

Mô hình ELK trong Elasticsearch

1. Elasticsearch là gì, ông lớn nào đang dùng nó :

1.1 Elasticsearch là gì:

- Là một search engine.
- Kế thừa từ apache lucene (gần 20 năm kinh nghiệm trong search text).
- Người anh em của nó là "apache solr", tuy nhiên hướng đi khác nhau. (Solr chỉ tập trung vào full text search trong khi elastic giải quyết khá nhiều bài toán khác nữa...)
- Trả về kết quả về kết quả thông qua Restful. Do đó không phụ thuộc vào ngôn ngữ lập trình.

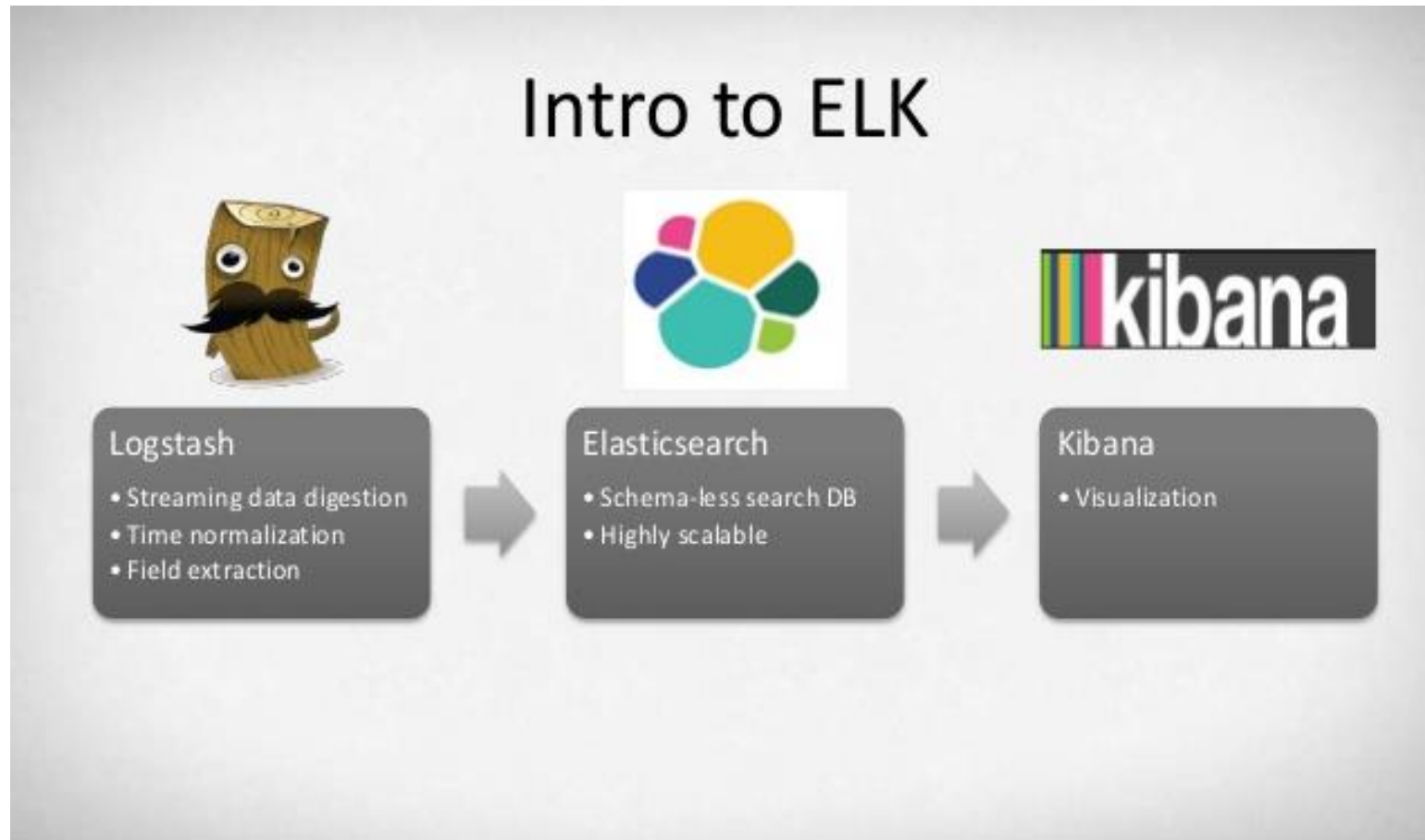
1.2 Những ông lớn nào đang dùng nó:

- Wikimedia, Adobe Systems, Facebook , Github, Soundcloud, Quora...

Mô hình ELK trong Elasticsearch

2. Sơ đồ và nguyên tắc hoạt động :

2.1 Sơ đồ:



Mô hình ELK trong Elasticsearch

2. Sơ đồ và nguyên tắc hoạt động :

2.2 Nguyên tắc hoạt động :

- Logstash là 1 "data center" làm nhiệm vụ đẩy log, tiền xử lý, trước khi đẩy vào elastic search.
 - Elastic search là nhân, 1 search engine, cho phép search data từ logstash đẩy vào theo cú pháp query được quy định.
 - Kibana là giao diện cho phép quản lý elastic search (ram, memory, disk,...), search trực tiếp...
- > Map vào 1 chương trình, nếu như elastic search là "logic, controller" thì logstash là "database" còn kibana là "giao diện"

Mô hình ELK trong Elasticsearch

2. Sơ đồ và nguyên tắc hoạt động :

2.3 Vì sao elastic search query cực nhanh :

- Theo một thống kê không chính thức (do tác giả từng thống kê, không có yếu tố chém gió !) thì tốc độ query đạt được là tìm thấy gần 10 triệu bản ghi trong vòng hơn 1s.
- > Bởi vì cơ chế đánh index của elastic search có đôi chút khác biệt, gọi là "inverted index".

Ví dụ: Có 3 bản ghi như sau:

Doc1 = Chỉ còn 1 chiếc lá cuối thu mỏng manh

Doc2 = Chỉ còn 1 mình em xót xa chờ anh

Doc3 = Chỉ còn đêm nay mai lá kia rơi

Elastic sẽ tách thành từng từ và map những từ đó xem nó đang có mặt ở bản ghi nào.

Đây chính là cơ chế **inverted index**. Cụ thể ta sẽ có các **inverted index** như sau:

{Chỉ} =(Doc1, Doc2, Doc3), {Còn} =(Doc1, Doc2, Doc3), {1} ={Doc1, Doc2}, {chiếc} =(Doc1), {lá} =(Doc1, Doc3)

...Giả sử search : "Chỉ còn lá" thì sẽ là phép AND: (Doc1, Doc2, Doc3) AND (Doc1, Doc2, Doc3)

AND (Doc1, Doc3) =(Doc1, Doc3). Việc search chỉ là việc thực hiện cái phép toán AND, OR . => nhanh !

Mô hình ELK trong Elasticsearch

3. Cài đặt trên ubuntu linux:

-> Elasticsearch viết bằng java nên trước khi cài đặt cần cài JDK (nếu chưa có).

3.1 Cài đặt screen:

- Vì sao phải cài: Tưởng tượng là bạn có 1 chương trình đang chạy, nhưng bạn muốn thoát server để về nhà sớm. (ý nói đến màn hình server nơi mà chương trình này đang chạy).

Như vậy, thoát server đồng nghĩa với việc tắt luôn chương trình ? Để giải quyết vấn đề này, trên linux có nhiều giải pháp, cách đơn giản nhất là dùng lệnh "nohup" có sẵn. Một cách hay hơn là dùng screen, screen giúp quản lý các "screen session" tốt hơn.

-> Để cài screen, dùng lệnh sau:

sudo apt-get install screen

(với nhân là redhat, fedora chẳng hạn, câu lệnh sẽ là: ***sudo yum install screen***)

Mô hình ELK trong Elasticsearch

3. Cài đặt trên ubuntu linux:

3.1 Cài đặt screen:

- Sau khi cài đặt xong, tạo sẵn 3 screen để quản lý elasticseach, logstash và kibana. Để tạo từng screen, ta dùng câu lệnh:

screen -S [name]

-> Tạo screen cho elasticseach: `screen -S elastic_seach`

-> Tạo screen cho logstash: `screen -S logstash`

-> Tạo screen cho kibana: `screen -S kibana`

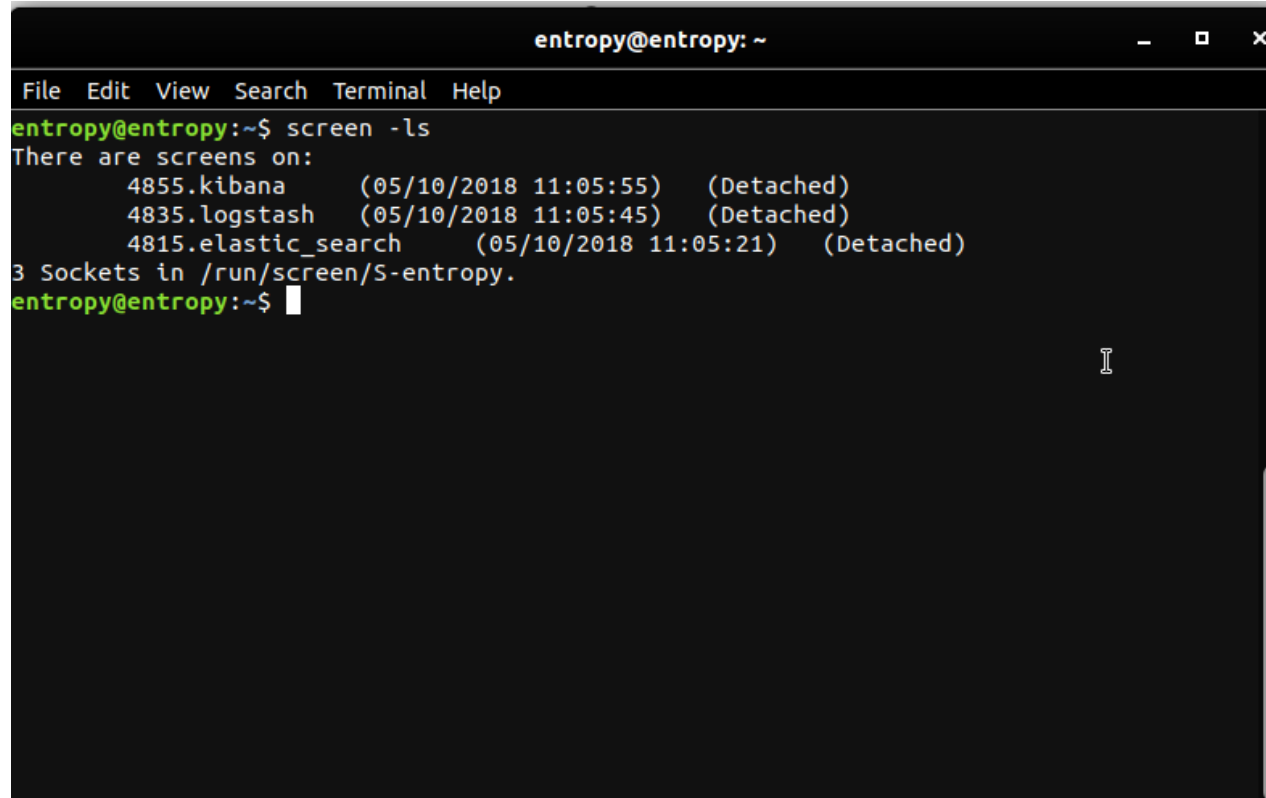
(Lưu ý: Mỗi khi tạo xong 1 screen, sẽ tự động "nhảy vào" screen đó, để thoát screen vừa tạo ra tiếp tục tạo screen tiếp theo dùng tổ hợp phím: `ctr A + ctr D`)

Mô hình ELK trong Elasticsearch

3. Cài đặt trên ubuntu linux:

3.1 Cài đặt screen:

- Để xem thành quả vừa tạo, gõ lệnh "screen -ls". Nếu hiện ra list screen vừa tạo là thành công:

A screenshot of a terminal window titled 'entropy@entropy: ~'. The terminal shows the command 'screen -ls' being executed. The output lists three detached screens: '4855.kibana', '4835.logstash', and '4815.elastic_search', all created on 05/10/2018 at 11:05:55, 11:05:45, and 11:05:21 respectively. It also shows '3 Sockets in /run/screen/S-entropy.' and the prompt 'entropy@entropy:~\$' with a cursor.

```
entropy@entropy: ~  
File Edit View Search Terminal Help  
entropy@entropy:~$ screen -ls  
There are screens on:  
    4855.kibana      (05/10/2018 11:05:55)  (Detached)  
    4835.logstash   (05/10/2018 11:05:45)  (Detached)  
    4815.elastic_search (05/10/2018 11:05:21)  (Detached)  
3 Sockets in /run/screen/S-entropy.  
entropy@entropy:~$
```

Mô hình ELK trong Elasticsearch

3. Cài đặt trên ubuntu linux:


3.2 Cài đặt elasticsearch:

- Để cài elastic search trên linux, có thể cài file (debian) hoặc giải nén file tar đóng gói, docker. Ở đây chúng ta tải file debian và giải nén nó trong thư mục "etc".

-> Các bước cài đặt sẽ là:

3.2.1. Đăng ký key cho elasticsearch:

`wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -`



```
entropy@entropy: /etc
entropy@entropy:/etc$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
[sudo] password for entropy:
OK
entropy@entropy:/etc$
```


Mô hình ELK trong Elasticsearch

3. Cài đặt trên ubuntu linux:

3.2 Cài đặt elasticsearch:

3.2.2. Install package apt-transport-https (dùng cho việc download ES) cho linux trước khi cài đặt:

```
sudo apt-get install apt-transport-https
```

3. 2.3. Save package

```
echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-6.x.list
```

3.2.4 Install elasticsearch:

```
sudo apt-get update && sudo apt-get install elasticsearch
```

Mô hình ELK trong Elasticsearch

3. Cài đặt trên ubuntu linux:

3.2 Cài đặt elasticsearch:

3.2.5. Start/stop elastic seach với systemctl:

-> Ta đã tạo screen "elastic_search" cho việc run elastic search. Gõ lệnh ***screen -ls*** để view id, sau đó gõ lệnh ***screen -r [id của screen "elastic_search"]*** để vào screen elastic search, sau đó gõ lệnh dưới để setup service run:

```
sudo /bin/systemctl daemon-reload
```

```
sudo /bin/systemctl enable elasticsearch.service
```

```
sudo systemctl start elasticsearch.service
```

```
sudo systemctl stop elasticsearch.service
```

Mô hình ELK trong Elasticsearch

3. Cài đặt trên ubuntu linux:

3.2 Cài đặt elasticsearch:

3.2.6 Kiểm tra thành quả: gõ "localhost:9200" (9200 là port mặc định), thấy câu :**"You Know, for Search"** nổi tiếng là thành công.



```
{
  "name" : "QCQ_aj0",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "XN105I3pS30lgfuDFaeglg",
  "version" : {
    "number" : "6.4.0",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "595516e",
    "build_date" : "2018-08-17T23:18:47.308994Z",
    "build_snapshot" : false,
    "lucene_version" : "7.4.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

Mô hình ELK trong Elasticsearch

3. Cài đặt trên ubuntu linux:

3.3 Cài đặt kibana:

3.3.1 Install kibana (các bước đăng ký key + install package apt-transport-https + save package đã thực hiện trước đó khi cài elasticsearch nên bỏ qua):

```
sudo apt-get update && sudo apt-get install kibana
```

3.3.2 Start/stop kibana (vào screen "kibana" tương tự như trên) rồi gõ lệnh:

```
sudo /bin/systemctl daemon-reload
```

```
sudo /bin/systemctl enable kibana.service
```

```
sudo systemctl start kibana.service
```

```
sudo systemctl stop kibana.service
```

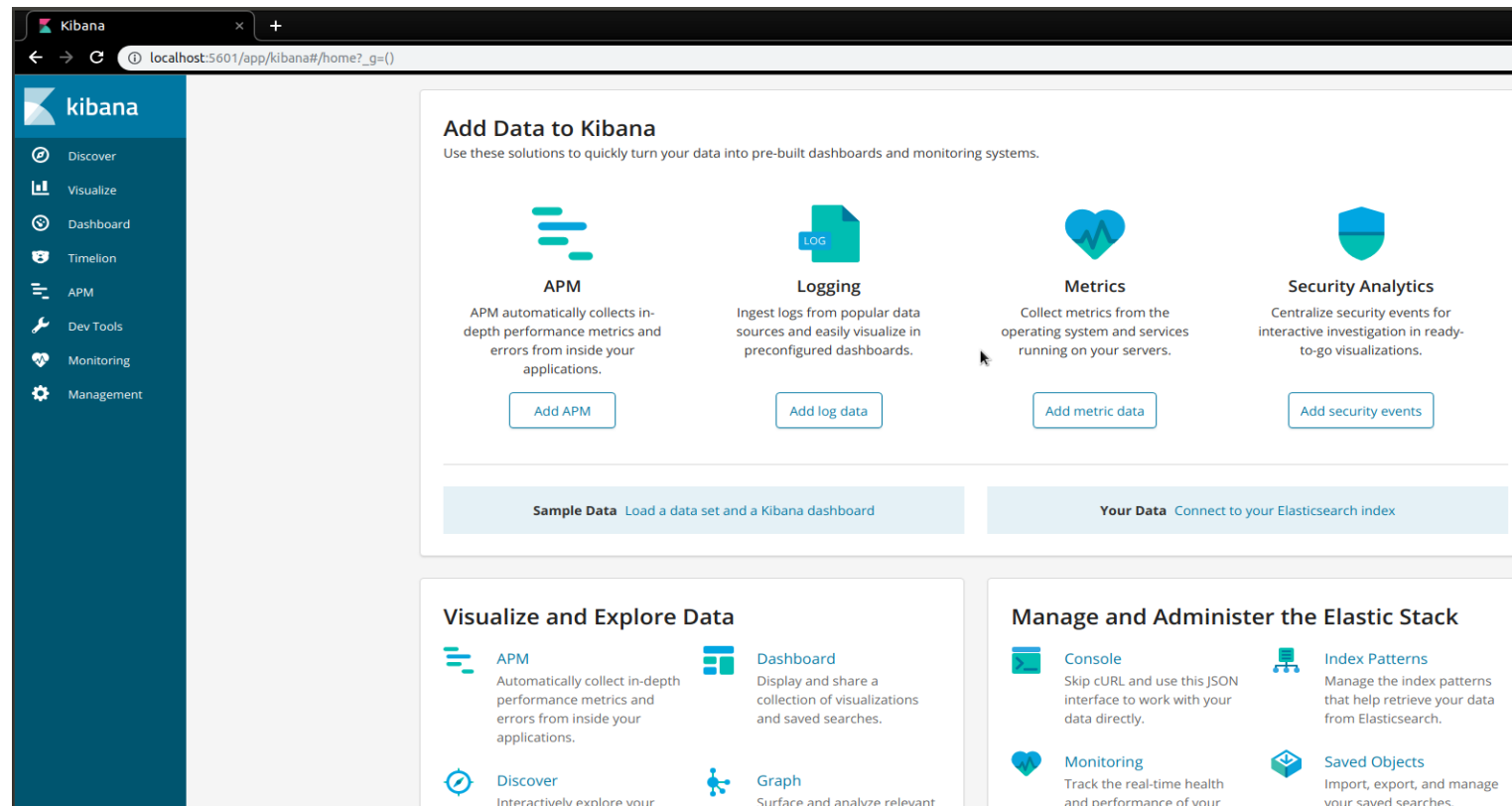
Mô hình ELK trong Elasticsearch

3. Cài đặt trên ubuntu linux:

3.3 Cài đặt kibana:

3.3.3 Kiểm tra thành quả:

Gõ "**localhost:5601**", nếu hiện ra giao diện kibana là cài đặt thành công (cổng 5601 là cổng mặc định của kibana)



Mô hình ELK trong Elasticsearch

3. Cài đặt trên ubuntu linux:

3.4 Cài đặt logstash:

3.4.1 Install logstash (các bước đăng ký key + install package apt-transport-https + save package đã thực hiện trước đó khi cài elasticsearch nên bỏ qua):

```
sudo apt-get update && sudo apt-get install logstash
```

3.4.2 Start/stop logstash (vào screen "logstash" tương tự như trên) rồi gõ lệnh:

```
sudo /bin/systemctl daemon-reload
```

```
sudo /bin/systemctl enable logstash.service
```

```
sudo systemctl start logstash.service
```

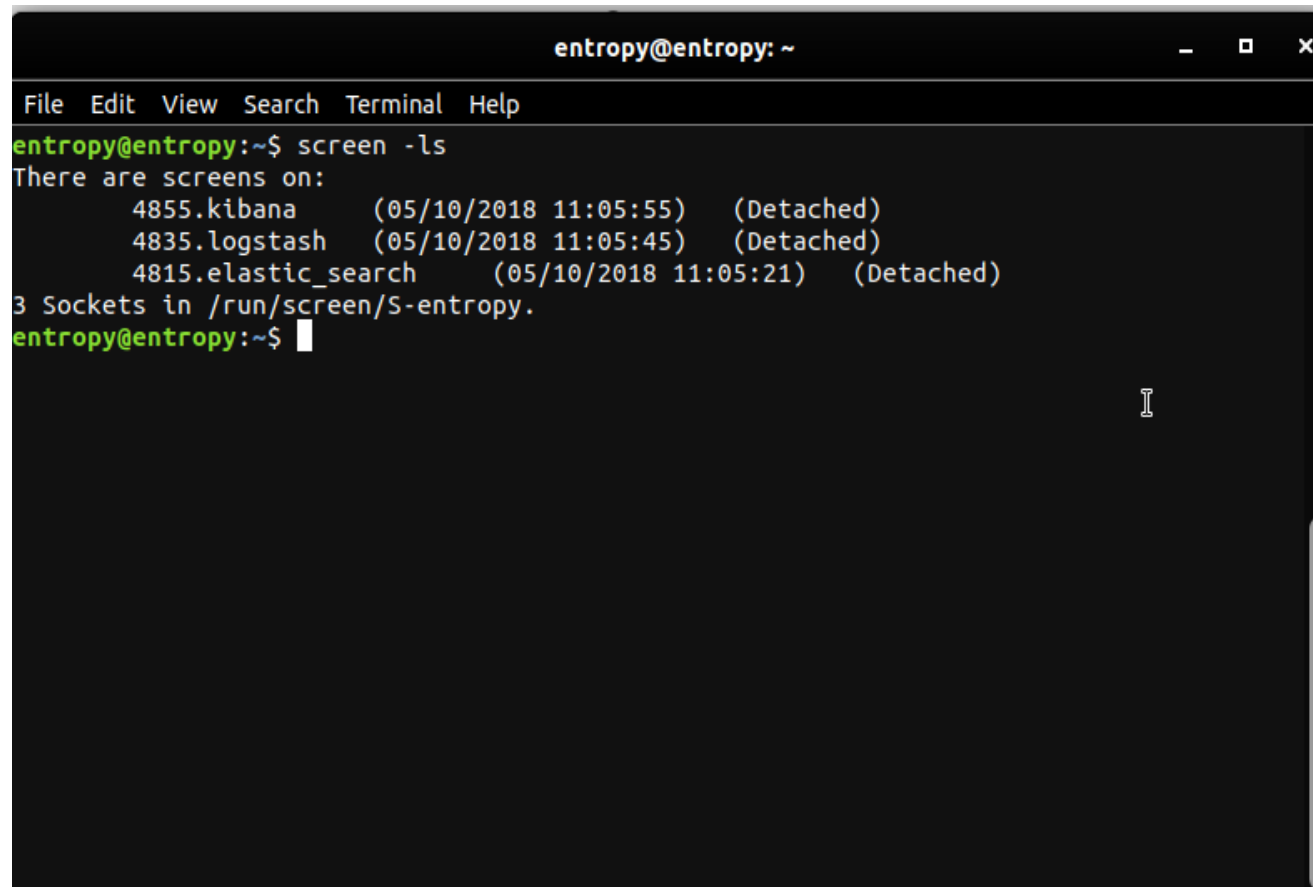
```
sudo systemctl stop logstas.service
```

Mô hình ELK trong Elasticsearch

4. Demo

4.1 Đẩy file vào logstash :

- Gõ ***screen -ls*** để hiển thị 3 screen vừa tạo:



```
entropy@entropy: ~  
File Edit View Search Terminal Help  
entropy@entropy:~$ screen -ls  
There are screens on:  
    4855.kibana      (05/10/2018 11:05:55)  (Detached)  
    4835.logstash   (05/10/2018 11:05:45)  (Detached)  
    4815.elastic_search (05/10/2018 11:05:21) (Detached)  
3 Sockets in /run/screen/S-entropy.  
entropy@entropy:~$
```

Mô hình ELK trong Elasticsearch

4. Demo

4.1 Đẩy file vào logstash :

- Vào screen logstash (id là "4835") để config. Câu lệnh ***screen -r 4835***
- Sau khi vào, tạo file config trong thư mục ***etc/logstash/conf.d/*** để đẩy file data vào logstash:



```
entropy@entropy: /etc/logstash/conf.d
File Edit View Search Terminal Help
input{
  file {
    path => "/home/entropy/Downloads/sql.sql"
    start_position => "beginning"
  }
}
output{

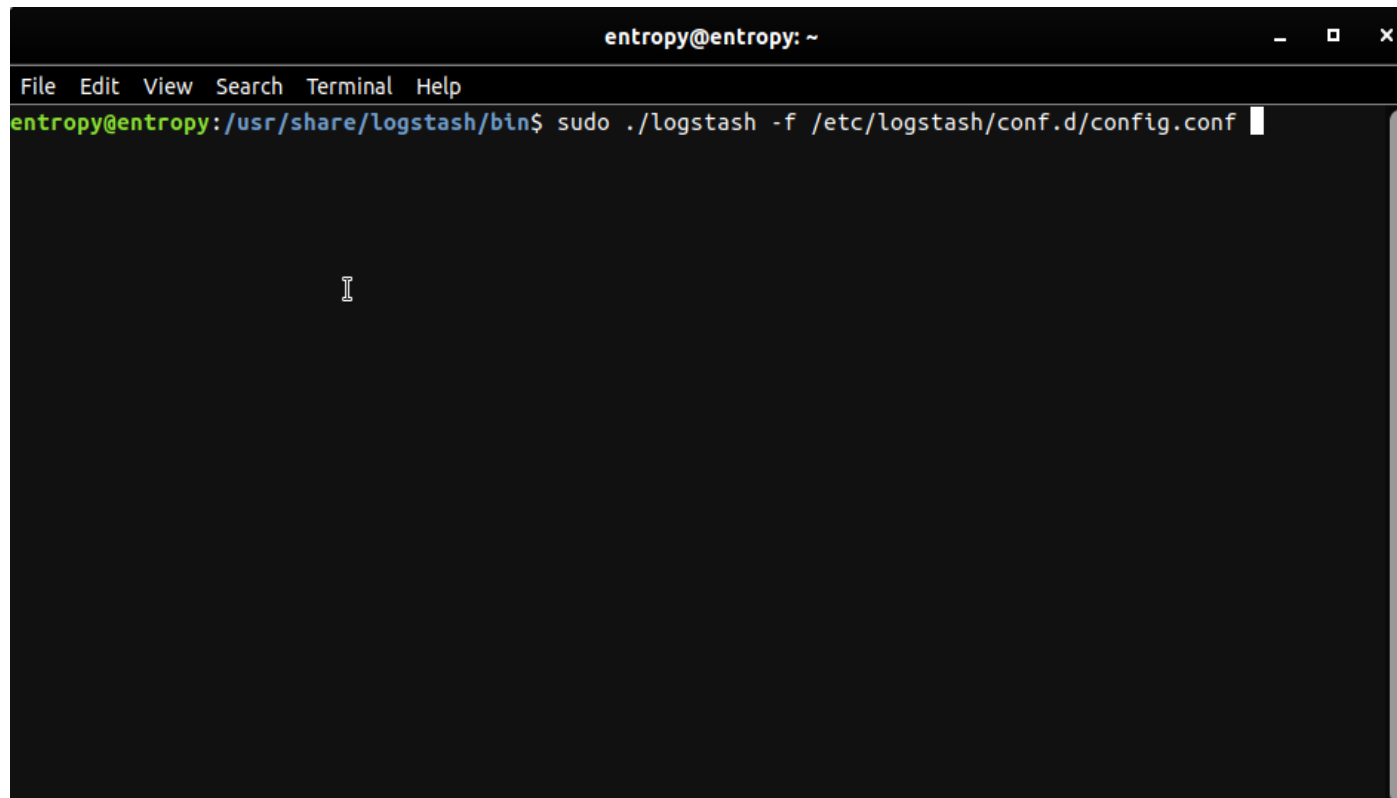
  elasticsearch {
    hosts => ["localhost:9200"]
    index => "test_elastic"
    document_type => "test"
  }
}
~
~
~
~
~
~
~
~
~
~
"config.conf" 14L, 210C 14,1 All
```


Mô hình ELK trong Elasticsearch

4. Demo

4.1 Đẩy file vào logstash :

- Vào thư mục bin của logstash: ***cd /usr/share/logstash/bin*** và chạy lệnh sau để logstash đẩy data:
sudo ./logstash -f /etc/logstash/conf.d/config.conf



```
entropy@entropy: ~  
File Edit View Search Terminal Help  
entropy@entropy:/usr/share/logstash/bin$ sudo ./logstash -f /etc/logstash/conf.d/config.conf
```

Mô hình ELK trong Elasticsearch

4. Demo

4.1 Đẩy file vào logstash :

- Logstash chạy thành công với port mặc định 9600:

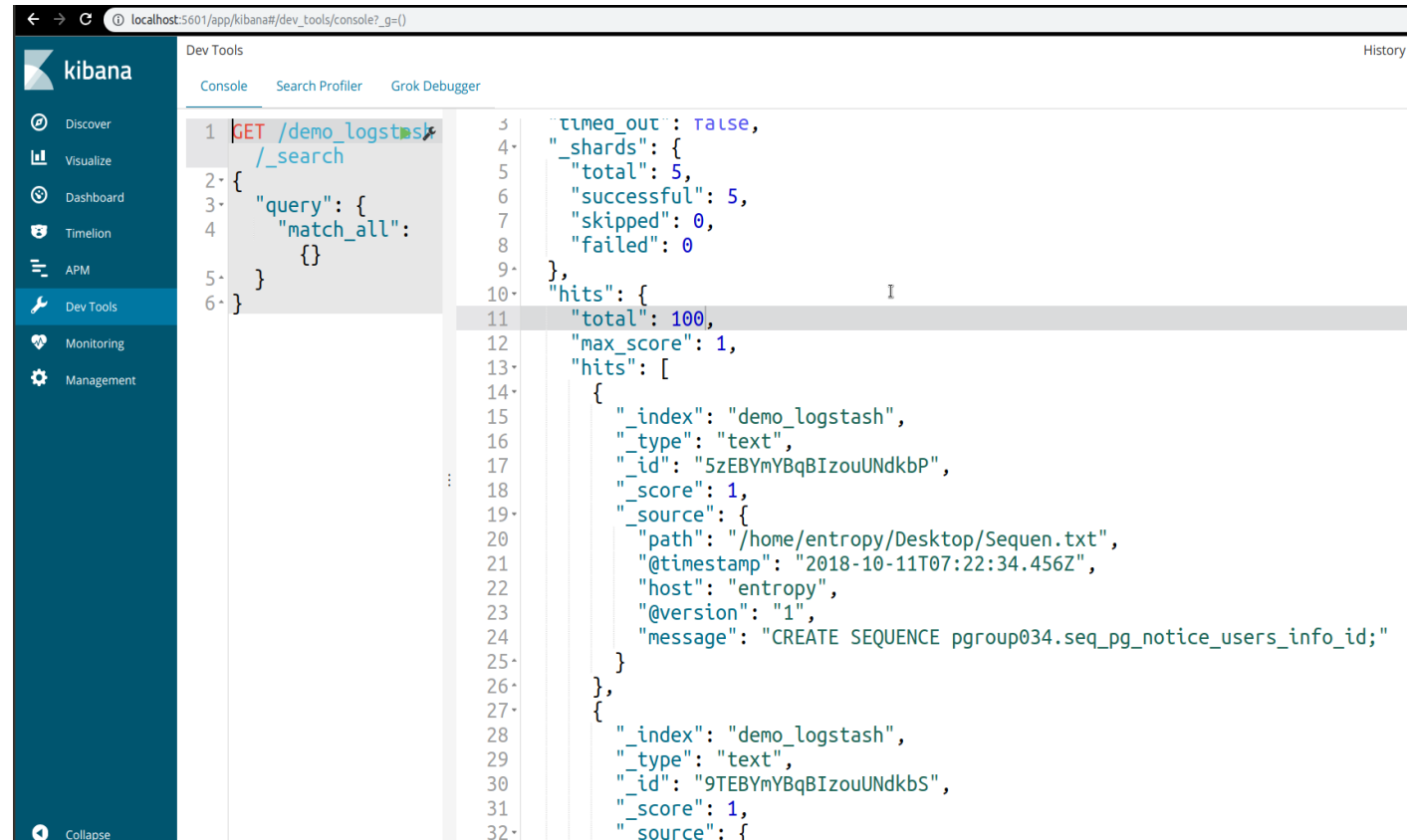
```
entropy@entropy: ~  
File Edit View Search Terminal Help  
ath=>"/"} [WARN ] 2018-10-05 13:52:16.681 [[main]-pipeline-manager] elasticsearch - Restored connection to ES  
instance {:url=>"http://localhost:9200/"} [INFO ] 2018-10-05 13:52:17.307 [[main]-pipeline-manager] elasticsearch - ES Output version determined {:es_version=>6}  
[WARN ] 2018-10-05 13:52:17.313 [[main]-pipeline-manager] elasticsearch - Detected a 6.x and above cluster: the `type` event field won't be used to determine the document _type {:es_version=>6} [INFO ] 2018-10-05 13:52:17.376 [[main]-pipeline-manager] elasticsearch - New Elasticsearch output {:class=>"LogStash::Outputs::ElasticSearch", :hosts=>["//localhost:9200"]} [INFO ] 2018-10-05 13:52:17.405 [Ruby-0-Thread-5: :1] elasticsearch - Using mapping template from {:path=>nil} [INFO ] 2018-10-05 13:52:17.443 [Ruby-0-Thread-5: :1] elasticsearch - Attempting to install template {:manage_template=>{"template"=>"logstash-*", "version"=>60001, "settings"=>{"index.refresh_interval"=>"5s"}, "mappings"=>{"_default_"=>{"dynamic_templates"=>[{"message_field"=>{"path_match"=>"message", "match_mapping_type"=>"string", "mapping"=>{"type"=>"text", "norms"=>false}}], {"string_fields"=>{"match"=>"*", "match_mapping_type"=>"string", "mapping"=>{"type"=>"text", "norms"=>false, "fields"=>{"keyword"=>{"type"=>"keyword", "ignore_above"=>256}}}}]}, "properties"=>{"@timestamp"=>{"type"=>"date"}, "@version"=>{"type"=>"keyword"}, "geoip"=>{"dynamic"=>true, "properties"=>{"ip"=>{"type"=>"ip"}, "location"=>{"type"=>"geo_point"}, "latitude"=>{"type"=>"half_float"}, "longitude"=>{"type"=>"half_float"}}}}}} [INFO ] 2018-10-05 13:52:17.852 [[main]>worker3] file - No sincedb_path set, generating one based on the "path" setting {:sincedb_path=>"/usr/share/logstash/data/plugins/inputs/file/.sincedb_906009606377931aa6988a2a7ec2542d", :path=>"/home/entropy/Downloads/sql.sql"} [INFO ] 2018-10-05 13:52:17.909 [Converge PipelineAction::Create<main>] pipeline - Pipeline started successfully {:pipeline_id=>"main", :thread=>"#<Thread:0x7248105f run>"} [INFO ] 2018-10-05 13:52:18.000 [Ruby-0-Thread-1: /usr/share/logstash/lib/bootstrap/environment.rb:6] agent - Pipelines running {:count=>1, :running_pipelines=>[:main], :non_running_pipelines=>[]} [INFO ] 2018-10-05 13:52:18.047 [[main]<file] observingtail - START, creating Discoverer, Watch with file and sincedb collections [INFO ] 2018-10-05 13:52:18.535 [Api Webserver] agent - Successfully started Logstash API endpoint {:port=>9600}
```

Mô hình ELK trong Elasticsearch

4. Demo

4.2 Kiểm tra thành quả:

- Vào kibana tìm đến tab "dev tools" search với source là `/[tên index]/_search`, ra kết quả là đã đẩy data vào thành công:



The screenshot shows the Kibana Dev Tools console with the following content:

```
localhost:5601/app/kibana#/dev_tools/console?_g=()
Dev Tools
Console Search Profiler Grok Debugger
1 GET /demo_logstash/_search
2 {
3   "query": {
4     "match_all": {}
5   }
6 }
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
{
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 100,
    "max_score": 1,
    "hits": [
      {
        "_index": "demo_logstash",
        "_type": "text",
        "_id": "5zEBYmYBqBIzouUNdkbP",
        "_score": 1,
        "_source": {
          "path": "/home/entropy/Desktop/Sequen.txt",
          "@timestamp": "2018-10-11T07:22:34.456Z",
          "host": "entropy",
          "@version": "1",
          "message": "CREATE SEQUENCE pgroup034.seq_pg_notice_users_info_id;"
        }
      },
      {
        "_index": "demo_logstash",
        "_type": "text",
        "_id": "9TEBYmYBqBIzouUNdkbS",
        "_score": 1,
        "_source": {

```

Mô hình ELK trong Elasticsearch

5[Nâng cao]. Đồng bộ với 1 database khác:

5.1 Lý do:

- Chưa bao giờ elastic search được coi là 1 database chính thống.
- Do không hỗ trợ cơ chế transaction -> dễ mất mát dữ liệu nếu thao tác với các lệnh insert, update, delete.
- Mặc dù query nhanh nhưng các thao tác CRUD chậm hơn nhiều so với database truyền thống.
- > Nên tận dụng tốc độ query của elastic search kèm khả năng lưu trữ của DB đối với bài toán có dữ liệu lớn.

5.2 Elasticsearch phù hợp với những DB nào:

- Có thể tương thích với hầu hết các loại DB tuy nhiên để phát huy thế mạnh vốn có là query nhanh với các bản ghi không ràng buộc thì các loại DB NoSQL là phù hợp nhất.
- Elastic search sinh ra để giải quyết 2 bài toán quan trọng : analytic và statistic, do đó những dữ liệu không ràng buộc là rất thích hợp.

Mô hình ELK trong Elasticsearch

5[Nâng cao]. Đồng bộ với 1 database khác:

5.3 Đồng bộ với MongoDB:

- MongoDB là db được dùng nhiều trong công ty chúng ta, vì thế chúng ta sẽ thử đẩy dữ liệu vào logstash xem sao !
- Đầu tiên ta cần 1 plugin đảm nhận việc này, nó là ***logstash-output/input-mongodb*** (output là đẩy data từ elastic ra mongodb, còn input là đẩy data từ mongo vào elastic. Trong trường hợp này chúng ta dùng input. Tiện thì install cả 2 cũng được còn không ta chỉ cần input là được).
- Sau khi input plugin xong, chúng ta sẽ dùng nó để đẩy data từ mongo vào elastic. Mỗi khi có sự thay đổi về data plugin sẽ tự động đẩy.

Mô hình ELK trong Elasticsearch

5[Nâng cao]. Đồng bộ với 1 database khác:

5.3 Đồng bộ với MongoDB:

- Câu lệnh install như 2 hình dưới đây:

```
entropy@entropy: /usr/share/logstash/bin
File Edit View Search Terminal Help
entropy@entropy:~$ cd /usr/share/logstash/bin/
entropy@entropy:/usr/share/logstash/bin$ sudo ./logstash-plugin install logstash
-output-mongodb
[sudo] password for entropy:
Validating logstash-output-mongodb
Installing logstash-output-mongodb
Installation successful
entropy@entropy:/usr/share/logstash/bin$
```

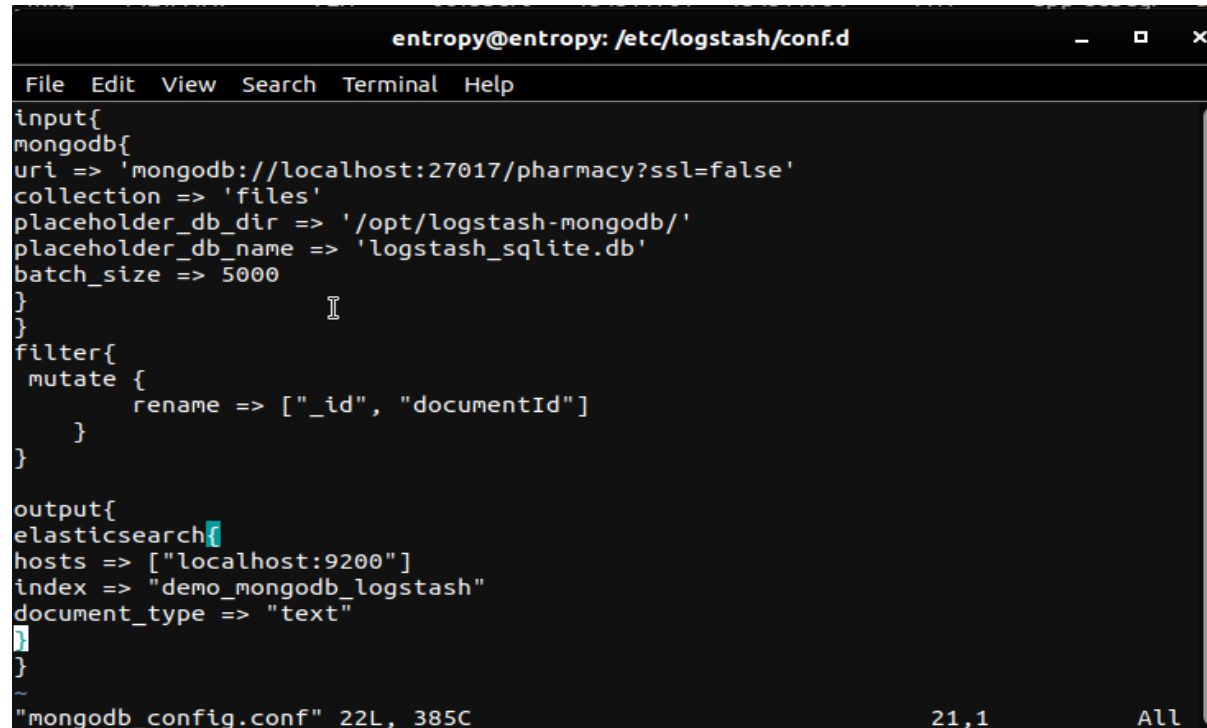
```
entropy@entropy: /usr/share/logstash/bin
File Edit View Search Terminal Help
entropy@entropy:/usr/share/logstash/bin$ sudo ./logstash-plugin install logstash
-input-mongodb
[sudo] password for entropy:
Validating logstash-input-mongodb
Installing logstash-input-mongodb
Installation successful
entropy@entropy:/usr/share/logstash/bin$
```

Mô hình ELK trong Elasticsearch

5[Nâng cao]. Đồng bộ với 1 database khác:

5.3 Đồng bộ với MongoDB:

- Đẩy data từ mongo vào elastic, giả sử trên localhost ta có db với tên là "pharmacy" với port 27017.
=> Ta tạo 1 file config đơn giản như sau để đẩy data từ collection có tên là "files"
(nếu muốn đẩy all db ta bỏ thuộc tính collection).



```
entropy@entropy: /etc/logstash/conf.d
File Edit View Search Terminal Help
input{
  mongodb{
    uri => 'mongodb://localhost:27017/pharmacy?ssl=false'
    collection => 'files'
    placeholder_db_dir => '/opt/logstash-mongodb/'
    placeholder_db_name => 'logstash_sqlite.db'
    batch_size => 5000
  }
}
filter{
  mutate {
    rename => ["_id", "documentId"]
  }
}
output{
  elasticsearch{
    hosts => ["localhost:9200"]
    index => "demo_mongodb_logstash"
    document_type => "text"
  }
}
```

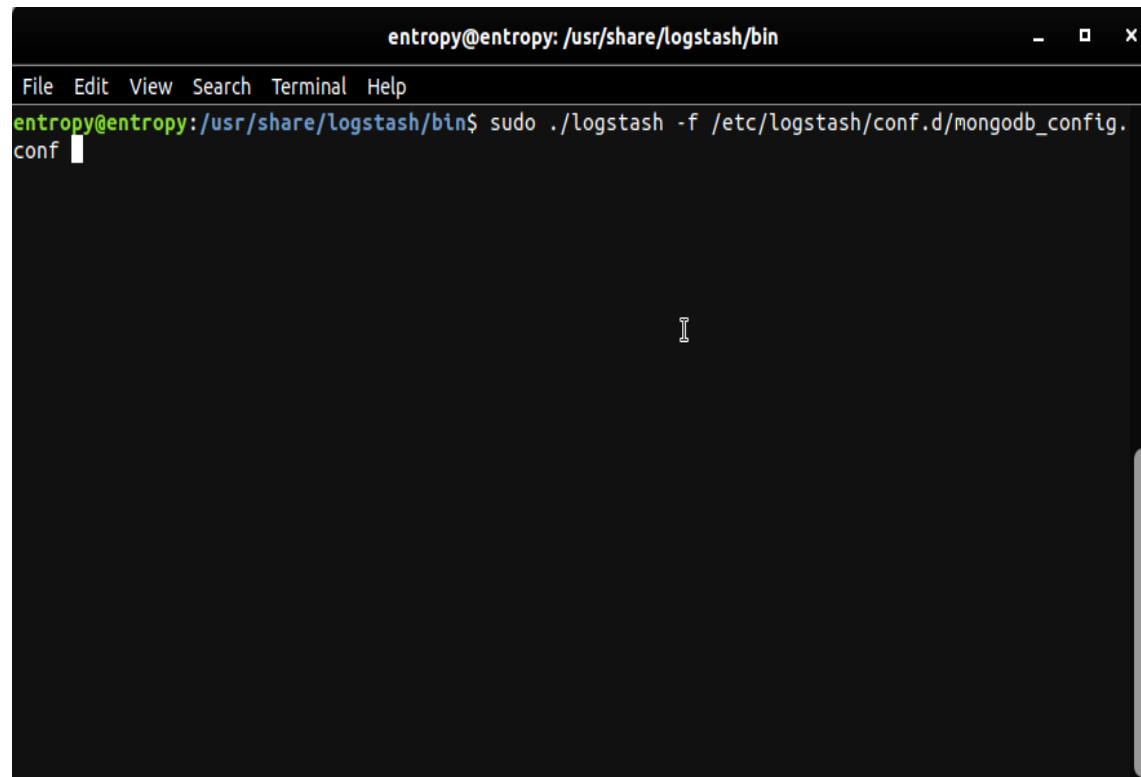
"mongodb_config.conf" 22L, 385C 21,1 All

Mô hình ELK trong Elasticsearch

5[Nâng cao]. Đồng bộ với 1 database khác:

5.3 Đồng bộ với MongoDB:

- Vào thư mục bin của logstash để chạy lệnh đẩy data của mongo. Run xong vào kibana/dev tool để tận hưởng thành quả !



```
entropy@entropy: /usr/share/logstash/bin
File Edit View Search Terminal Help
entropy@entropy:/usr/share/logstash/bin$ sudo ./logstash -f /etc/logstash/conf.d/mongodb_config.conf
```


Mô hình ELK trong Elasticsearch

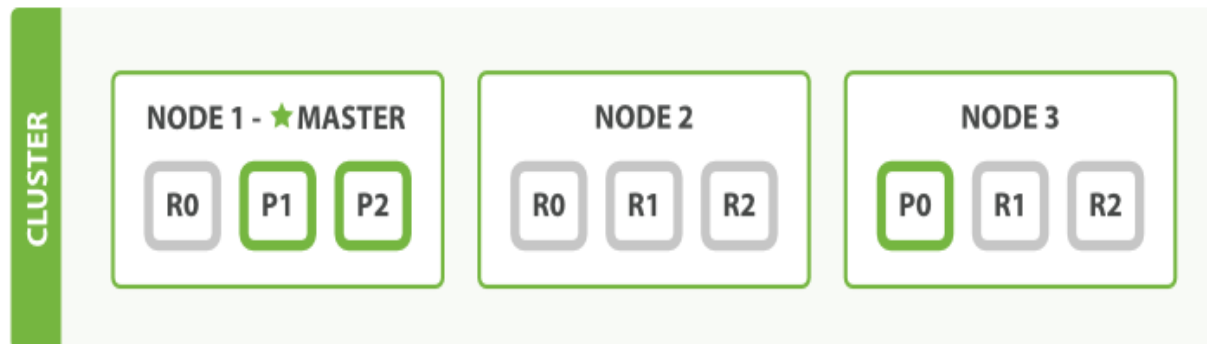
6[Nâng cao]. Cluster (tạo cụm) trong elasticsearch:

6.1 Vì sao phải tạo cụm:

- Theo kinh nghiệm thực tế của tác giả, với khoảng 50 triệu bản ghi lộn lạo, 1 elastic có thể xử lý ngon lành !
- Thế nhưng khi số lượng bản ghi tính bằng trăm triệu, thậm chí là tỉ thì sao?(Dữ liệu của các trang báo mạng Vccorp như [kenh14.vn](#), [dantri.vn](#), [genk.vn](#), [gamek.vn](#)...gộp lại đang là vài chục tỉ record.)
- Lúc này chúng ta cần cài đặt phân cụm theo mô hình master node.

6.2 Master node là gì:

- Hiểu đơn giản, master node (các server elastic) chính là 1 cụm các node hoạt động với nhau cùng mục đích.
- Hễ node này tèo thì node kia sẽ backup, đảm bảo sự toàn vẹn dữ liệu cũng như tải.



Mô hình ELK trong Elasticsearch

6[Nâng cao]. Cluster (tạo cụm) trong elasticsearch:

6.2 Master node là gì:

6.2.1: Các khái niệm:

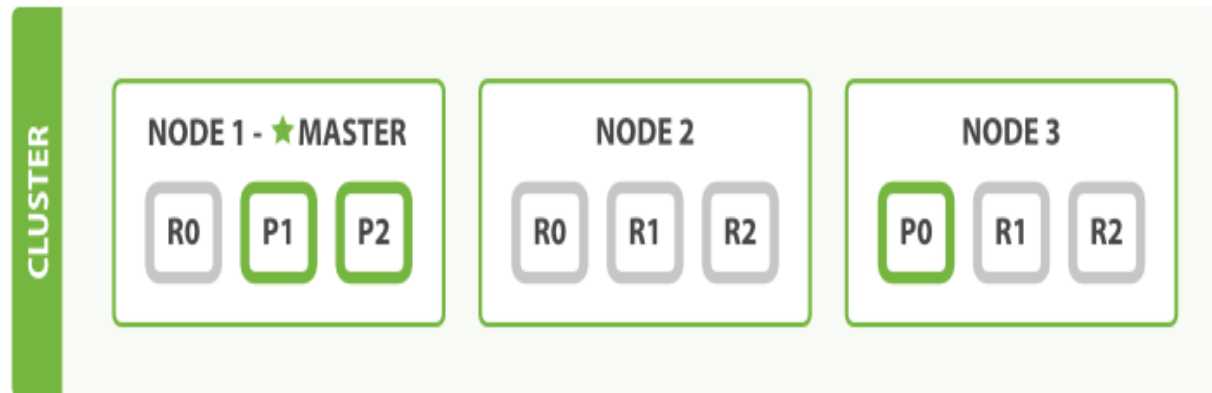
- Node: Mỗi elastic là 1 node.
- Cluster: Gom các elastic lại với nhau thành 1 cụm , gọi là cluster.
- Shard: Đối tượng của lucene (core của elastic và solr) là mức thấp nhất, để lưu trữ, chúng ta ít khi làm việc với shard. Shard gồm "Primary Shard" và "Replica Shard".
- Primary Shard: Là shard có nhiệm vụ lưu dữ liệu, dữ liệu được lưu tại 1 primary shard, được đánh index ở đây trước khi chuyển đến replica shard.
- Replica Shard: Chính là backup của Primary Shard, khi Primary Shard vì lý do nào đó tèo thì Replica Shard sẽ "ra tay back up", nhằm đảm bảo toàn vẹn dữ liệu.

Mô hình ELK trong Elasticsearch

6[Nâng cao]. Cluster (tạo cụm) trong elasticsearch:

6.2 Master node là gì:

6.2.2: Nguyên lý hoạt động:



Mô hình ELK trong Elasticsearch

6[Nâng cao]. Cluster (tạo cụm) trong elasticsearch:

6.2 Master node là gì:

6.2.2: Nguyên lý hoạt động:

- Nhìn vào hình trên chúng ta có thể thấy, dữ liệu được lưu trữ ở cluster với 3 nodes trong đó node 1 là master.
- Có 3 primary shards, 2 trong số đó được đặt ở node 1, còn lại ở node 3.
- Mỗi primary shard có 2 replica shard (ví dụ primary shard P0 ở Node3 thì có replica shard R0 ở node 1 và một shard nữa ở Node 2).
- Việc sắp đặt vị trí primary shard là ngẫu nhiên, còn các replica shard luôn được đảm bảo là nó không nằm cùng node với primary shard (Tại sao? Tưởng tượng node ngưng hoạt động mà toàn bộ shard ở node đó thì sẽ mất hết sạch dữ liệu).
- Thêm nữa là không bắt buộc primary shard đều nằm ở node master, vì việc phân tán các primary shard giúp phân tán công đoạn ghi dữ liệu, giúp giảm tải cho một node.
- Việc lựa chọn node cho các thao tác đọc được thực hiện bởi thuật toán Round-robin.

Mô hình ELK trong Elasticsearch

7. Áp dụng vào thực tế (hệ thống admage):

7.1 Mục đích:

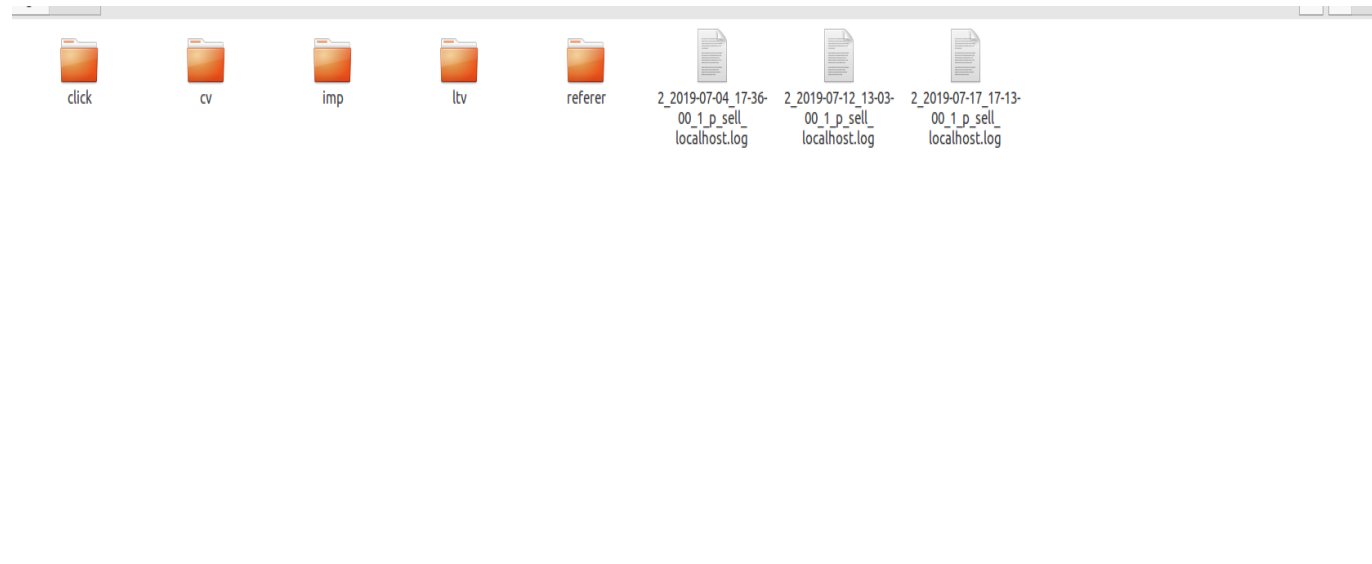
- Quản lý log tốt hơn (với kibana giao diện). Khi số lượng user tăng kèm theo đó là số lượng log cũng tăng phát sinh bài toán bigdata, việc theo dõi nó một cách trực quan là vô cùng quan trọng.
- Giảm thời gian query tới mức "chóng mặt". => Đây là lý do chính nên dùng hệ thống này. Như đã đề cập, tốc độ query đạt 10 triệu bản ghi/ hơn 1s (khoảng 1.2 -> 1.3s). Giải quyết vấn đề query thống kê cho hệ thống "batch": Bỏ logic đọc file, thay vào đó query tới elastic theo cú pháp mong muốn để lấy kết quả về.

Mô hình ELK trong Elasticsearch

7. Áp dụng vào thực tế (hệ thống admage):

7.2 Demo với hệ thống admage:

- Log của hệ thống được "deliver" ghi lại ở đường dẫn `"/var/admage/count"`. Như thế chúng ta sẽ config logstash trở đến đường dẫn này để lấy được log.



Mô hình ELK trong Elasticsearch

7. Áp dụng vào thực tế (hệ thống admage):

7.2 Demo với hệ thống admage:

- Sau khi tạo các screen và start logstash, elastic, kibana chúng ta tạo 1 file config bất kỳ. Ví dụ : "fam_elasticsearch.conf" trong đường dẫn : **/etc/logstash/conf.d/**

```
y@duyba: /etc/logstash/conf.d
input {
  file{

    path => "/var/admage/count/*.log"
    type => "syslog"
    start_position => "beginning"
  }
}
filter{
}
output{
  elasticsearch {
    hosts => ["localhost:9200"]
    index => "fam_elasticsearch"
  }
}
~
~
~
~
~
~
```

Mô hình ELK trong Elasticsearch

7. Áp dụng vào thực tế (hệ thống admage):

7.2 Demo với hệ thống admage:

- Chạy lệnh để logstash đẩy log vào elastic. Mỗi lần có bản ghi mới, logstash sẽ lắng nghe và tự động update.

```
entropy@duybac:~/Desktop$ cd /usr/share/logstash/bin/
entropy@duybac:~/share/logstash/bin$ sudo ./logstash -f /etc/logstash/conf.d/fan_project.conf
[sudo] password for entropy:
WARNING: Could not find logstash.yml which is typically located in $LS_HOME/config or /etc/logstash. You can specify the path using --path.settings. Continuing using the defaults
Could not find log4j2 configuration at path /usr/share/logstash/config/log4j2.properties. Using default config which logs errors to the console
[WARN ] 2019-07-18 16:19:55.901 [LogStash::Runner] multilocal - Ignoring the 'pipelines.yml' file because modules or command line options are specified
[INFO ] 2019-07-18 16:19:55.935 [LogStash::Runner] runner - Starting Logstash {"logstash.version"=>"6.8.1"}
[INFO ] 2019-07-18 16:20:03.927 [Converge PipelineAction::Create<main>] pipeline - Starting pipeline {:pipeline_id=>"main", "pipeline.workers"=>4, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50}
[INFO ] 2019-07-18 16:20:04.487 [[main]-pipeline-manager] elasticsearch - Elasticsearch pool URLs updated {:changes=>{:removed=>[], :added=>[http://localhost:9200/]}]
[WARN ] 2019-07-18 16:20:04.711 [[main]-pipeline-manager] elasticsearch - Restored connection to ES instance {:url=>"http://localhost:9200/"}
[INFO ] 2019-07-18 16:20:04.878 [[main]-pipeline-manager] elasticsearch - ES Output version determined {:es_version=>6}
[WARN ] 2019-07-18 16:20:04.881 [[main]-pipeline-manager] elasticsearch - Detected a 6.x and above cluster: the 'type' event field won't be used to determine the document _type {:es_version=>6}
[INFO ] 2019-07-18 16:20:04.908 [[main]-pipeline-manager] elasticsearch - New Elasticsearch output {:class=>"LogStash::Outputs::ElasticSearch", :hosts=>["//localhost:9200"]}
[INFO ] 2019-07-18 16:20:04.921 [Ruby-0-Thread-5: :1] elasticsearch - Using default mapping template
[INFO ] 2019-07-18 16:20:04.971 [Ruby-0-Thread-5: :1] elasticsearch - Attempting to install template {:manage_template=>{"template"=>"logstash-*", "version"=>60001, "settings"=>{"index.refresh_interval"=>"5s"}, "mappings"=>{"_default_"=>{"dynamic_templates"=>[{"message_field"=>{"path_match"=>"message", "match_mapping_type"=>"string", "mapping"=>{"type"=>"text", "norms"=>false}}, {"string_fields"=>{"match"=>"*", "match_mapping_type"=>"string", "mapping"=>{"type"=>"text", "norms"=>false, "fields"=>{"keyword"=>{"type"=>"keyword", "ignore_above"=>256}}}], "properties"=>{"@timestamp"=>{"type"=>"date"}, "@version"=>{"type"=>"keyword"}, "geoip"=>{"dynamic"=>true, "properties"=>{"ip"=>{"type"=>"ip"}, "location"=>{"type"=>"geo_point"}, "latitude"=>{"type"=>"half_float"}, "longitude"=>{"type"=>"half_float"}}}}}}
[INFO ] 2019-07-18 16:20:05.334 [[main]-pipeline-manager] file - No sincedb_path set, generating one based on the 'path' setting {:sincedb_path=>"/usr/share/logstash/data/plugins/inputs/file/.sincedb_58df9f57a76f89e6385342f47bf18822", :path=>["/var/admage/count/*.log"]}
[INFO ] 2019-07-18 16:20:05.384 [Converge PipelineAction::Create<main>] pipeline - Pipeline started successfully {:pipeline_id=>"main", :thread=>"#<Thread:0x1a6bc411 sleep>"}
[INFO ] 2019-07-18 16:20:05.507 [Ruby-0-Thread-1: /usr/share/logstash/lib/bootstrap/environment.rb:6] agent - Pipelines running {:count=>1, :running_pipelines=>[:main], :non_running_pipelines=>[]}
[INFO ] 2019-07-18 16:20:05.507 [[main]<file>] observingtail - START!, creating Discoverer, Watch with file and sincedb collections
[INFO ] 2019-07-18 16:20:06.010 [Api Webserver] agent - Successfully started Logstash API endpoint {:port=>9601}
```


Mô hình ELK trong Elasticsearch

7. Áp dụng vào thực tế (hệ thống admage):

7.2 Demo với hệ thống admage:

- Vào kibana sẽ thấy kết quả đã được đẩy vào elastic search:

```
GET /fam_elasticsearch/_search
{
  "query": {
    "match_all": {
      }
    }
  }
}

{
  "hits" : {
    "total" : 16,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "fam_elasticsearch",
        "_type" : "doc",
        "_id" : "dAX6_msBAqCxfKbBJkOZ",
        "_score" : 1.0,
        "_source" : {
          "message" : "\"\"1 0 2019-07-12_13-03-00 1562904225
admx6be459e867xa7e 1 0:0:0:0:0:0:1 1 2 61 0 743 0 0 0 0 0 16
568 0 0 0 0
%SFsite%3D61%26%SFloc%3D743%26%SFmstype%3D1%26%SFwidth%3D%26%SFheig
ht%3D0 Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML like Gecko) Chrome/75.0.3770.100 Safari/537.36 0 0 0.0
0.0 0 0 0 0 0 0 0 0 \"",
"@version" : "1",
"path" : "/var/admage/count/2_2019-07-12_13-03-00_1_p_sell_localhost
.log",
"type" : "syslog",
"host" : "duybac",
"@timestamp" : "2019-07-17T08:08:54.001Z"
}
},
{
  "_index" : "fam_elasticsearch",
  "_type" : "doc",
  "_id" : "cwX6_msBAqCxfKbBJUOn",
  "_score" : 1.0,
  "_source" : {
    "message" : "\"\"1 0 2019-07-04_17-36-00 1562229365
admx6bbc2059b3xc91 1 0:0:0:0:0:0:1 1 2 61 0 743 0 0 0 0 0 16
568 0 0 0 0
%SFsite%3D61%26%SFloc%3D743%26%SFmstype%3D1%26%SFwidth%3D%26%SFheig
ht%3D0 Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML like Gecko) Chrome/75.0.3770.100 Safari/537.36 0 0 0.0
0.0 0 0 0 0 0 0 0 0 \"",
"@version" : "1",
"path" : "/var/admage/count/2_2019-07-04_17-36-00_1_p_sell_localhost
.log",
"type" : "syslog",
"host" : "duybac",
"@timestamp" : "2019-07-17T08:08:54.001Z"
}
}
```

Mô hình ELK trong Elasticsearch

7. Áp dụng vào thực tế (hệ thống admage):

7.2 Demo với hệ thống admage:

- Thử query với trường "_id":

```

1 GET /fan_elasticsearch/_search
2 {
3   "query": {
4     "match": {
5       "_id": "dAX6_msBAqCxfkBjK0Z"
6     }
7   }
8 }
9
10 {
11   "took": 155,
12   "timed_out": false,
13   "_shards": {
14     "total": 5,
15     "successful": 5,
16     "skipped": 0,
17     "failed": 0
18   },
19   "hits": {
20     "total": 1,
21     "max_score": 1.0,
22     "hits": [
23       {
24         "_index": "fan_elasticsearch",
25         "_type": "doc",
26         "_id": "dAX6_msBAqCxfkBjK0Z",
27         "_score": 1.0,
28         "_source": {
29           "message": "\"\"1 0 2019-07-12 13-03-00 1562904225 admx0be459e867xa7e 1 0:0:0:0:0:0:1 1 2 61 0 743 0 0
30             0 0 0 0 16 568 0 0 0 0 %SFsite%3D61%26%SFloc%3D743%26%SFmstype%3D1%26%SFwidth%3D%26%SFheight%3D 0
31             Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML like Gecko) Chrome/75.0.3770.100 Safari/537.36
32             0 0 0 0 0.0 0 0 0 0 0 0 0 0 0 \"\"",
33           "@version": "1",
34           "path": "/var/admimage/count/2_2019-07-12-13-03-00_1_p_sell_localhost.log",
35           "type": "syslog",
36           "host": "duybac",
37           "@timestamp": "2019-07-17T08:08:54.001Z"
38         }
39       }
40     ]
41   }
42 }

```

Mô hình ELK trong Elasticsearch

7. Áp dụng vào thực tế (hệ thống admage):

7.2 Demo với hệ thống admage:

- Có rất nhiều kiểu query cũng như thống kê mà elastic search hỗ trợ. Tùy thuộc vào bài toán cũng như nhu cầu sử dụng, chúng ta sẽ lựa chọn kiểu query phù hợp. Doc hướng dẫn trên trang của elastic dưới đây :

<https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-match-query.html>

-> Chú ý: Với mỗi phiên bản cập nhật, có thể có 1 số thay đổi nhỏ cho cú pháp query.

Mô hình ELK trong Elasticsearch

7. Áp dụng vào thực tế (hệ thống admage):

7.3 Dọn dẹp elastic search:

- Sau một thời gian, elastic sẽ đầy lên, với những bản ghi cũ không có giá trị sử dụng, việc chúng ta nghĩ đến lúc này là remove nó. Có nhiều cách làm việc này. Chúng ta có thể dùng REST query thẳng lên elastic để delete. Cách hay hơn là dùng plugin hỗ trợ, đó là elastic-curator.

- Chúng ta sẽ đặt job, sau mỗi khoảng thời gian plugin này sẽ chạy và xóa đi 1 lượng bản ghi theo chỉ định. Chi tiết, cách cài đặt cũng như đặt cron tab xem tại link sau :

<http://www.ragingcomputer.com/2014/02/removing-old-records-for-logstash-elasticsearch-kibana>

Mô hình ELK trong Elasticsearch

8. Kết thúc:

- Do mô hình node master khá phức tạp, cần cài trên nhiều môi trường (server), nên trong phạm vi slide chưa thể liệt kê hết được. Vì vậy nếu có nhu cầu thắc mắc vui lòng liên hệ skype: **duybac512** mình sẽ hướng dẫn triển khai.
- Cú pháp query có thể tìm trên doc trang chủ của elastic search để viết.
- Chân thành cảm ơn nếu bạn đọc tới dòng này :D