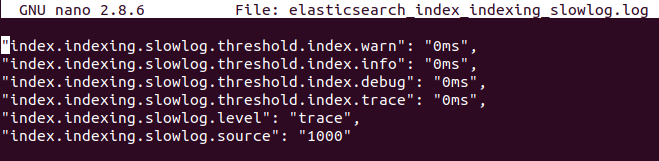
"index.indexing.slowlog.threshold.index.trace": "0ms",

"index.indexing.slowlog.level": "trace",

"index.indexing.slowlog.source": "1000"

}

~/elasticsearch-6.2.4/logs/elasticsearch\_index\_search\_slowlog.log

**

*Source:*

*https://www.elastic.co/guide/en/elasticsearch/reference/current/index-modules-slowlog.html*

## Pregunta 10 – [Puntuación 1]

### ¿Qué hace ElasticSearch si en un cluster de 3 nodos de datos habilitamos un factor de replicación de 3?

*What does ElasticSearch do if we enable a replication factor of 3 in a cluster of 3 data nodes?*

* For 3 replica factors, we would need FOUR data nodes with primary shards. This is because ES is waiting to create the three replica shards of an index.
* Since we only have 3, the cluster health state will turn yellow.
* To check cluster health, we can use GET \_cluster/health request
  + Red = specific shard is not allocated in the cluster
  + Yellow = primary shard is allocated but replicas are not
  + Green = all shards are allocated

*Source:*

*https://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-health.html*