Kenu 허광남 2016.06.25 판교 kosta

Elastic

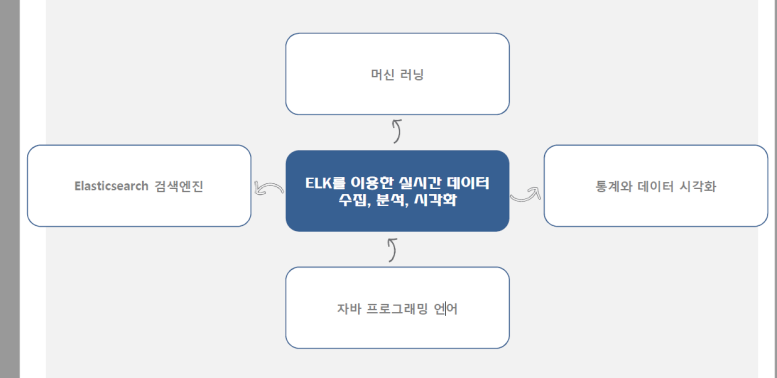
https://www.elastic.co/  
<http://bit.ly/kosta-elk>

|  |  |
| --- | --- |
| <https://www.dropbox.com/s/v6emkuos0792fs3/ELK%EB%A5%BC%20%EC%9D%B4%EC%9A%A9%ED%95%9C%20%EC%8B%A4%EC%8B%9C%EA%B0%84%20%EB%8D%B0%EC%9D%B4%ED%84%B0%20%EC%88%98%EC%A7%91%2C%20%EB%B6%84%EC%84%9D%2C%20%EC%8B%9C%EA%B0%81%ED%99%94%20%EA%B5%90%EC%9E%AC_20160625_V1.pdf?dl=0a> | 교재 |
| **필요 소프트웨어** |  |
| jdk 1.8 | [java.sun.com](http://java.sun.com) |
| git bash | [git-scm.com](http://git-scm.com) |
| elasticsearch 2.3.x | [elastic.co](http://elastic.co) |
| kibana 4.x | [elastic.co](http://elastic.co) |
| logstash 2.3.x | [elastic.co](http://elastic.co) |
| filebeat | [elastic.co](http://elastic.co) |
| <http://52.79.194.183:5601/> | okky.kr elk |

엘라스틱 자바로 만들어져있음

자바 EE 가디언즈. 출범 (오라클 이새끼들..)

책

 ****

**ELK를 사용하면 기획자에게 대들수있다**

**데이터로써 증명 할수 있다.ㅋㅋ**

**개요**

|  |  |  |
| --- | --- | --- |
| 엘라스틱 스택 | ELK 검색엔진 | Elastic에서 개발한 ElasticSearch(검색엔진)  Elasticsearch는 Apache Lucene 기반의 실시간 분산 검색 엔진 |
| 로그 수집기 | Logstash는 각종 로그를 가져와 JSON형태로 만들어 Elasticsearch로 전송  점점 비츠(Beats)쪽으로 기능이 넘어가고 있다. – 김종민  JRuby 스크립트안쪽 필드..프로그래밍 |
| 시각화도구 | Kibana는 Elasticsearch에 저장된 Data를 사용자에게 Dashboard 형태로 보여주는 솔루션(node.js 들어가있다) |
| 구성된 수집, 검색, 시각화를 실시간으로 처리할 수 있는  오픈소스 패키지이다.  Elasticsearch(검색엔진) + Logstash(로그수집기) + Kibana(시각화도구)  Beats를 추가해 Elastic Stack이라 부름  로그에 적제할수 있는것들은 ELK쪽에서 모든 것을 로깅할수 있다. | |
| 특징 |  | |

|  |  |
| --- | --- |
| GA | Google Analytics  (직접 소스단에 스크립트가 박혀야된다, 시간대 별, ip별로는 볼수 없어서 좀 ..)  사용자의 front 화면의 정보를 가져올수도 있다. |
| DataViz | 넷플릭스에서 유행하는 단어 Data Visualization |

ELK

Elastic Stack

L

o

g

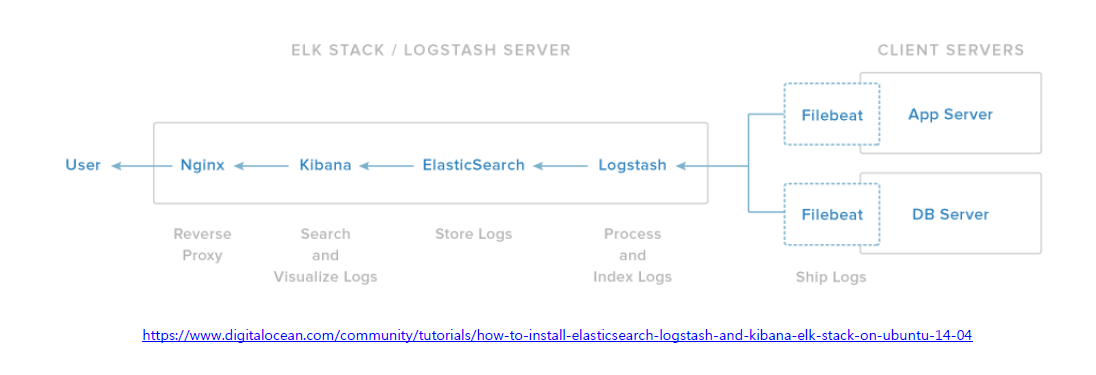
S

t

a

s

h

****

<https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elk-stack-on-ubuntu-14-04>

****

**엑세스로그 처리**

|  |  |
| --- | --- |
| Logstash | |
| • COMMONAPACHELOG | %{IPORHOST:clientip} %{HTTPDUSER:ident} %{USER:auth}  \[%{HTTPDATE:timestamp}\] "(?:%{WORD:verb} %{NOTSPACE:request}(?:  HTTP/%{NUMBER:httpversion})?|%{DATA:rawrequest})" %{NUMBER:response}  (?:%{NUMBER:bytes}|-) |
|  |
| • COMBINEDAPACHELOG | %{COMMONAPACHELOG} %{QS:referrer} %{QS:agent} |
|  |

https://github.com/logstash-plugins/logstash-patterns-core/blob/master/patterns/grok-patterns

**AWS 서비스 가입**

* http://aws.amazon.com EC2 가입

**실시간 데이터 수집, 검색, 분석, 시각화 절차의 이해**

1. **데이터 선정**
2. **채널 (open api등등 시스템 로그등)**
3. **파싱 패턴 매칭**
4. **시각화 도구 선정**
5. **차트 시각화**

**팁**

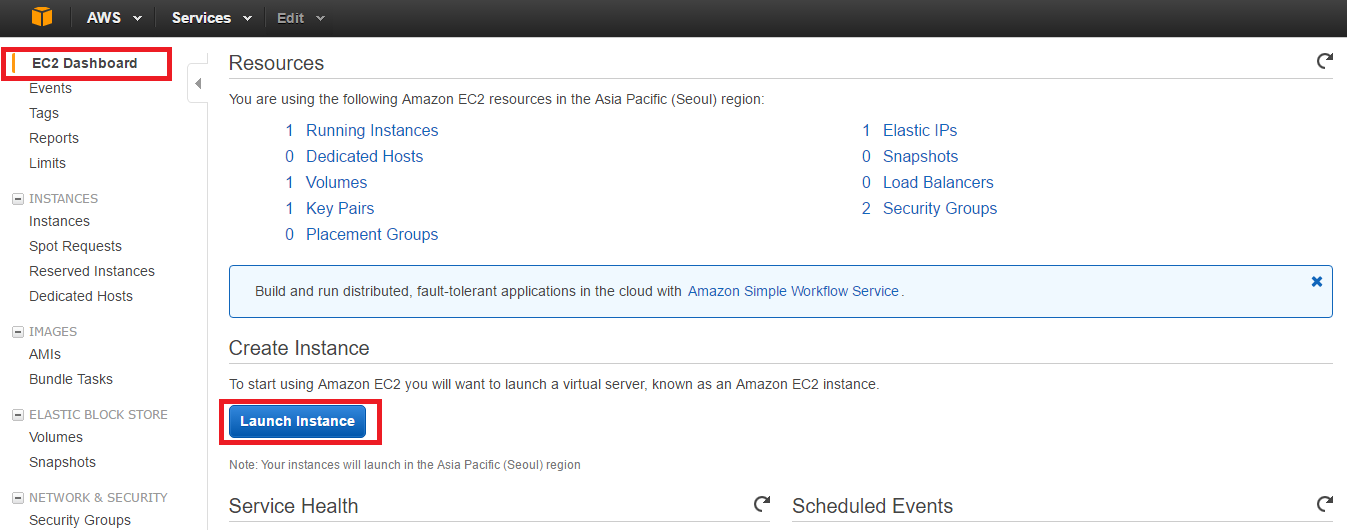
* 통계, R언어 쪽을 다시한번 찾아봐라
* 머신러닝 쪽으로도 갈수도 있으므로 (파이썬 이용하는것들이 많다)
* Semantic Versioning 소개
  + https://spoqa.github.io/2012/12/18/semantic-versioning.html

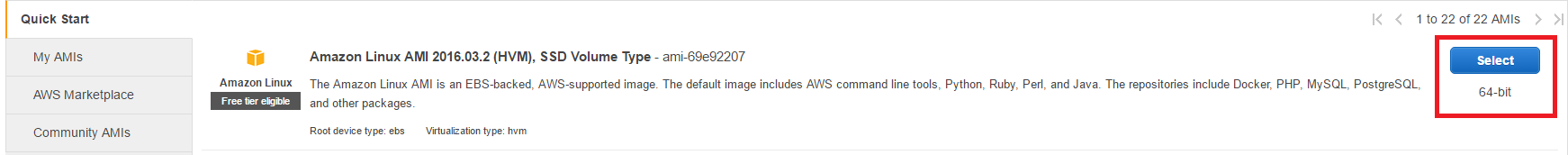
**주의**

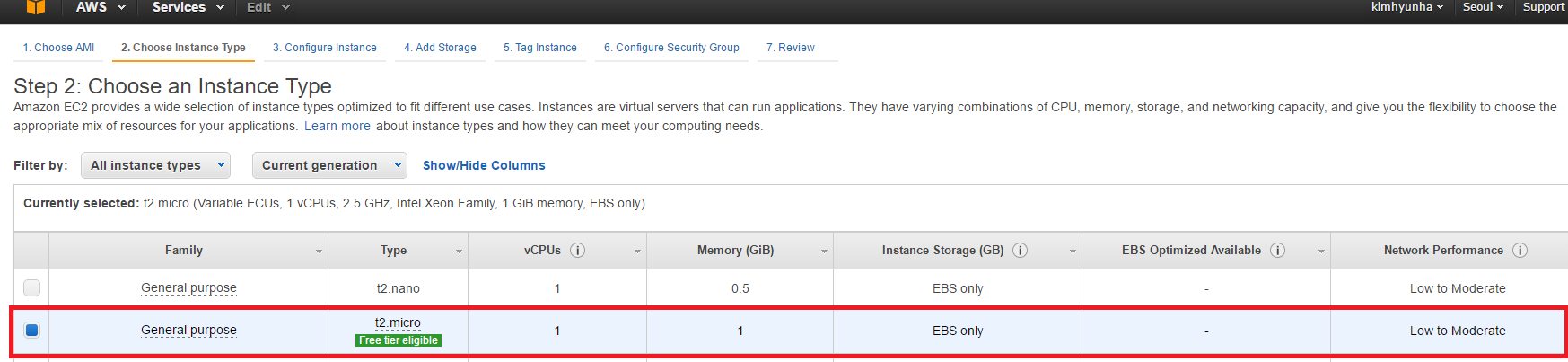
* 버전 영향력이 있으므로 버전 조합잘해야된다.

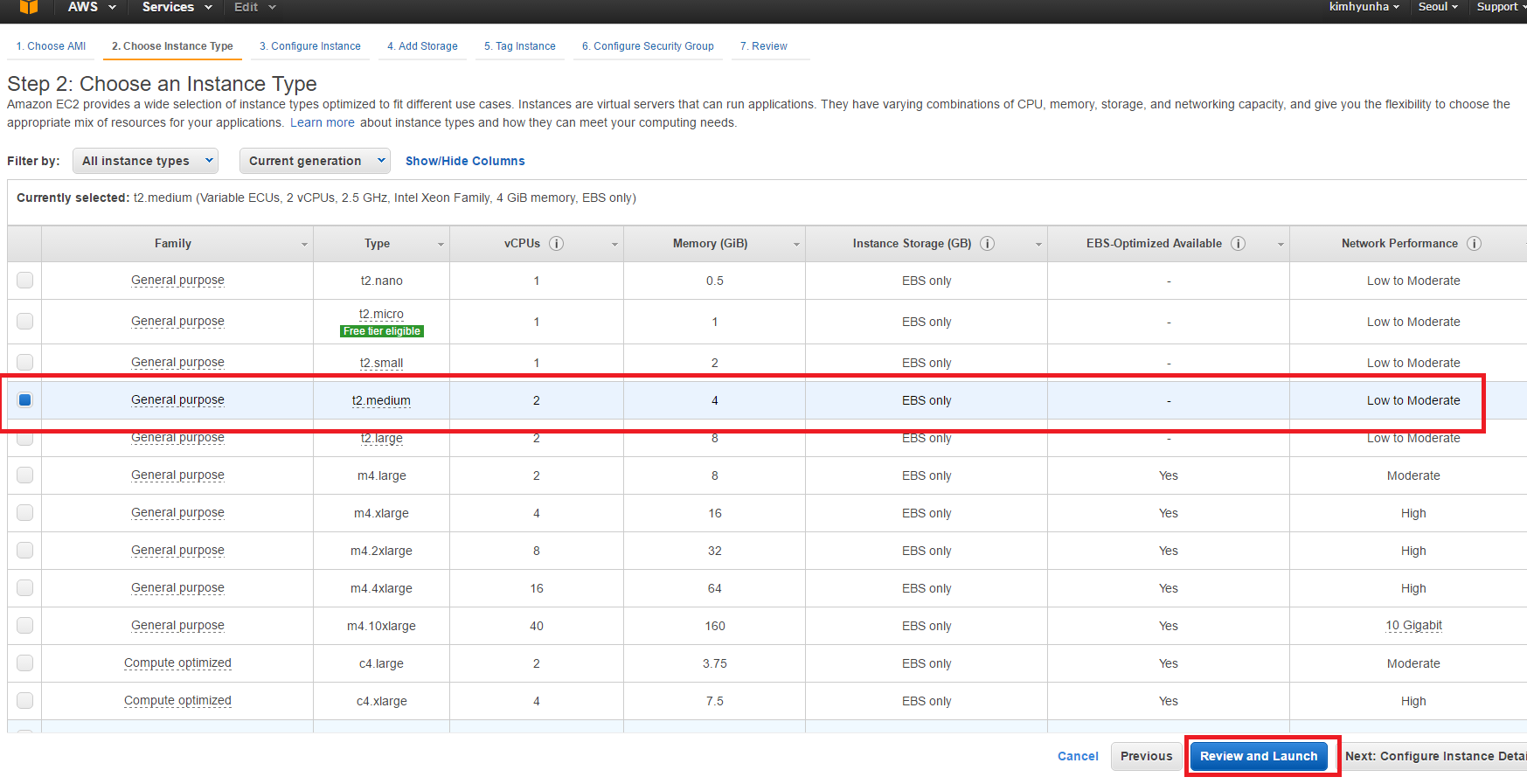
**실습**

aws EC2

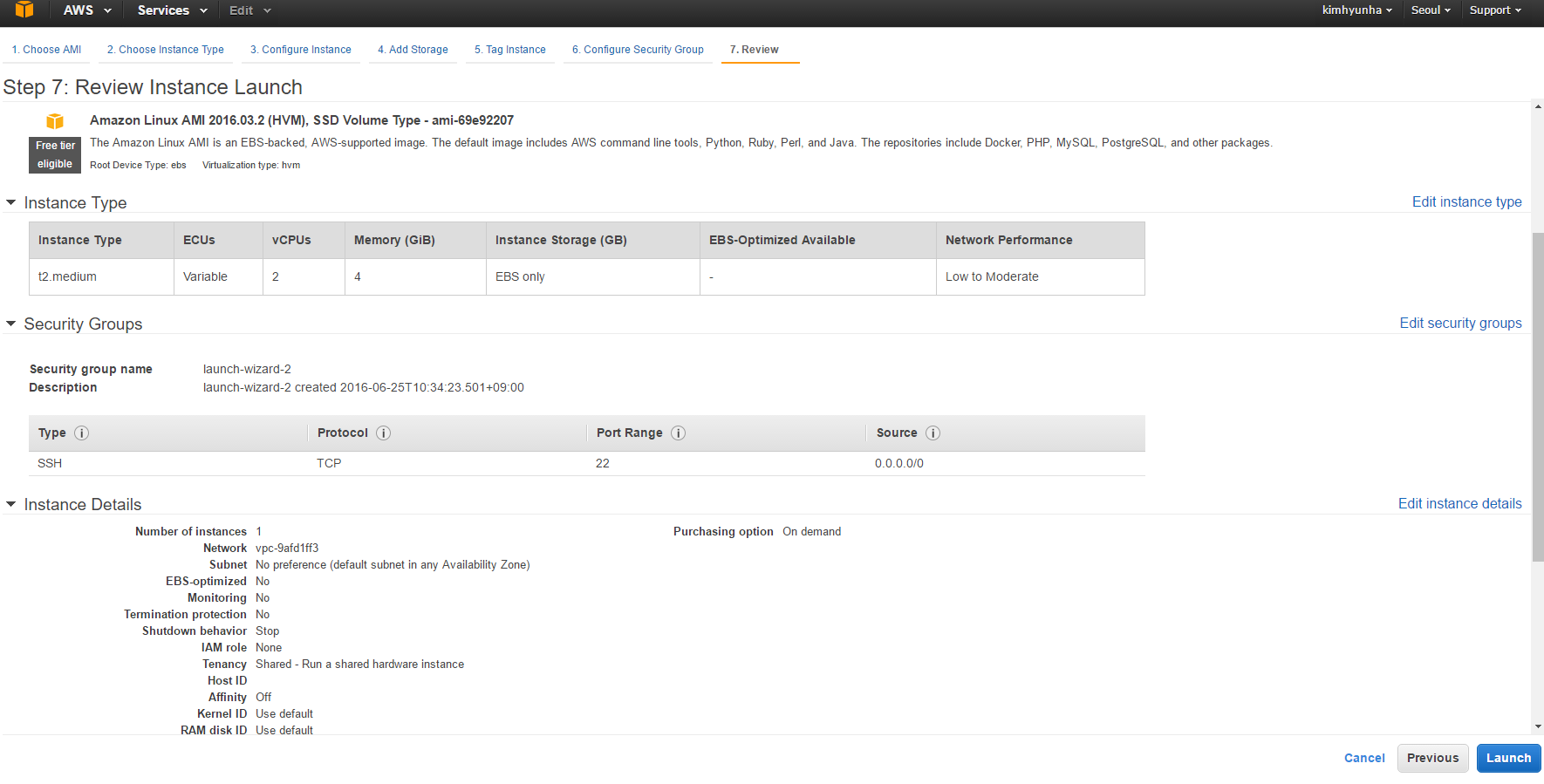


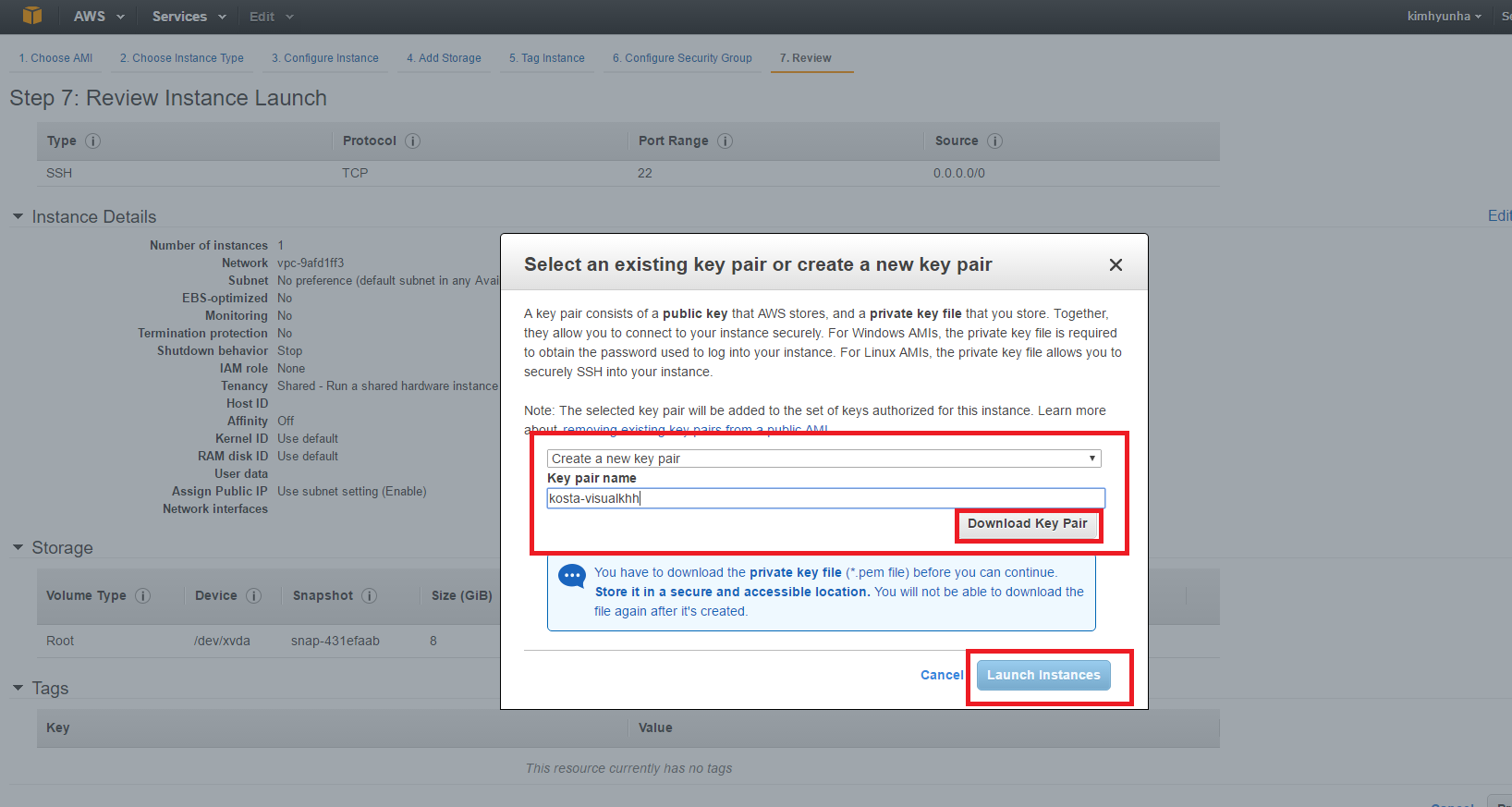


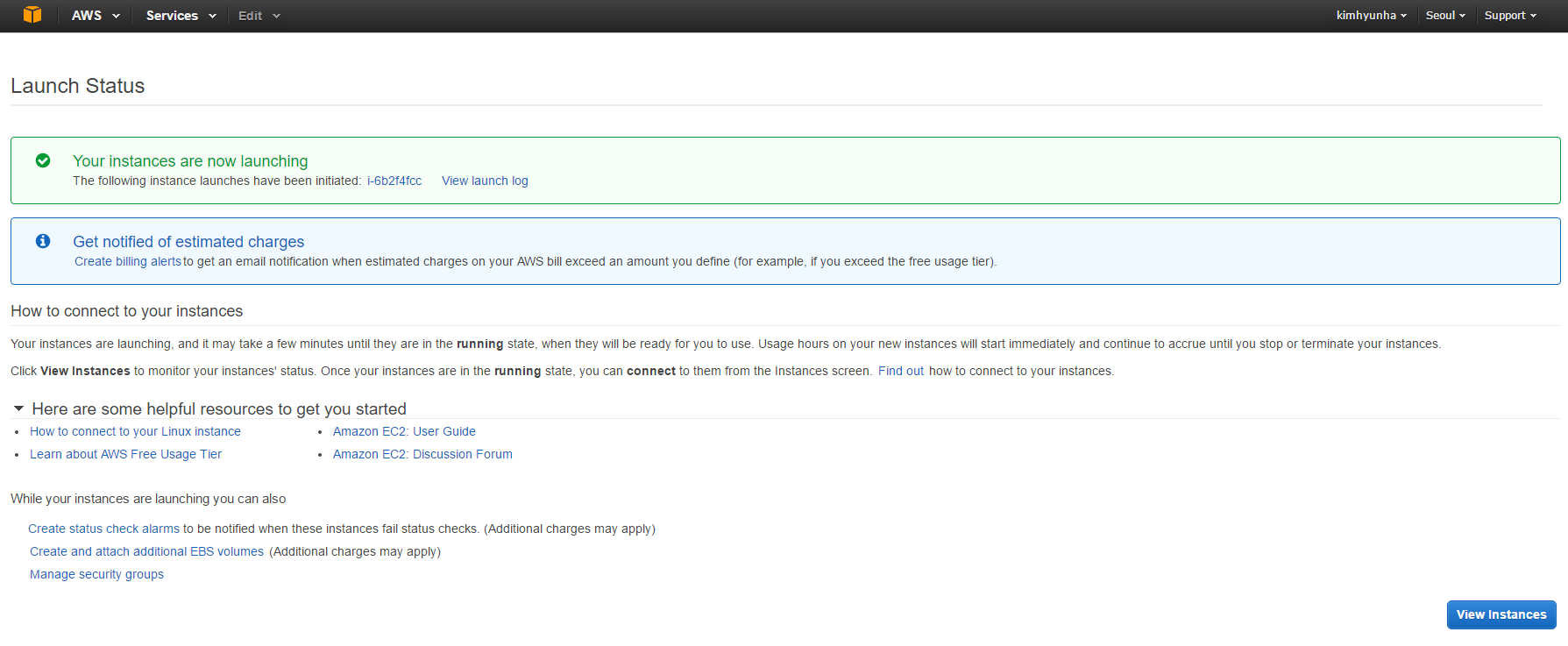
이건 무료 1년



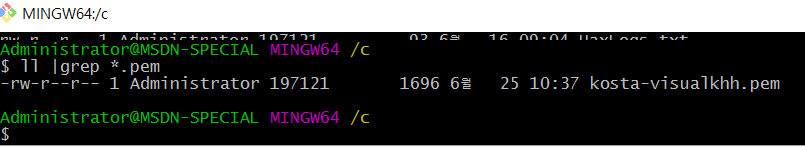
엘라스틱 하려면 위에꺼 사양좋은걸로..ㅠ\_ㅠ

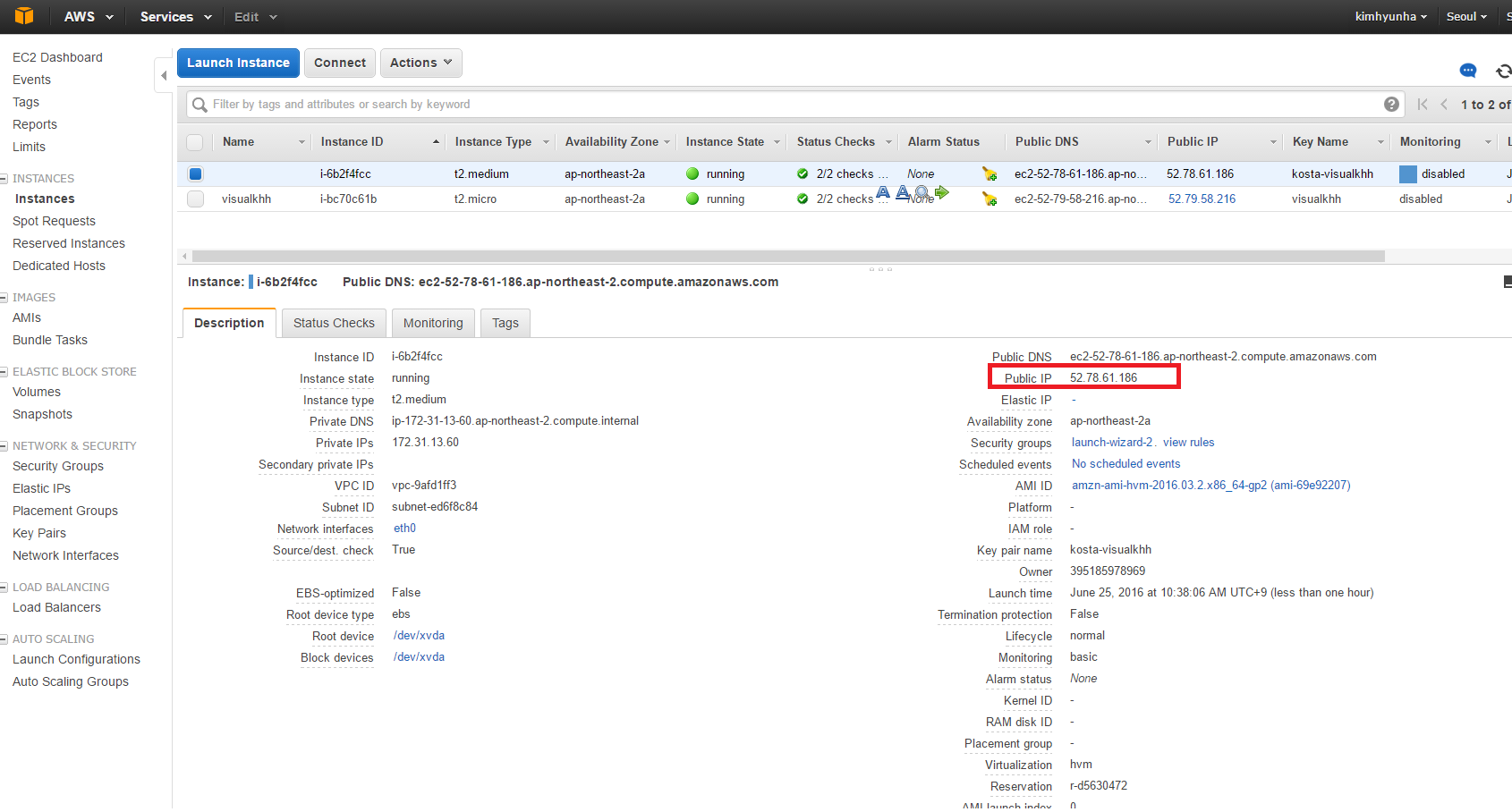


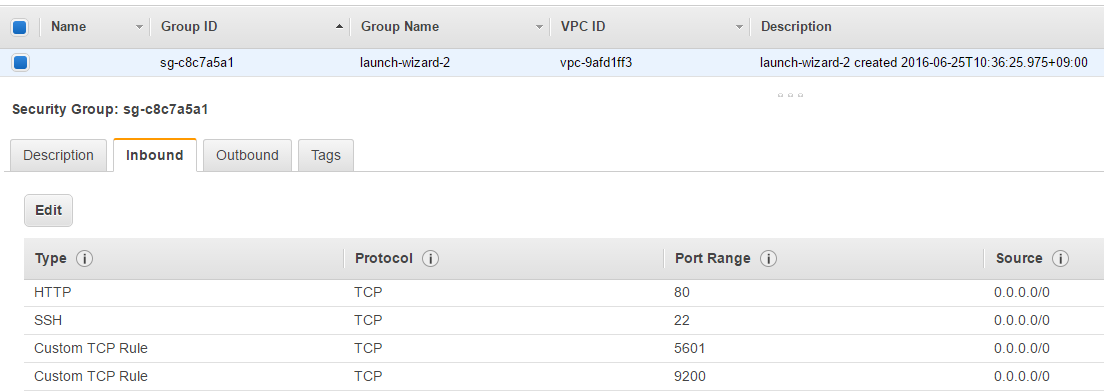
위 런치를 누르면 과금 시작 다운로드(키 받아라) 어디에 흘리고 다니지말어라



git bash 에서 받은 kosta-visualkhh.pem으로 ssh를 붙어보자







* EC2 Security Groups
* 외부 접근 포트 추가(inbound)
  + http(80)
  + elasticsearch(9200)
  + kibana(5601)

|  |
| --- |
| Administrator@MSDN-SPECIAL MINGW64 ~/elkdev  $ ll  total 8  -r--r--r-- 1 Administrator 197121 1696 6월 25 10:37 **kosta-visualkhh.pem**  Administrator@MSDN-SPECIAL MINGW64 ~/elkdev  $ **ssh -i ~/elkdev/kosta-visualkhh.pem ec2-user@52.78.61.186**  [ec2-user@ip-172-31-13-60 ~]$ **sudo yum update -y**  서버 업데이트… |

in aws

설치 4가지 (okdevtv.com/mib/elk/elk)

* nginx (아파치 보다 경량화되어 나온 웹서버, 이벤트방식이라 빠르다)
  + <http://okdevtv.com/mib/nginx/nginx>

|  |
| --- |
| sudo yum install nginx -y  sudo service nginx start  curl -i http://localhost  sudo chown -R ec2-user:ec2-user /var/log/nginx /usr/share/nginx/html  echo "<h1>Hello World</h1>" > /usr/share/nginx/html/hello.html |

|  |
| --- |
| [ec2-user@ip-172-31-13-60 ~]$ mkdir ~/local  [ec2-user@ip-172-31-13-60 ~]$ cd ~/local/  sudo yum install nginx -y  sudo service nginx start  curl -i http://localhost  sudo chown -R ec2-user:ec2-user /var/log/nginx /usr/share/nginx/html  echo "<h1>Hello World</h1>" > /usr/share/nginx/html/hello.html |
| curl -i http://localhost/hello.html  HTTP/1.1 200 OK  Server: nginx/1.8.1  Date: Sat, 25 Jun 2016 02:27:46 GMT  Content-Type: text/html  Content-Length: 21  Last-Modified: Sat, 25 Jun 2016 02:22:16 GMT  Connection: keep-alive  ETag: "576dead8-15"  Accept-Ranges: bytes  <h1>Hello World</h1> |
| 제대로 파일 호출되는지 웹에서 확인 |
|  |
| 설치 |
| cd ~/local  wget https://download.elastic.co/elasticsearch/release/org/elasticsearch/distribution/tar/elasticsearch/2.3.3/elasticsearch-2.3.3.tar.gz  tar xvfz elasticsearch-2.3.3.tar.gz  ln -s elasticsearch-2.3.3 elasticsearch  cd elasticsearch  ………  [ec2-user@ip-172-31-13-60 elasticsearch]$ ll  total 44  drwxrwxr-x 2 ec2-user ec2-user 4096 Jun 25 02:33 bin  drwxrwxr-x 2 ec2-user ec2-user 4096 Jun 25 02:33 config  drwxrwxr-x 2 ec2-user ec2-user 4096 Jun 25 02:33 lib  -rw-rw-r-- 1 ec2-user ec2-user 11358 Jan 27 12:53 LICENSE.txt  drwxrwxr-x 5 ec2-user ec2-user 4096 May 17 15:48 modules  -rw-rw-r-- 1 ec2-user ec2-user 150 May 12 13:24 NOTICE.txt  -rw-rw-r-- 1 ec2-user ec2-user 8700 May 12 13:24 README.textile  [ec2-user@ip-172-31-13-60 elasticsearch]$ |
|  |
| 엘라스틱서치 시작 |
| cd ~/local/elasticsearch  bin/elasticsearch –d  [ec2-user@ip-172-31-13-60 elasticsearch]$ cd ~/local/elasticsearch  [ec2-user@ip-172-31-13-60 elasticsearch]$ bin/elasticsearch -d  [ec2-user@ip-172-31-13-60 elasticsearch]$  [ec2-user@ip-172-31-13-60 elasticsearch]$ ps -ef |grep elasticsearch  ec2-user 7188 1 39 02:35 pts/0 00:00:04 /usr/lib/jvm/jre/bin/java -Xms256m -Xmx1g -Djava.awt.headless=true -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -XX:+HeapDumpOnOutOfMemoryError -XX:+DisableExplicitGC -Dfile.encoding=UTF-8 -Djna.nosys=true -Des.path.home=/home/ec2-user/local/elasticsearch -cp /home/ec2-user/local/elasticsearch/lib/elasticsearch-2.3.3.jar:/home/ec2-user/local/elasticsearch/lib/\* org.elasticsearch.bootstrap.Elasticsearch start -d  ec2-user 7240 2639 0 02:35 pts/0 00:00:00 grep --color=auto elasticsearch |
|  |
| 설정하기 설정변경하기 |
| vi config/elasticsearch.yml  network.host:0.0.0.0  kill -9 7188  [ec2-user@ip-172-31-13-60 elasticsearch]$ kill -9 7188  [ec2-user@ip-172-31-13-60 elasticsearch]$ ps -ef |grep elasticsearch  ec2-user 7247 2639 0 02:42 pts/0 00:00:00 grep --color=auto elasticsearch |
|  |
| 재실행 확인 |
| [ec2-user@ip-172-31-13-60 elasticsearch]$ kill -9 7188  [ec2-user@ip-172-31-13-60 elasticsearch]$ ps -ef |grep elasticsearch  ec2-user 7247 2639 0 02:42 pts/0 00:00:00 grep --color=auto elasticsearch  [ec2-user@ip-172-31-13-60 elasticsearch]$ bin/elasticsearch -d  [ec2-user@ip-172-31-13-60 elasticsearch]$ curl -i http://localhost:9200/  HTTP/1.1 200 OK  Content-Type: application/json; charset=UTF-8  Content-Length: 321  {  "name" : "Chan Luichow",  "cluster\_name" : "elasticsearch",  "version" : {  "number" : "2.3.3",  "build\_hash" : "218bdf10790eef486ff2c41a3df5cfa32dadcfde",  "build\_timestamp" : "2016-05-17T15:40:04Z",  "build\_snapshot" : false,  "lucene\_version" : "5.5.0"  },  "tagline" : "You Know, for Search"  }  [ec2-user@ip-172-31-13-60 elasticsearch]$ |
|  |
| Kibana 설치 |
| cd ~/local  wget https://download.elastic.co/kibana/kibana/kibana-4.5.1-linux-x64.tar.gz  tar xvfz kibana-4.5.1-linux-x64.tar.gz  ln -s kibana-4.5.1-linux-x64 kibana  cd kibana |
|  |
| Kibana 실행 백그라운드 실행 (elasticsearch 가 실행되있어야된다) |
| cd ~/local/kibana  bin/kibana  nohup bin/kibana & |
|  |
| Logstash 설치 (설정 파일이 필요하다 로그관련) |
| cd ~/local  wget https://download.elastic.co/logstash/logstash/logstash-2.3.2.tar.gz  tar xvfz logstash-2.3.2.tar.gz  ln -s logstash-2.3.2 logstash  cd logstash |
|  |
| Logstash conf 파일 생성 |
| cd ~/local/logstash  mkdir logconf  vi logconf/nginx.conf |
| input {  file {  path => "/var/log/nginx/access.log"  start\_position => beginning  }  }  filter {  grok {  match => { "message" => "%{COMBINEDAPACHELOG}"}  }  geoip {  source => "clientip"  }  }  output {  elasticsearch {}  stdout {}  }  위의 화살표 => 는 다른 언어와 다르게 msg => info 라면 msg는 info 값을 가진다(바라본다) 처럼 이해해야된다 |
|  |
| logstash 실행 |
| cd ~/local/logstash  # debug  bin/logstash -f logconf/nginx.conf --debug  # run  bin/logstash -f logconf/nginx.conf  # background run  nohup bin/logstash -f logconf/nginx.conf & |
|  |
|  |
| 확인… http://52.78.61.186:5601/ |
|  |

옥희 로그를 보고 분석해보장

|  |
| --- |
|  |
| cd ~/local/  mkdir okky  curl -O <http://okky.kr/examples/access.log-20160624.gz>  [ec2-user@ip-172-31-13-60 okky]$ cp access.log-20160624.gz access-okky.log.gz  [ec2-user@ip-172-31-13-60 okky]$ ll  total 16360  -rw-rw-r-- 1 ec2-user ec2-user 8372739 Jun 25 04:56 access.log-20160624.gz  -rw-rw-r-- 1 ec2-user ec2-user 8372739 Jun 25 04:58 access-okky.log.gz  [ec2-user@ip-172-31-13-60 okky]$ gunzip access-okky.log.gz  [ec2-user@ip-172-31-13-60 okky]$ ll  total 113864  -rw-rw-r-- 1 ec2-user ec2-user 8372739 Jun 25 04:56 access.log-20160624.gz  -rw-rw-r-- 1 ec2-user ec2-user 108217793 Jun 25 04:58 access-okky.log  [ec2-user@ip-172-31-13-60 okky]$ cp access-okky.log /var/log/nginx |
| 필요 유틸 추가 dstat 설치 |
| sudo yum install dstat –y  dstat 60 |
|  |
| 데이터 강제 적제, 확인 |
| cd ~/local/elasticsearch/data/elasticsearch/nodes/0/indices/  cd ~/local/logstash/logconf  vi nginx.conf  #logstash 패턴변경    재시작  ps -ef| grep logstash  ec2-user 7347 2639 0 02:54 pts/0 00:00:51 /usr/lib/jvm/jre/bin/java -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -Djava.awt.headless=true -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -XX:+HeapDumpOnOutOfMemoryError -Xmx1g -Xss2048k -Djffi.boot.library.path=/home/ec2-user/local/logstash-2.3.2/vendor/jruby/lib/jni -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -Djava.awt.headless=true -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/home/ec2-user/local/logstash/heapdump.hprof -Xbootclasspath/a:/home/ec2-user/local/logstash-2.3.2/vendor/jruby/lib/jruby.jar -classpath : -Djruby.home=/home/ec2-user/local/logstash-2.3.2/vendor/jruby -Djruby.lib=/home/ec2-user/local/logstash-2.3.2/vendor/jruby/lib -Djruby.script=jruby -Djruby.shell=/bin/sh org.jruby.Main --1.9 /home/ec2-user/local/logstash/lib/bootstrap/environment.rb logstash/runner.rb agent -f logconf/nginx.conf --debug  ec2-user 29724 7841 0 05:07 pts/1 00:00:00 grep --color=auto logstash  kill -9 7347  cd ~/local/  [ec2-user@ip-172-31-13-60 local]$ cd logstash  [ec2-user@ip-172-31-13-60 logstash]$ nohup bin/logstash -f logconf/nginx.conf &  dstat 60  [ec2-user@ip-172-31-13-60 logstash]$ dstat 60  ----total-cpu-usage---- -dsk/total- -net/total- ---paging-- ---system--  usr sys idl wai hiq siq| read writ| recv send| in out | int csw  1 0 99 0 0 0| 26k 195k| 0 0 | 0 0 | 187 255  72 2 10 15 0 0| 0 **9093k**|1461B 4108B| 0 0 |2126 4453  쓰고있음 |
| Kibana확인 |
|  |

IP 검색

검색어는 따옴표("")로 감싼다.

"123.123.123.123"

clientip : "123.123.123.123"

URL 검색

"/order/form.html"

request : "/order/form.html"

device 검색 크롤러 제외

데이터 수집시 useragent 플러그인이 된 경우 가능

-device: "Spider"

여러 URL 검색

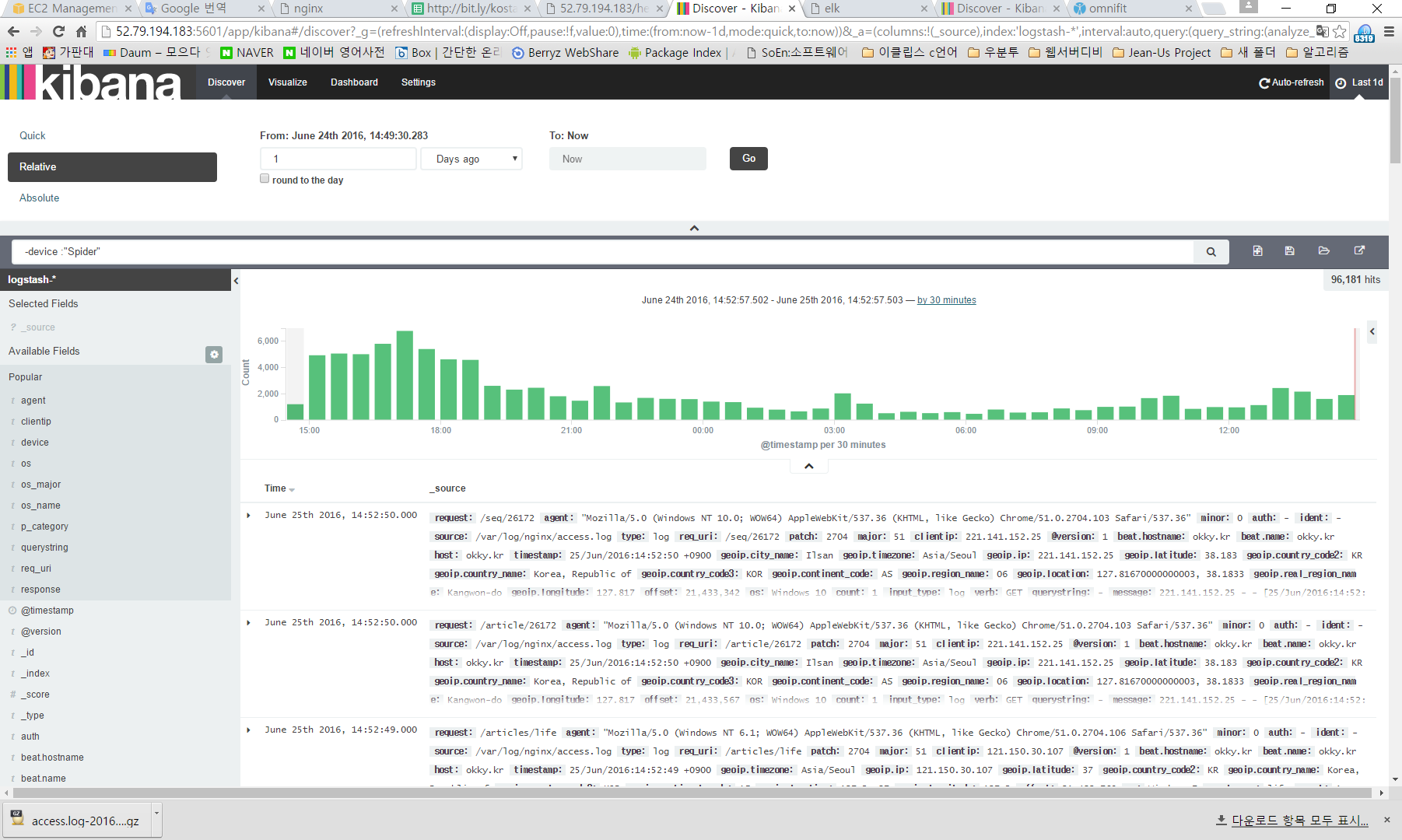
"/order/form.html" OR "/order/end.html"

OR 또는 AND는 대문자 dstat 설치

<https://dl.dropboxusercontent.com/u/2385737/Kibana-basic.pdf>

<https://okdevtv.com/mib/elk/kibana>

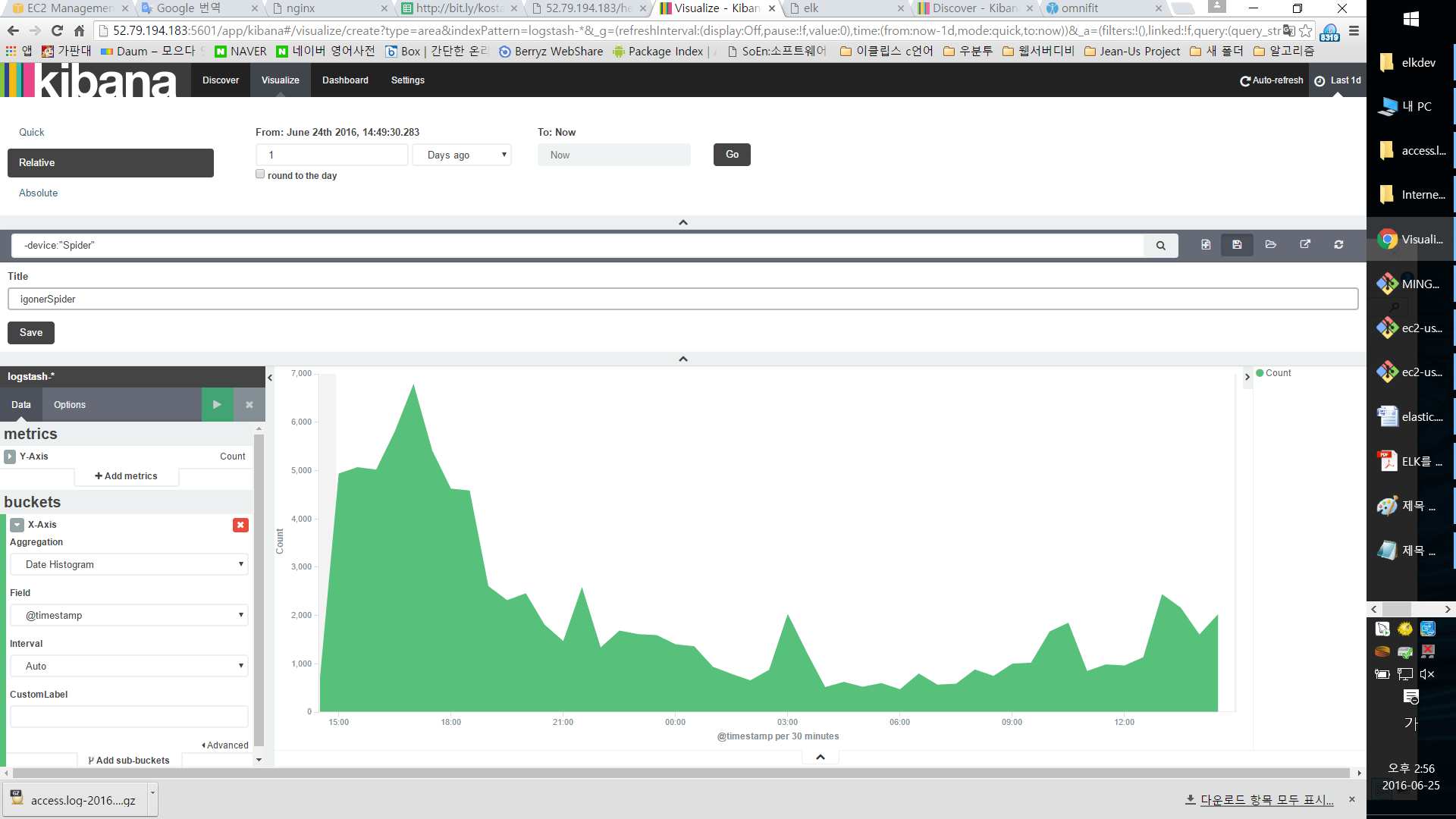
Kibana

device : “Spider”

하면은 검색봇에 대한 응답값들..

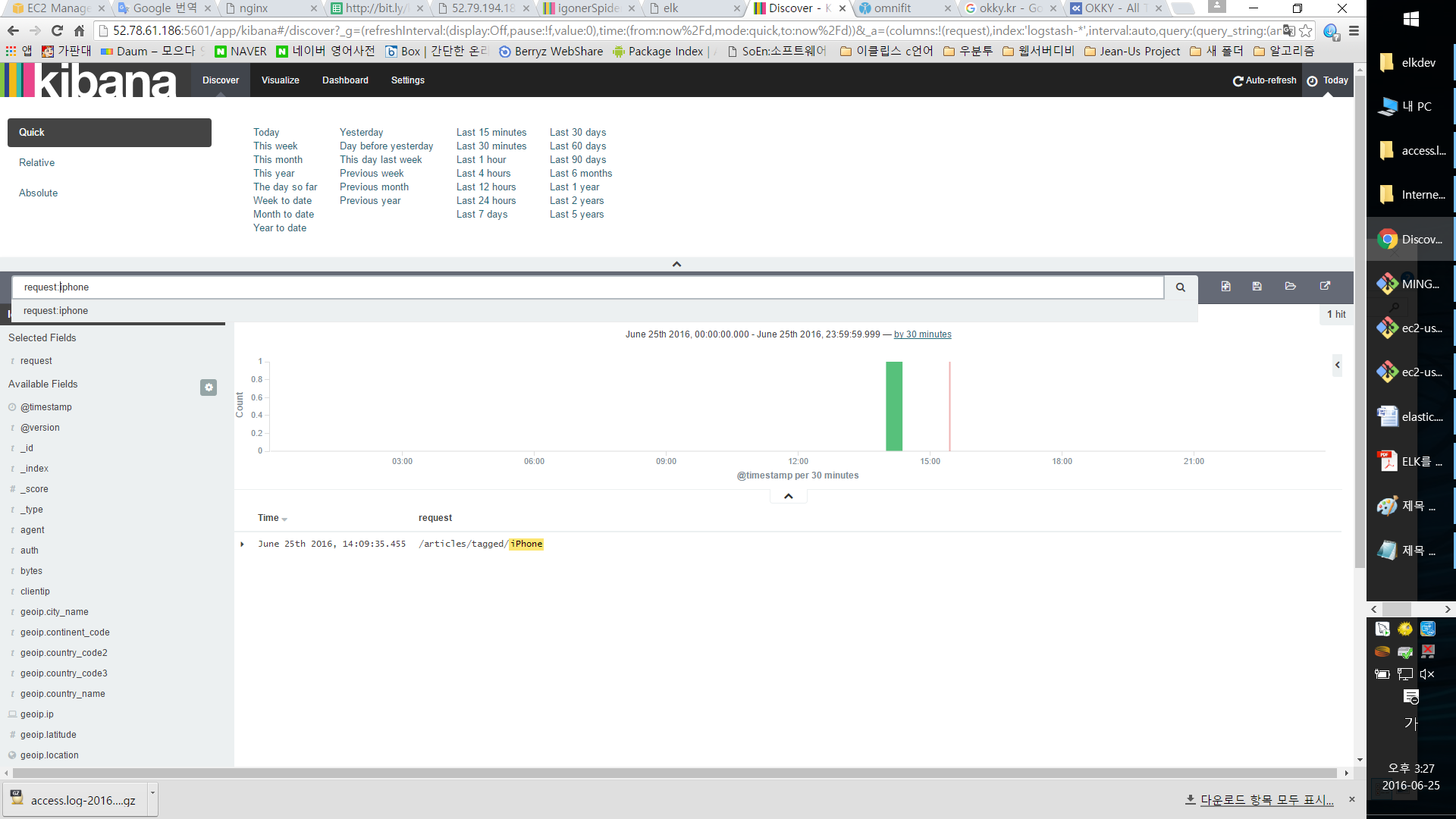
-device : “Spider”

앞부분에 –를 붙이면 제외다.

그래프 생성 및 저장

서울 지하철 승하차 데이터 elk처리하기

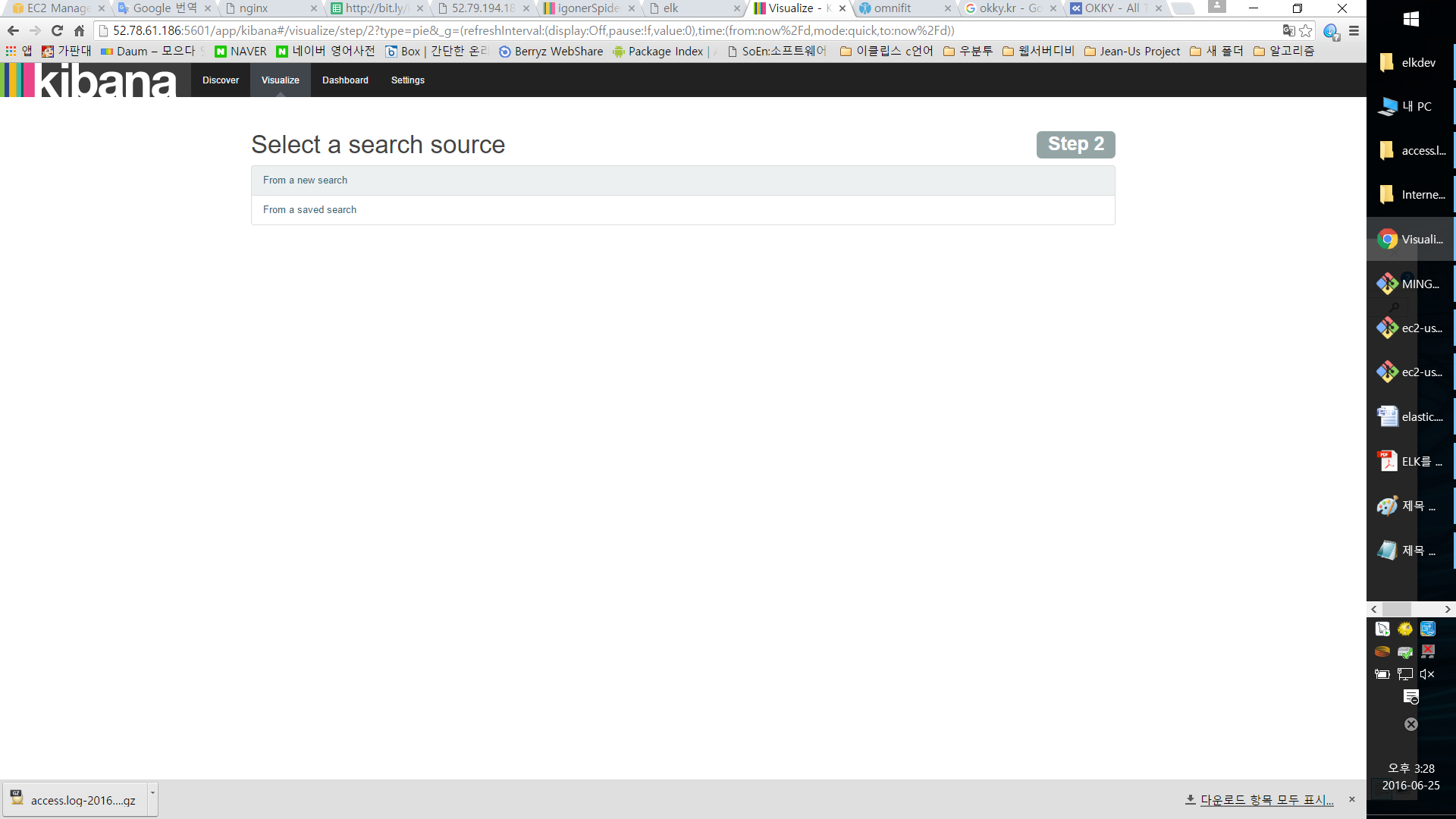
<https://www.youtube.com/watch?v=ec-XzM6_CgU&feature=youtu.be>

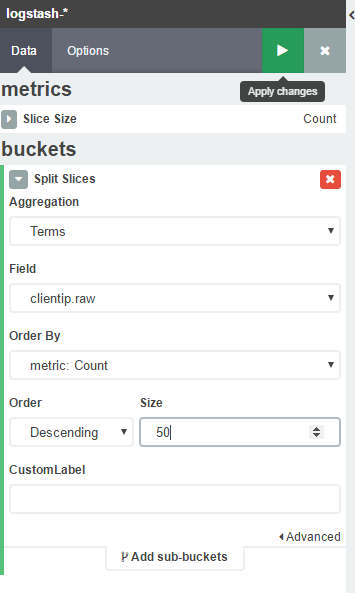
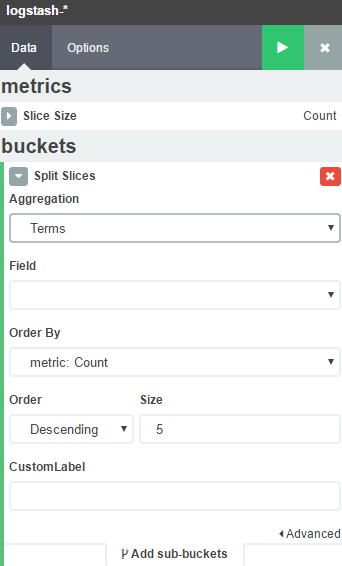


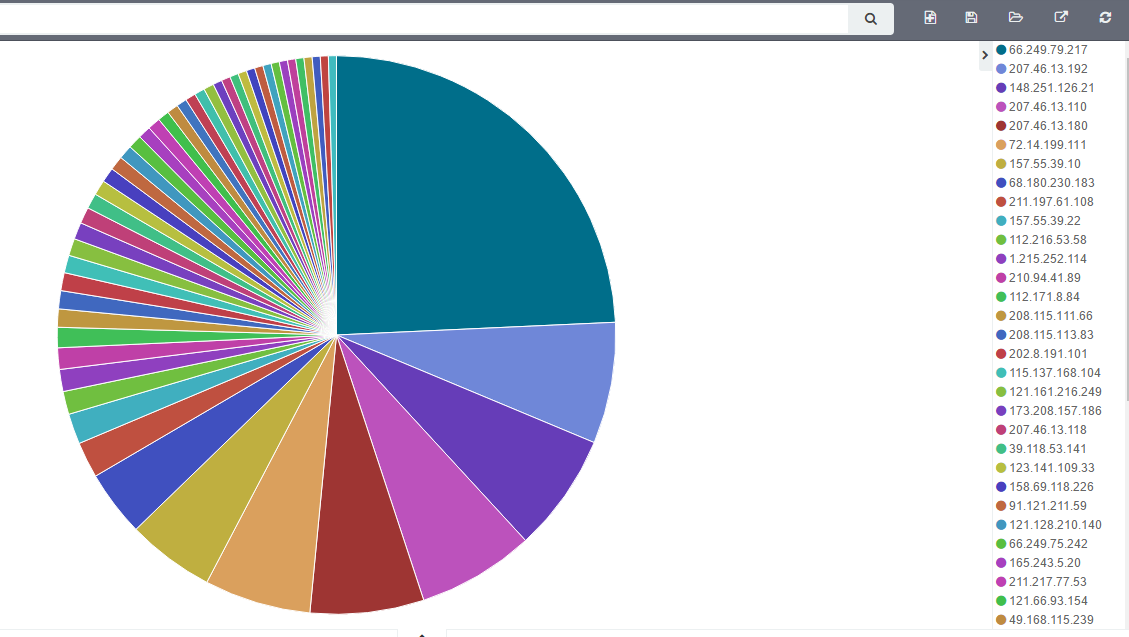
request:article

차트 만들기



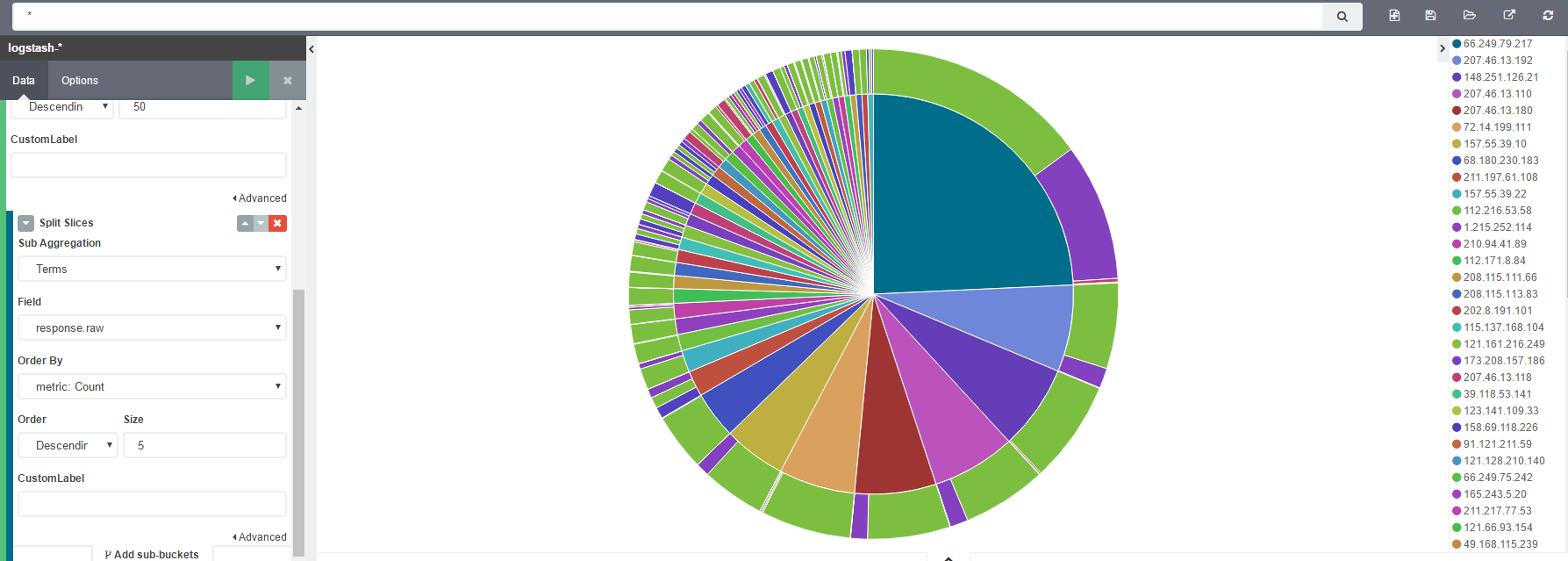






상태코드 까지 추가해볼까요~

add sub-buckets



위쪽에 질의를 넣을수도 있다.

특정 필드를 시간타임 필드로 지정

* 이미지 제거
* filter {
* if [message] =~ "^#|\.(css|js|ico|png|xml|jpg|JPG|gif|jpeg|eot\?) " {
* drop {}
* }
* }
* useragent 파싱
* useragent {
* source => "agent"
* }
* timestamp 조정(apache log)
* date {
* match => [ "timestamp", "dd/MMM/yyyy:HH:mm:ss Z" ]
* }
* <https://www.elastic.co/guide/en/logstash/current/plugins-filters-date.html>
* urldecode
* urldecode {
* field => "params"
* }

LogStash 필터 변경

|  |
| --- |
| cd ~/local/logstash/logconf/  vi nginx.conf |
| input {  file {  path => "/var/log/nginx/access\*.log"  start\_position => beginning  }  }  filter {  grok {  match => { "message" => "%{COMBINEDAPACHELOG}"}  }  geoip {  source => "clientip"  }  date {  match => [ "timestamp", "dd/MMM/yyyy:HH:mm:ss Z" ]  }  }  output {  elasticsearch {}  stdout {}  } |
| Logstash 죽이기  ps -ef| grep logstash  ec2-user 7347 2639 0 02:54 pts/0 00:00:51 /usr/lib/jvm/jre/bin/java -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -Djava.awt.headless=true -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -XX:+HeapDumpOnOutOfMemoryError -Xmx1g -Xss2048k -Djffi.boot.library.path=/home/ec2-user/local/logstash-2.3.2/vendor/jruby/lib/jni -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -Djava.awt.headless=true -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/home/ec2-user/local/logstash/heapdump.hprof -Xbootclasspath/a:/home/ec2-user/local/logstash-2.3.2/vendor/jruby/lib/jruby.jar -classpath : -Djruby.home=/home/ec2-user/local/logstash-2.3.2/vendor/jruby -Djruby.lib=/home/ec2-user/local/logstash-2.3.2/vendor/jruby/lib -Djruby.script=jruby -Djruby.shell=/bin/sh org.jruby.Main --1.9 /home/ec2-user/local/logstash/lib/bootstrap/environment.rb logstash/runner.rb agent -f logconf/nginx.conf --debug  ec2-user 29724 7841 0 05:07 pts/1 00:00:00 grep --color=auto logstash  kill -9 7347 |
|  |
| elastic search index 삭제 다시 데이터를 지우고 다시 입력한다. |
| cd ~/local/elasticsearch/data/elasticsearch/nodes/0/indices  curl -XDELETE "localhost:9200/logstash\*" |
|  |
| logstash 재시작 |
| cd ~/local/  cd logstash  nohup bin/logstash -f logconf/nginx.conf &  확인  dstat 60  안쌓인다 띠발 |
| cd /var/log/nginx  cp ./access-okky.log ./access-okky2.log  파일 이벤트 발생시켜주면…  cd ~/local/elasticsearch/data/elasticsearch/nodes/0/indices  에가서 보면  [ec2-user@ip-172-31-13-60 indices]$ ls -al  total 20  drwxrwxr-x 5 ec2-user ec2-user 4096 Jun 25 06:59 .  drwxrwxr-x 4 ec2-user ec2-user 4096 Jun 25 03:07 ..  drwxrwxr-x 4 ec2-user ec2-user 4096 Jun 25 02:46 .kibana  drwxrwxr-x 8 ec2-user ec2-user 4096 Jun 25 06:58 logstash-2016.06.22  drwxrwxr-x 8 ec2-user ec2-user 4096 Jun 25 06:59 logstash-2016.06.23  [ec2-user@ip-172-31-13-60 indices]$  점점 쌓인다.  파일 수정날짜로 체크를 하나보다 띠발. |

플러그인 관리 사용하기

|  |
| --- |
| • 기본 플러그인  • input {}  • filter {}  • output {}  • 설정 파일을 만들어서 관리  • https://www.elastic.co/guide/en/logstash/current/first-event.html  cd logstash-2.3.0  bin/logstash -e 'input { stdin { } } output { stdout {} }' |
| 입력 소스 설정  • **beats**, couchdb\_changes, drupal\_dblog, elasticsearch, exec, eventlog, **file**, ganglia, gelf,  generator, graphite, github, heartbeat, heroku, http, http\_poller, irc, imap, jdbc, jmx, kafka,  log4j, lumberjack, meetup, pipe, puppet\_facter, relp, rss, rackspace, rabbitmq, redis,  salesforce, snmptrap, **stdin**, sqlite, s3, sqs, stomp, syslog, tcp, twitter, unix, udp,  varnishlog, wmi, websocket, xmpp, zenoss, zeromq  • <https://www.elastic.co/guide/en/logstash/current/input-plugins.html> |
| 출력 형태 설정  • <https://www.elastic.co/guide/en/logstash/current/output-plugins.html> |
| 필터 설정  • <https://www.elastic.co/guide/en/logstash/current/filter-plugins.html> |
| 이벤트 표시 설정  • avro, cef, cloudtrail, cloudfront, collectd, compress\_spooler, dots, edn, edn\_lines, es\_bulk,  fluent, graphite, gzip\_lines, json, json\_lines, line, msgpack, multiline, netflow, nmap,  oldlogstashjson, plain, **rubydebug**, s3\_plain, spool  • <https://www.elastic.co/guide/en/logstash/current/codec-plugins.html> |
| • filebeat 플러그인  cd ~/local/logstash  ./bin/logstash-plugin install logstash-input-beats  **• logstash 설정**  **input {**  **beats {**  **port => 5044**  **}**  **}** |

logstash구조

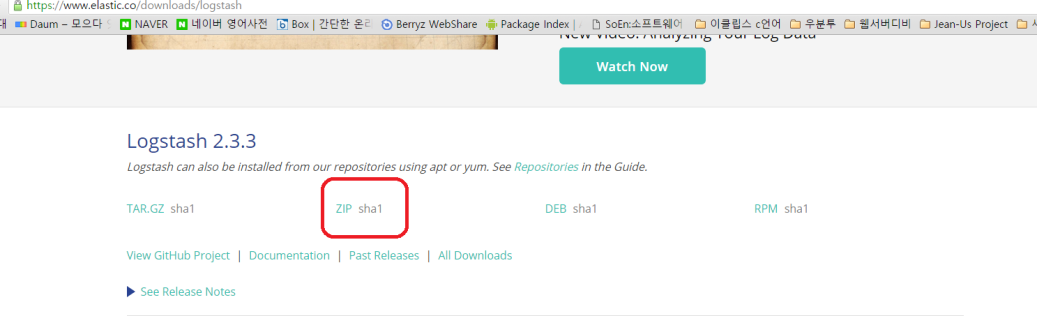
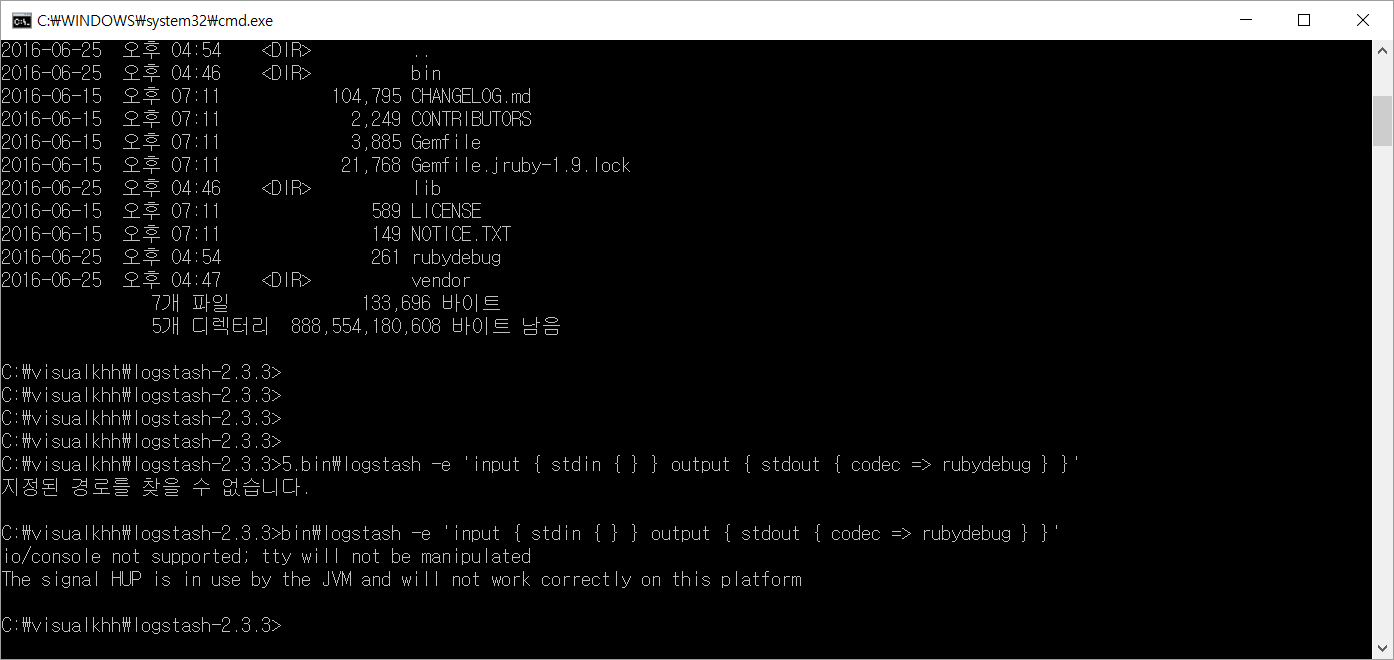
|  |  |  |
| --- | --- | --- |
| input(입력) | filter(데이터 수정 및 필터) | output(출력) |

파일 비츠 filebeat 설정 및 실행

|  |
| --- |
| • filebeat 설정  cd ~/local  wget https://download.elastic.co/beats/filebeat/filebeat-1.2.3-  x86\_64.tar.gz  tar xvfz filebeat-1.2.3-x86\_64.tar.gz  ln -s filebeat-1.2.3-x86\_64 filebeat  cd filebeat  # elasticsearch 부분 #으로 주석 처리  # elasticsearch:  #hosts: ["localhost:9200"]  # logstash 부분 # 주석 해제  logstash:  hosts: ["localhost:5044"]  # filebeat.yml 내용 중 로그 위치 변경 `/var/log/nginx/\*.log`  •실행  ./filebeat -e -c filebeat.yml  또는  nohub ….. & |

**로컬에 logstash 설치하여**

**서버측 elasticsearch 쪽에 데이터 전송하기**

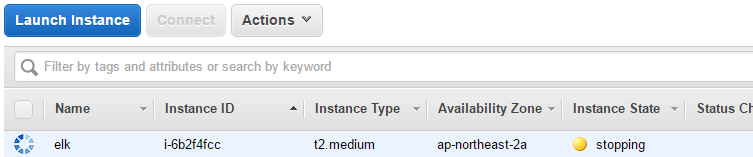
1. 로컬에 자바 1.8설치하기
   1. wget url또는 curl -O url…
   2. curl -O https://download.elastic.co/logstash/logstash/logstash-2.3.3.zip
2. <https://www.elastic.co/downloads/logstash>
   1. 
3. unzip logstash-2.3.3.zip
4. ~/logstash-2.3.3/bin/logstash -e 'input { stdin { } } output { stdout { } }'
   1. for window 🡪 bin/logstash -e 'input { stdio { } } output { stdout { } }'
5. bin\logstash -e 'input { stdin { } } output { stdout { codec => rubydebug } }'
   1. 오류 ㅠ\_ㅠ 
6. 위처럼 오류나서 파일로 직접 저정후 호출하여 해결 (window -> cmd) 에서실행

|  |
| --- |
| test.conf 파일 내용 |
| input{  stdin{}  }  output{  stdout{  codec => "rubydebug"  }  } |
| C:\visualkhh\logstash-2.3.3>bin\logstash -f test.conf  io/console not supported; tty will not be manipulated  Settings: Default pipeline workers: 4  Pipeline main started  {  "message" => "\r",  "@version" => "1",  "@timestamp" => "2016-06-25T08:02:12.037Z",  "host" => "MSDN-SPECIAL"  }  {  "message" => "\r",  "@version" => "1",  "@timestamp" => "2016-06-25T08:02:12.196Z",  "host" => "MSDN-SPECIAL"  } |

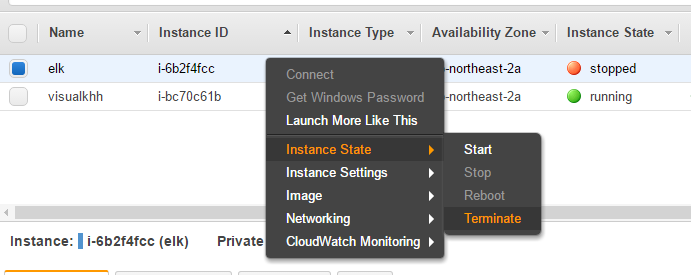
설정하자

|  |
| --- |
| mkdir logconf  notepad nginx.conf  input {  stdin{}  }  filter {  grok {  match => { "message" => "%{COMBINEDAPACHELOG}"}  }  geoip {  source => "clientip"  }  }  output {  stdout {  codec => "rubydebug"  }  } |
|  |
| 66.249.79.217 - - [25/Jun/2016:17:33:44 +0900] "GET /articles/community?query=250&sort=id&order=desc HTTP/1.1" 200 7341 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" "-"  붙쳐넣으면 기존거를 가지고 파싱을 한다 하지만 타임스템프가…틀리기 때문에 @ 가 안붙어서. 안되기떄문에  필터쪽에 걸어서 다시 해보자 |
| input {  stdin {}  }  filter {  grok {  match => { "message" => "%{COMBINEDAPACHELOG}"}  }  geoip {  source => "clientip"  }  date {  match => [ "timestamp", "dd/MMM/yyyy:HH:mm:ss Z" ]  }  }  output {  stdout {  codec => "rubydebug"  }  } |
| 66.249.79.217 - - [25/Jun/2016:17:33:44 +0900] "GET /articles/community?query=250&sort=id&order=desc HTTP/1.1" 200 7341 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" "-"  {  "message" => "66.249.79.217 - - [25/Jun/2016:17:33:44 +0900] \"GET /articles/community?query=250&sort=id&order=desc HTTP/1.1\" 200 7341 \"-\" \"Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)\" \"-\"\r",  "@version" => "1",  "@timestamp" => "2016-06-25T08:33:44.000Z",  "host" => "MSDN-SPECIAL",  "clientip" => "66.249.79.217",  "ident" => "-",  "auth" => "-",  "timestamp" => "25/Jun/2016:17:33:44 +0900",  "verb" => "GET",  "request" => "/articles/community?query=250&sort=id&order=desc",  "httpversion" => "1.1",  "response" => "200",  "bytes" => "7341",  "referrer" => "\"-\"",  "agent" => "\"Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)\"",  "geoip" => {  "ip" => "66.249.79.217",  "country\_code2" => "US",  "country\_code3" => "USA",  "country\_name" => "United States",  "continent\_code" => "NA",  "region\_name" => "CA",  "city\_name" => "Mountain View",  "postal\_code" => "94043",  "latitude" => 37.41919999999999,  "longitude" => -122.0574,  "dma\_code" => 807,  "area\_code" => 650,  "timezone" => "America/Los\_Angeles",  "real\_region\_name" => "California",  "location" => [  [0] -122.0574,  [1] 37.41919999999999  ]  }  } |
|  |
| request의 쿼리를 분리하고 싶다. |
| input {  stdin {}  }  filter {  grok {  match => { "message" => "%{COMBINEDAPACHELOG}"}  }  geoip {  source => "clientip"  }  date {  match => [ "timestamp", "dd/MMM/yyyy:HH:mm:ss Z" ]  }  **mutate {**  **add\_field => {**  **"tmp" => "%{request}"**  **}**  **}**  **if [tmp] =~ "\?" {**  **mutate {**  **split => [**  **"tmp", "?"**  **]**  **add\_field => {**  **"params" => "%{[tmp][1]}"**  **}**  **}**  **kv {**  **field\_split => "&"**  **source => "params"**  **include\_keys => [ "category", "utm\_source" ]**  **prefix => "param\_"**  **}**  **}**  **}**  output {  stdout {  codec => "rubydebug"  }  } |
| 테스트 데이터  66.249.79.217 - - [25/Jun/2016:17:33:44 +0900] "GET /articles/community?query=250&sort=id&order=desc HTTP/1.1" 200 7341 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" "-"  66.249.79.217 - - [25/Jun/2016:17:33:44 +0900] "GET /articles/community?category=hello&query=250&sort=id&order=desc HTTP/1.1" 200 7341 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" "-" |
| 위결과 : 둘다 쿼리부분 따로 빠지고  첫번째는 param\_category안나옴 , 두번째꺼는 나옴 |
| 이미지 제거 처리 |
| input {  stdin {}  }  filter {  if [message] =~ "^#|\.(css|js|ico|png|xml|jpg|JPG|gif|jpeg|eot\?) " {  drop {}  }  grok {  match => { "message" => "%{COMBINEDAPACHELOG}"}  }  …  output {  stdout {  codec => "rubydebug"  }  } |
| 테스트 데이터  66.249.79.217 - - [25/Jun/2016:17:33:44 +0900] "GET /images/hello.gif HTTP/1.1" 200 7341 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" "-" |
| 결과  안나온다 ㅋㅋ |
|  |
| useragent 필터 처리 |
| input {  stdin {}  }  filter {  …  useragent {  source => "agent"  }  ….  output {  stdout {  codec => "rubydebug"  }  } |
| 테스트 데이터  116.39.189.14 - - [25/Jun/2016:17:53:50 +0900] "GET /article/333290 HTTP/1.1" 200 14015 "http://okky.kr/" "Mozilla/5.0 (Linux; Android 4.4.2; ko-kr; IM-A870K/2.13 Build/KOT49H) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36" "-" |

aws instance삭제



terminated



## Kibana 통계

### 시각화(Visualize)

* Terms(request.raw, clientip.raw, ...) 또는 Filters(request: "/hello.html", ...) 이용해서 차트 생성
* 테이블, 라인차트, 파이차트, 지도 등 가능
* 만들어진 차트는 저장 가능

### 대시보드 만들기

* 저장된 차트를 한 화면에서 볼 수 있도록 추가, 레이아웃 가능

## part 2

### Logstash

* 필드 추가
* field{
* mutate {
* add\_field => {
* "reqs" => "%{request}"
* }
* }
* }
* 분리
* field{
* mutate {
* split => ["reqs", "?"]
* add\_field => { "uri" => "%{reqs[0]}" }
* add\_field => { "req\_uri" => "%{reqs[0]}" }
* # add\_field => { "querystring" => "%{reqs[1]}" }
* }
* }
* 필드 제거
* mutate {
* remove\_field => [
* "reqs",
* "uri"
* ]
* }
* 파라미터 필드 만들기
* filter {
* mutate {
* add\_field => {
* "tmp" => "%{request}"
* }
* }
* if [tmp] =~ "\?" {
* mutate {
* split => [
* "tmp", "?"
* ]
* add\_field => {
* "params" => "%{[tmp][1]}"
* }
* }
* kv {
* field\_split => "&"
* source => "params"
* include\_keys => [ "category", "utm\_source" ]
* prefix => "param\_"
* }
* }
* }
* 또는

# params

if [request] =~ "\?" {

kv {

field\_split => "&"

source => "querystring"

include\_keys => [ "query", "redirectUrl" ]

prefix => "param\_"

}

}

* 이미지 제거
* filter {
* if [message] =~ "^#|\.(css|js|ico|png|xml|jpg|JPG|gif|jpeg|eot\?) " {
* drop {}
* }
* }
* useragent 파싱
* useragent {
* source => "agent"
* }
* timestamp 조정(apache log)
* date {
* match => [ "timestamp", "dd/MMM/yyyy:HH:mm:ss Z" ]
* }
* <https://www.elastic.co/guide/en/logstash/current/plugins-filters-date.html>
* urldecode
* urldecode {
* field => "params"
* }

### Kibana

* 질의어 문법(query syntax)
  + Lucene 검색 엔진의 문법 그대로 사용(<https://lucene.apache.org/core/2_9_4/queryparsersyntax.html>)
* request: "uri"
* 제외 -device : "Spider"

### elasticsearch

* 데이터 지우기
  + curl -XDELETE http://localhost:9200/logstash\*

## Filebeat with logstash

* (Optional)
* logstash forwarder(deprecated) 의 경량(lightweight) 버전
* logstash plugin 설치
* cd ~/local/logstash
* ./bin/logstash-plugin install logstash-input-beats
* filebeat 설치

cd ~/local

wget https://download.elastic.co/beats/filebeat/filebeat-1.2.3-x86\_64.tar.gz

tar xvfz filebeat-1.2.3-x86\_64.tar.gz

ln -s filebeat-1.2.3-x86\_64 filebeat

cd filebeat

# elasticsearch 부분 #으로 주석 처리

# elasticsearch:

#hosts: ["localhost:9200"]

# logstash 부분 # 주석 해제

logstash:

hosts: ["localhost:5044"]

# filebeat.yml 내용 중 로그 위치 변경 `/var/log/nginx/\*.log`

* logconf/nginx.conf 파일 변경

input {

beats {

port => 5044

}

}

filter {

grok {

match => [

"message", "%{COMBINEDAPACHELOG}",

"message", "%{COMMONAPACHELOG}"

]

}

geoip {

source => "clientip"

}

}

output {

elasticsearch {

hosts => "localhost:9200"

manage\_template => false

index => "%{[@metadata][beat]}-%{+YYYY.MM.dd}"

document\_type => "%{[@metadata][type]}"

}

}

./filebeat -e -c filebeat.yml

* start shell

echo "nohup ./filebeat -e -c filebeat.yml &" > start.sh

chmod +x start.sh

./start.sh

## ELK with PM2

* 2G짜리 메모리의 인스턴스에서 ELK를 돌리면 OutOfMemory 때문에 종종 Elasticsearch 또는 Kibana가 죽습니다.
* 고육지책으로 Kibana는 node 기반이기 때문에 pm2로 Kibana가 죽으면 자동으로 살리는 방법입니다.
* download from [http://nodejs.org](http://nodejs.org/) and install node.js
* npm install -g pm2
* cd ~/local/kibana
* pm2 start bin/cli
* check kibana status with pm2 list
* pm2 logs path is placed in ~/.pm2/logs

## kibana 인증 with nginx

sudo vi /etc/nginx/nginx.conf

* server\_name: 아래 kibana 프록시 설정
* auth\_basic "Restricted Access";
* auth\_basic\_user\_file /etc/nginx/htpasswd.users;
* location / {
* proxy\_pass http://localhost:5601;
* proxy\_http\_version 1.1;
* proxy\_set\_header Upgrade $http\_upgrade;
* proxy\_set\_header Connection 'upgrade';
* proxy\_set\_header Host $host;
* proxy\_cache\_bypass $http\_upgrade;
* }
* nginx 재시작
  + sudo service nginx restart
* 5601 포트는 막고 80으로만 접속

## 참고

* Logstash grok patterns
  + <https://github.com/logstash-plugins/logstash-patterns-core/blob/master/patterns/grok-patterns>
* ELKR (ElasticSearch + Logstash + Kibana + Redis) 를 이용한 로그분석 환경 구축하기
  + <http://brantiffy.axisj.com/archives/418>
* 2016 ELK 스택으로 서울시 지하철 대시보드 만들기 추천
  + <https://youtu.be/xPjNtd8xUZo>
* EMOCON 2015 F/W ELK 스택을 사용한 서울시 지하철 대시보드 만들기
  + <https://youtu.be/ec-XzM6_CgU>
* ELK 구축하기 1 – LOGSTASH
  + [http://linux.systemv.pe.kr/elk-구축하기-1-logstash/](http://linux.systemv.pe.kr/elk-%EA%B5%AC%EC%B6%95%ED%95%98%EA%B8%B0-1-logstash/)
* [Ubuntu] ELK 설치 및 테스트 하기
  + <http://digndig.kr/ubuntu/449/>
* Splunk 대체 Solution으로서의 ELK Stack
  + <http://blog.embian.com/18>
* How To Install Elasticsearch, Logstash, and Kibana 4 on Ubuntu 14.04
  + <https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-4-on-ubuntu-14-04>
* ELK 프로그래밍 방송 영상
  + <http://bit.ly/okdevtv-elk>
* Logstash Configuration
  + <https://www.elastic.co/guide/en/logstash/current/event-dependent-configuration.html>
* Elasticsearch(Lucene) Query Syntax
  + <https://lucene.apache.org/core/2_9_4/queryparsersyntax.html>
* ELK Kibana 사용법
  + <https://dl.dropboxusercontent.com/u/2385737/Kibana-basic.pdf>

---------------------

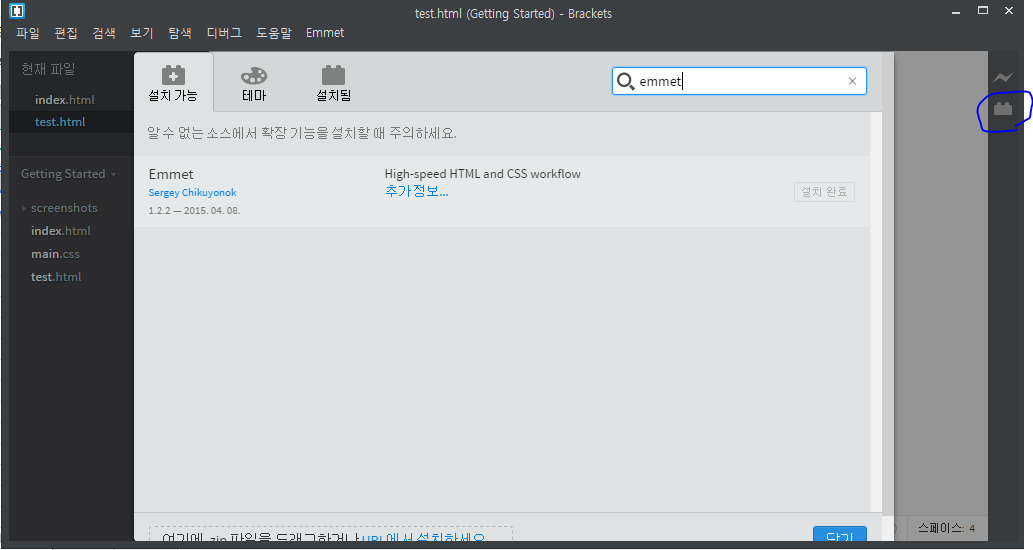
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| <https://www.dropbox.com/s/v6emkuos0792fs3/ELK%EB%A5%BC%20%EC%9D%B4%EC%9A%A9%ED%95%9C%20%EC%8B%A4%EC%8B%9C%EA%B0%84%20%EB%8D%B0%EC%9D%B4%ED%84%B0%20%EC%88%98%EC%A7%91%2C%20%EB%B6%84%EC%84%9D%2C%20%EC%8B%9C%EA%B0%81%ED%99%94%20%EA%B5%90%EC%9E%AC_20160625_V1.pdf?dl=0a> | 교재 |  |  |  |  |
| **필요 소프트웨어** |  |  |  |  |  |
| jdk 1.8 | [java.sun.com](http://java.sun.com) | java -version |  |  |  |
| git bash | [git-scm.com](http://git-scm.com) |  |  |  |  |
| elasticsearch 2.3.x | [elastic.co](http://elastic.co) |  |  |  |  |
| kibana 4.x | [elastic.co](http://elastic.co) |  |  |  |  |
| logstash 2.3.x | [elastic.co](http://elastic.co) |  |  |  |  |
| filebeat | [elastic.co](http://elastic.co) |  |  |  |  |
|  |  |  |  |  |  |
| aws console | [aws.amazon.com](http://aws.amazon.com) |  |  |  |  |
|  |  |  |  |  |  |
| <http://52.79.194.183:5601/> | okky.kr elk |  |  |  |  |
|  |  |  |  |  |  |
| [aws.amazon.com](http://aws.amazon.com) | hbenicloud@gmail.com | hbpass12 |  |  |  |
|  |  |  |  |  |  |
| [okdevtv.com/mib/elk/elk](http://okdevtv.com/mib/elk/elk) |  |  |  |  |  |
| mkdir ~/elkdev |  |  |  |  |  |
| cd ~/elkdev |  |  |  |  |  |
| pwd |  |  |  |  |  |
| explorer . |  |  |  |  |  |
| ssh -i ~/elkdev/kosta-kenu.pem ec2-user@ip |  |  |  |  |  |
| chmod 400 kosta-kenu.pem |  |  |  |  |  |
|  |  |  |  |  |  |
| aws |  |  |  |  |  |
| mkdir ~/local |  |  |  |  |  |
| cd ~/local |  |  |  |  |  |
|  |  |  |  |  |  |
| sudo yum update -y |  |  |  |  |  |
|  |  |  |  |  |  |
| [trends.google.com](http://trends.google.com) |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| <https://okdevtv.com/mib/nginx/nginx> |  |  |  |  |  |
|  |  |  |  |  |  |
| cd ~/local/elasticsearch |  |  |  |  |  |
| bin/elasticsearch -d |  |  |  |  |  |
| curl localhost:9200 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| vi config/elasticsearch.yml |  |  |  |  |  |
| /192. |  |  |  |  |  |
| yy |  |  |  |  |  |
| p |  |  |  |  |  |
| x | # 지우고 |  |  |  |  |
| network.host: 0.0.0.0 |  |  |  |  |  |
| :wq |  |  |  |  |  |
|  |  |  |  |  |  |
| ps -ef | grep elastic |  |  |  |  |  |
| kill -9 7151 |  |  |  |  |  |
|  |  |  |  |  |  |
| curl -O http://okky.kr/examples/access.log-20160624.gz |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| aws 접속 |  |  |  |  |  |
| cd ~/local |  |  |  |  |  |
| mkdir okky |  |  |  |  |  |
| cd okky |  |  |  |  |  |
| curl - Ohttp://okky.kr/examples/access.log-20160624.gz |  |  |  |  |  |
| cp access.log-20160624.gz access-okky.log.gz |  |  |  |  |  |
| gunzip access-okky.log.gz |  |  |  |  |  |
| cp access-okky.log /var/log/nginx/ |  |  |  |  |  |
| sudo yum install dstat -y | dstat 설치 |  |  |  |  |
| 데이터 적재 확인 |  |  |  |  |  |
| cd ~/local/elasticsearch |  |  |  |  |  |
| cd data/elasticsearch/nodes/0/indices/ |  |  |  |  |  |
|  |  |  |  |  |  |
| cd ~/local/logstash |  |  |  |  |  |
| vi logconf/nginx.conf |  |  |  |  |  |
| access.log | access\*.log |  |  |  |  |
| logstash 재시작 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| <https://youtu.be/ec-XzM6_CgU> | 강추 |  |  |  |  |
| <http://bit.ly/okdevtv-elk> | 초간단 설치 동영상 |  |  |  |  |
|  |  |  |  |  |  |
| :13 |  |  |  |  |  |
| date {  match => [ "timestamp", "dd/MMM/yyyy:HH:mm:ss Z" ]  } |  |  |  |  |  |
| logstash 재시작 |  |  |  |  |  |
| elasticsearch index삭제 |  |  |  |  |  |
| cd ~/local/elasticsearch/data/elasticsearch/nodes/0/indices/ |  |  |  |  |  |
| du -h |  |  |  |  |  |
|  |  |  |  |  |  |
| curl -XDELETE "localhost:9200/logstash\*" |  |  |  |  |  |
|  |  |  |  |  |  |
| cd ~/local/logstash |  |  |  |  |  |
| history | grep nohup |  |  |  |  |  |
| cd /var/log/nginx |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| <https://www.elastic.co/downloads/logstash> |  |  |  |  |  |
| curl -O https://download.elastic.co/logstash/logstash/logstash-2.3.3.zip |  |  |  |  |  |
| unzip logstash-2.3.3.zip |  |  |  |  |  |
|  |  |  |  |  |  |
| cd ~/elkdev/logstash-2.3.3 |  |  |  |  |  |
| bin/logstash -e 'input { stdin { } } output { stdout { } }' |  |  |  |  |  |
| bin\logstash -e 'input { stdio { } } output { stdout { } }' | for windows |  |  |  |  |
| bin\logstash -e 'input { stdin { } } output { stdout { } }' |  |  |  |  |  |
|  |  |  |  |  |  |
| bin\logstash -e 'input { stdin { } } output { stdout { codec => rubydebug } }' |  |  |  |  |  |
|  |  |  |  |  |  |
| The signal HUP is in use by the JVM and will not work correctly on this platform |  |  |  |  |  |
|  |  |  |  |  |  |
| notepad test.conf |  |  |  |  |  |
|  |  |  |  |  |  |
| input { stdin {}  }  output {  stdout { codec => "rubydebug" }  } |  |  |  |  |  |
| cmd 창 띄우기 |  |  |  |  |  |
| logstash-2.3.3 로 경로 이동 |  |  |  |  |  |
| mkdir logconf |  |  |  |  |  |
| cd logconf |  |  |  |  |  |
| notepad nginx.conf |  |  |  |  |  |
|  |  |  |  |  |  |
| <https://notepad-plus-plus.org/> |  |  |  |  |  |
| [brackets.io](http://brackets.io) |  |  |  |  |  |
| <https://atom.io/> |  |  |  |  |  |
| <https://code.visualstudio.com/> |  |  |  |  |  |
|  |  |  |  |  |  |
| input { stdin {} } filter { grok { match => { "message" => "%{COMBINEDAPACHELOG}"} } geoip { source => "clientip" } } output { stdout { codec <= "rubydebug" } } | input { stdin {} } filter { grok { match => { "message" => "%{COMBINEDAPACHELOG}"} } geoip { source => "clientip" } } output { stdout { codec => "rubydebug" } } |  |  |  |  |
|  |  |  |  |  |  |
| cd .. |  |  |  |  |  |
| bin\logstash -f logconf\nginx2.conf |  |  |  |  |  |
|  |  |  |  |  |  |
| 66.249.79.217 - - [25/Jun/2016:17:33:44 +0900] "GET /articles/community?query=250&sort=id&order=desc HTTP/1.1" 200 7341 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" "-" |  |  |  |  |  |
| 66.249.79.217 - - [25/Jun/2016:17:33:44 +0900] "GET /articles/community?category=hello&query=250&sort=id&order=desc HTTP/1.1" 200 7341 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" "-" | params test |  |  |  |  |
| 66.249.79.217 - - [25/Jun/2016:17:33:44 +0900] "GET /images/hello.gif HTTP/1.1" 200 7341 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" "-" | image test |  |  |  |  |
| 116.39.189.14 - - [25/Jun/2016:17:53:50 +0900] "GET /article/333290 HTTP/1.1" 200 14015 "http://okky.kr/" "Mozilla/5.0 (Linux; Android 4.4.2; ko-kr; IM-A870K/2.13 Build/KOT49H) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36" "-" |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| input { stdin {} } filter { grok { match => { "message" => "%{COMBINEDAPACHELOG}"} } geoip { source => "clientip" } date { match => [ "timestamp", "dd/MMM/yyyy:HH:mm:ss Z" ] } } output { stdout { codec => "rubydebug" } } |  |  |  |  |  |

**2회차 수업**

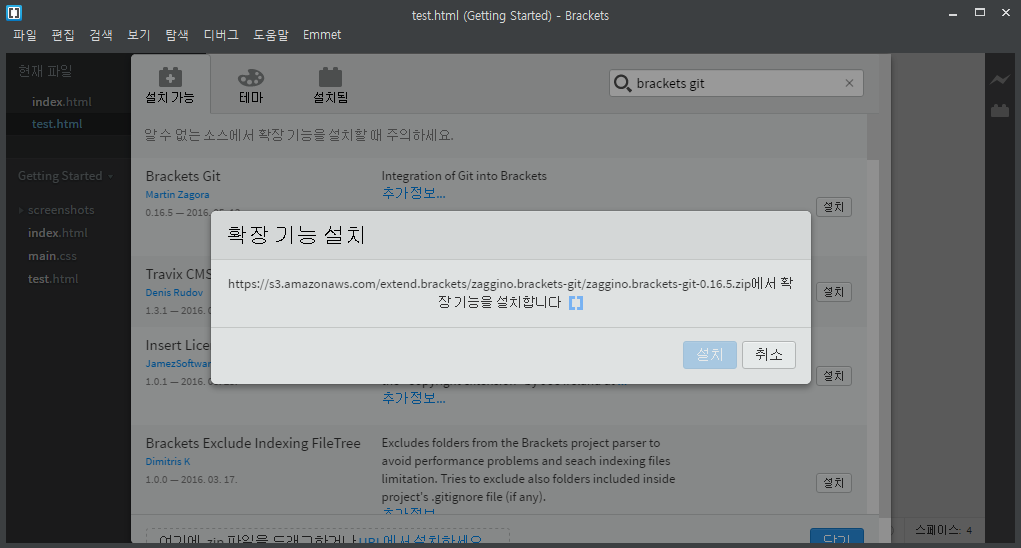
작업하기전 http://brackets.io/  추천 EDIT TOOL

플러그인이 많아 GIT / FTP 등 추가 설치 할 수 있는 EDIT TOOL 입니다.

플러그인 설치 emmet



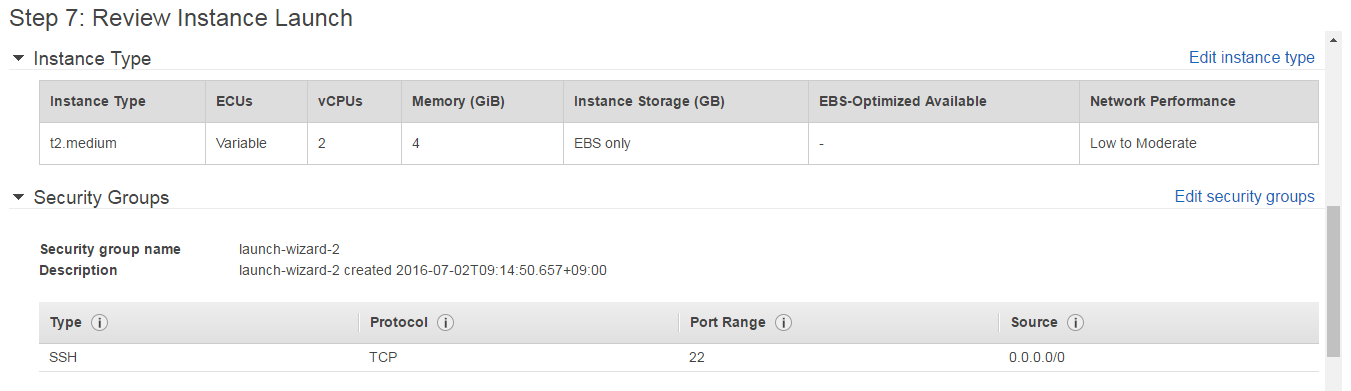
플러그인설치 brackets git

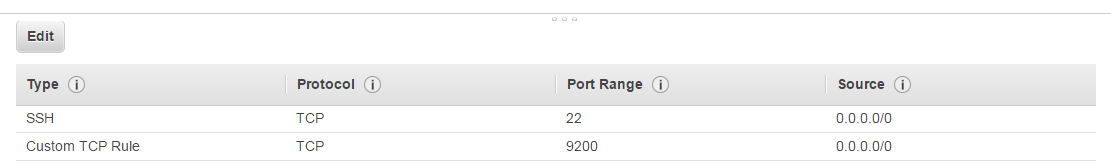


플러그인 설치 markdown preview

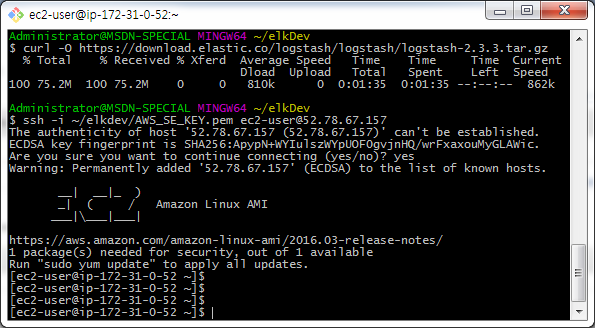


작업전 AWS Instance 다시 생성





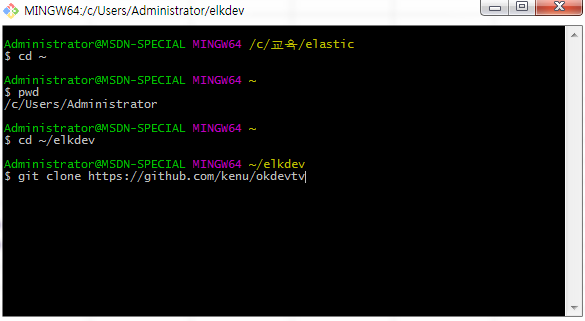
로컬에 elkdev를 생성 합니다.



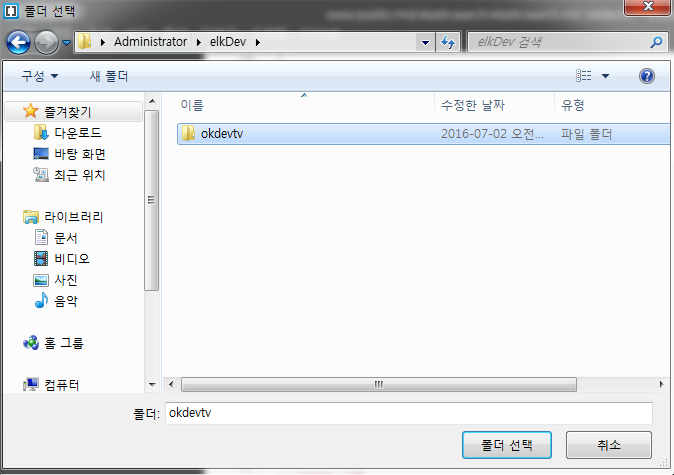
실행

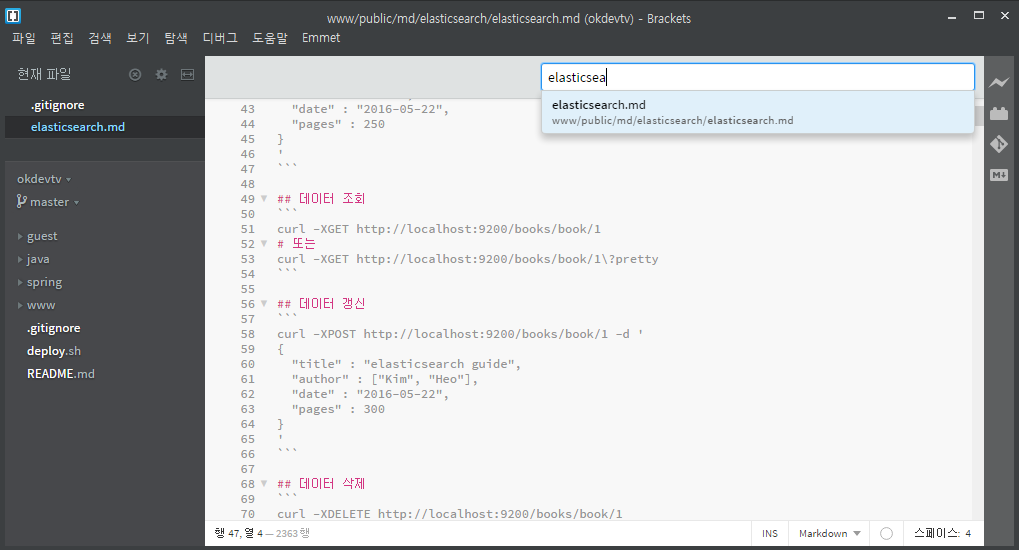
sudo yum update -y

로컬에서는 소스샘플을 다운받아 놔야 합니다.



다운로드가 완료되면 EDIT 파일 > 폴더열기 > okdevtv 선택



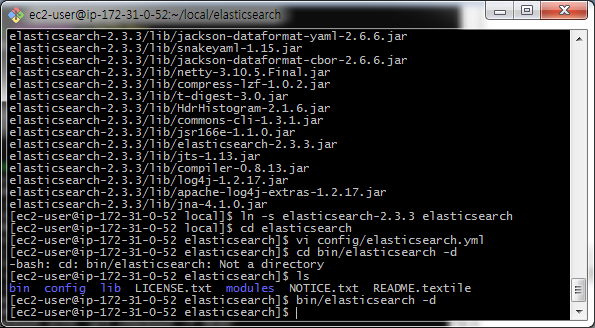


ctrl + shift + o 버튼으로 elasticsearch.md 파일을 Open 합니다.

관련자료 여기에도 있음

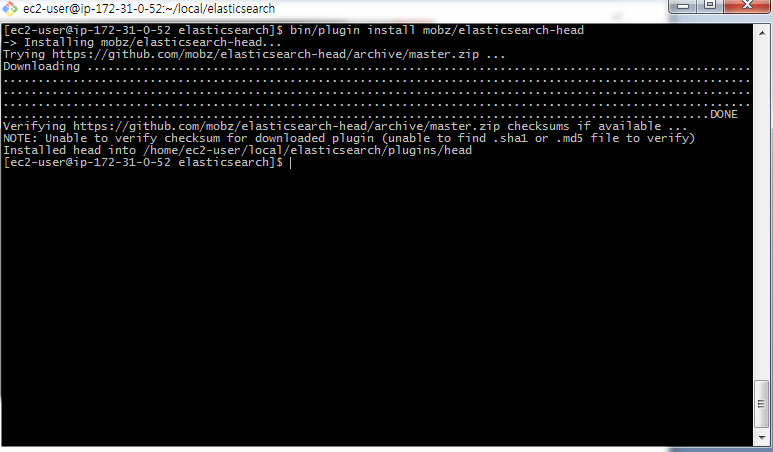
http://okdevtv.com/mib/elasticsearch/elasticsearch

**ElasticSearch**



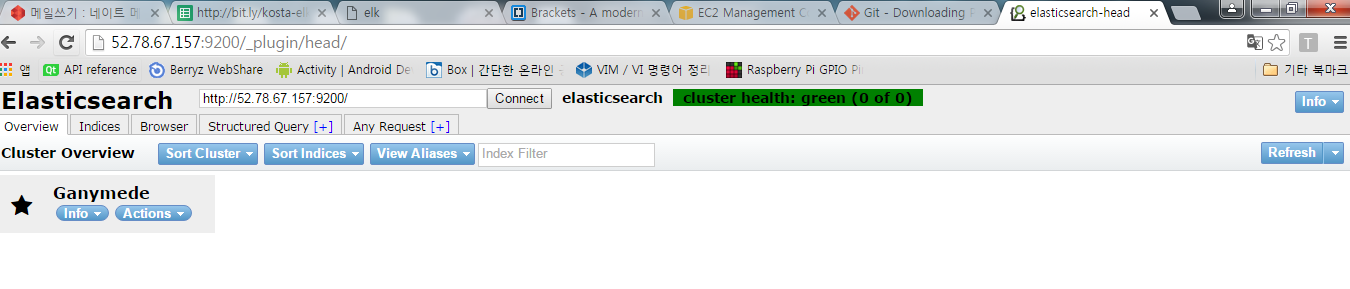
vi 에서 network.host: 0.0.0.0 으로 변경해줘야 합니다.





위의것을 설치하고 아래 주소로 접속하면 아래와 같이 나옴

ec2-user@ip-x.x.x.x] bin/plugin install mobz/elasticsearch-head



**데이터 구조 및 입출력**

* 데이터 구조

| **RDB** | **Elasticsearch** |
| --- | --- |
| Database | Index |
| Table | Type |
| Record | Document |
| Column | Field |
| Schema | Mapping |

* 데이터 입출력

| **HTTP method** | **Elasticsearch** |
| --- | --- |
| Create | POST |
| Read | GET |
| Update | PUT |
| Delete | DELETE |

**데이터 입력**

curl -XPOST http://localhost:9200/books/book/1 -d '

{

"title" : "elasticsearch guide",

"author" : "Kim",

"date" : "2016-05-22",

"pages" : 250

}

'

**데이터 조회**

curl -XGET http://localhost:9200/books/book/1

# 또는

curl -XGET http://localhost:9200/books/book/1\?pretty

**데이터 갱신**

curl -XPOST http://localhost:9200/books/book/1 -d '

{

"title" : "elasticsearch guide",

"author" : ["Kim", "Heo"],

"date" : "2016-05-22",

"pages" : 300

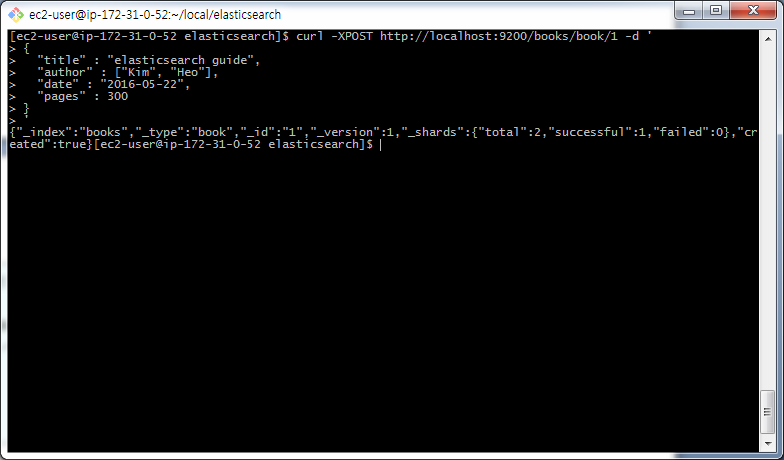
}

'

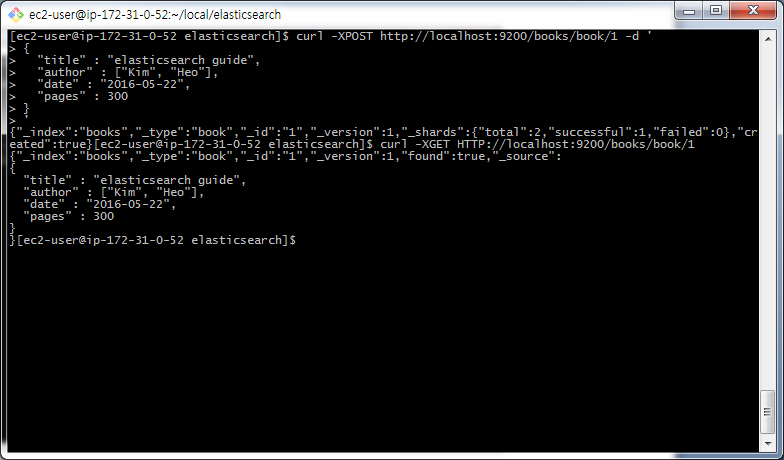
**데이터 삭제**

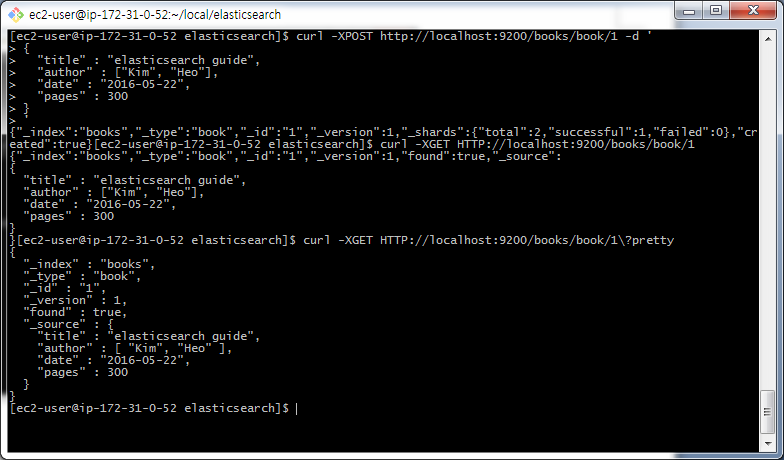
curl -XDELETE http://localhost:9200/books/book/1

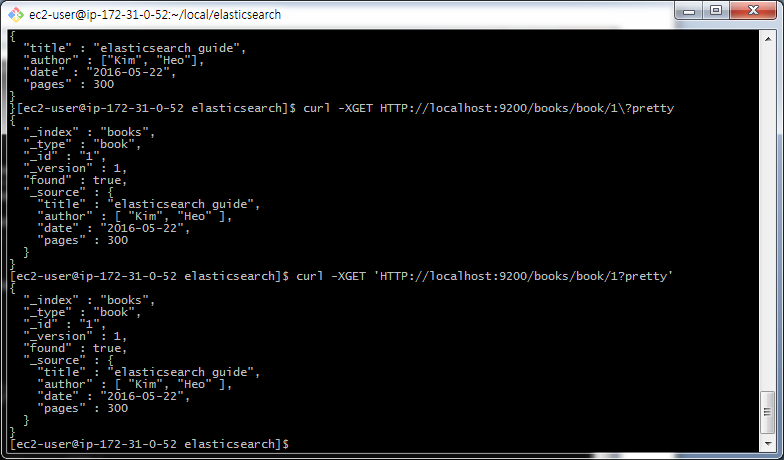
실작업



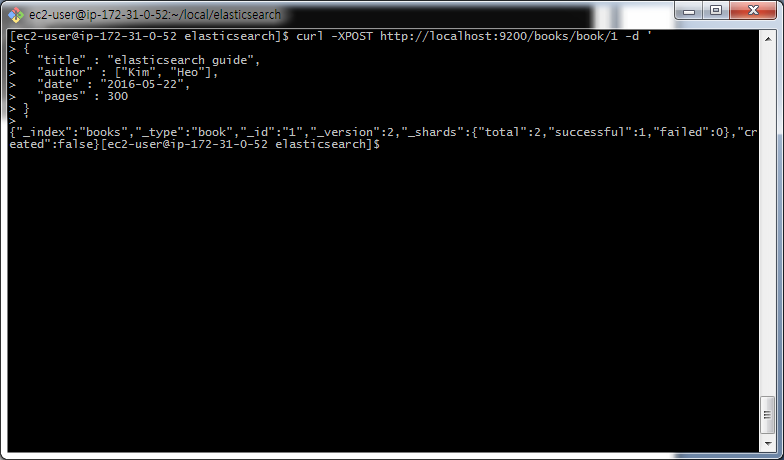
조회



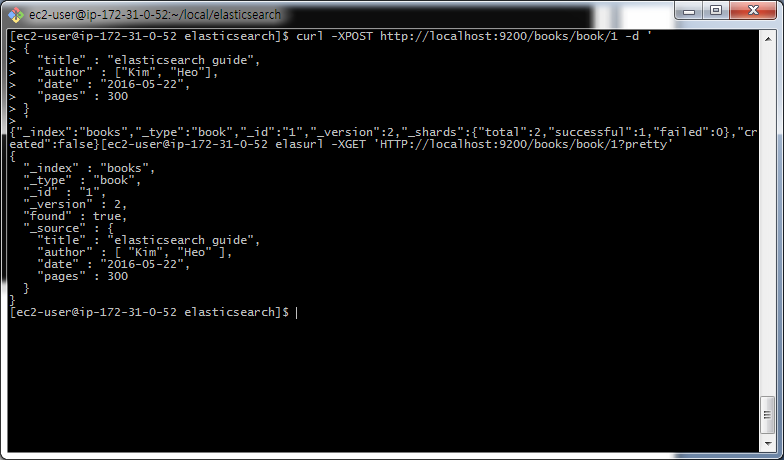




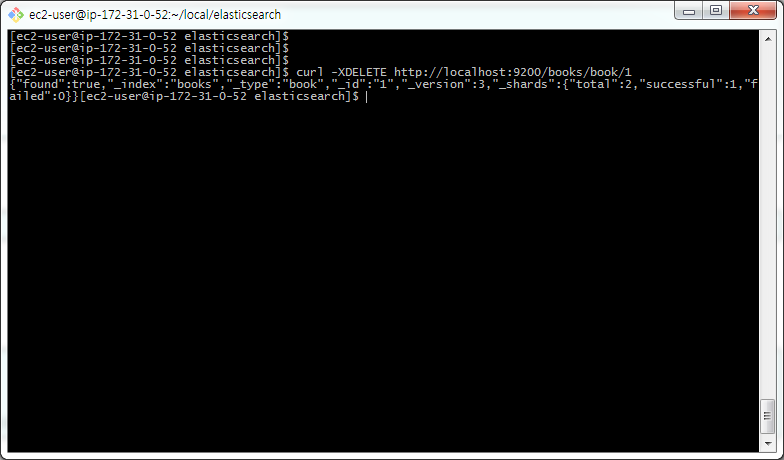
갱신



버전이 2로 올라감



삭제



데이터를 다시 넣으면 어떻게 될까?

curl -XPOST http://localhost:9200/books/book/1 -d '

{

"title" : "elasticsearch guide",

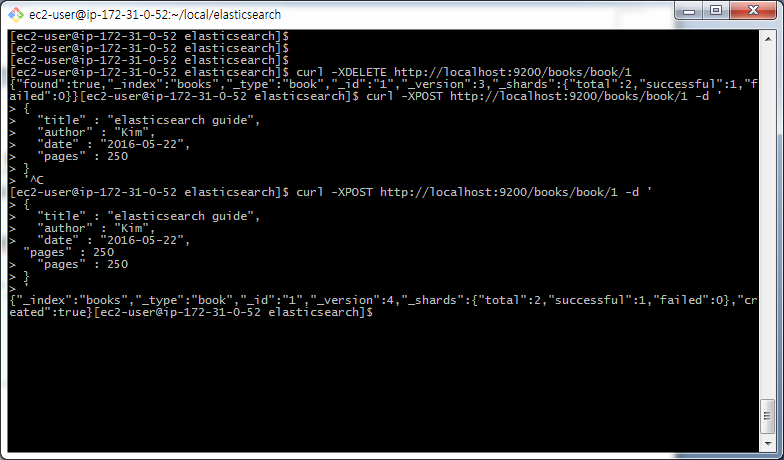
"author" : "Kim",

"date" : "2016-05-22",

"pages" : 250

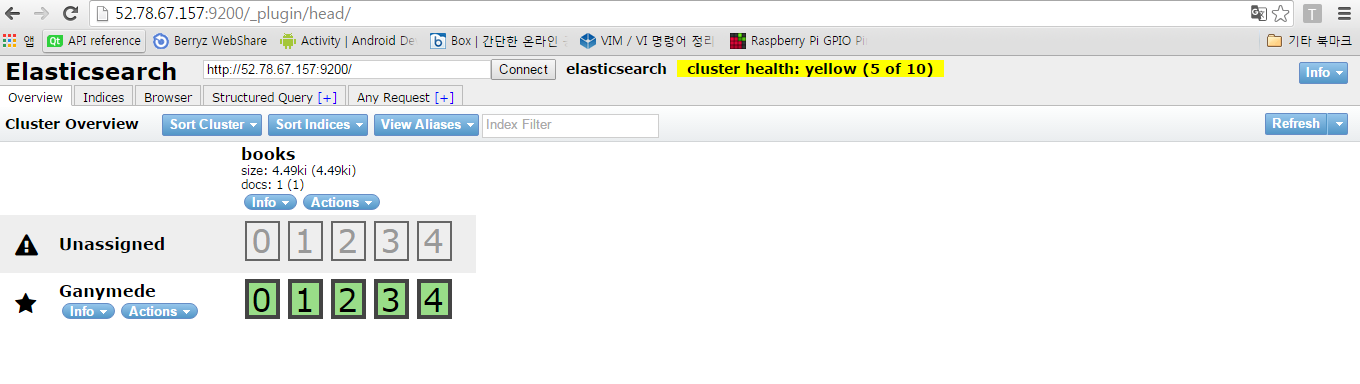
}

'



버전이 4가 되어있는것을 볼 수 있습니다.

화면으로 보기





**데이터 배치 입력(\_bulk API)**

* data.txt

{ "delete" : { "\_index" : "books", "\_type" : "book", "\_id" : "1" } }

{ "update" : { "\_index" : "books", "\_type" : "book", "\_id" : "2" } }

{ "doc" : { "date" : "2014-05-01" } }

{ "create" : { "\_index" : "books", "\_type" : "book", "\_id" : "3" } }

{ "title" : "Elasticsearch Guide II", "author" : "Park", "pages" : 400 }

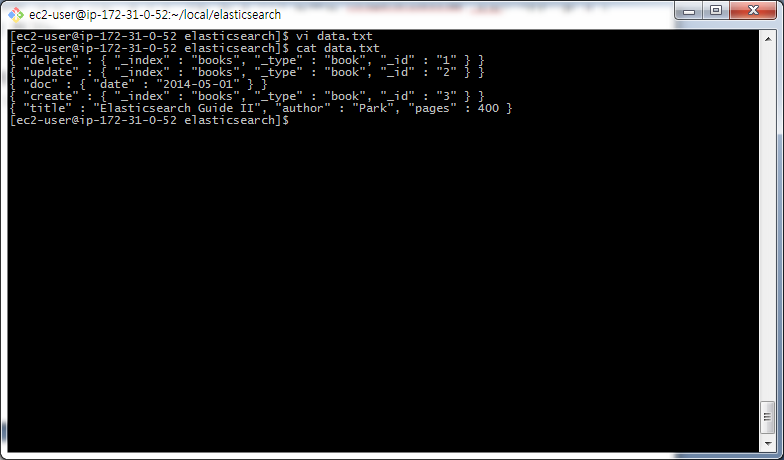
파일 입력

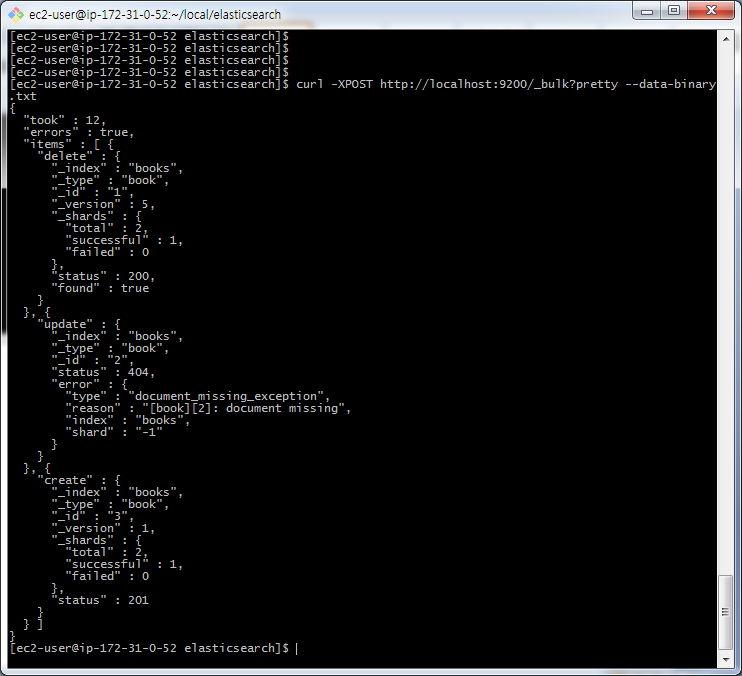
curl -XPOST http://localhost:9200/\_bulk?pretty --data-binary @data.txt

**검색**

* query방식은 2가지
  + URI방식 : REST API
  + request body방식 : http 데이터
* index/type 검색, index 검색, multi index 검색
* 시작하세요! 엘라스틱서치 예제
* downloads
  + curl -O https://codeload.github.com/wikibook/elasticsearch/zip/master
  + git clone https://github.com/wikibook/elasticsearch
  + #데이터 적재
  + cd elasticsearch/cd 05.검색
  + curl -XPOST http://localhost:9200/\_bulk --data-binary @5\_1\_books.json
  + curl -XPOST http://localhost:9200/\_bulk --data-binary @5\_2\_magazines.json

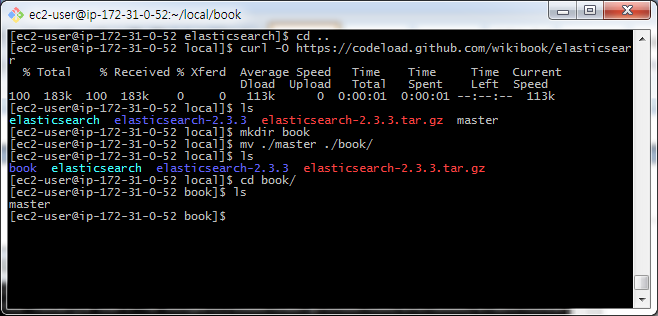
따라하기



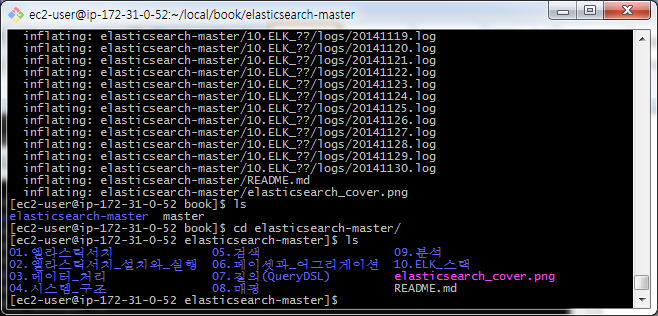


실습을 위한 다운로드

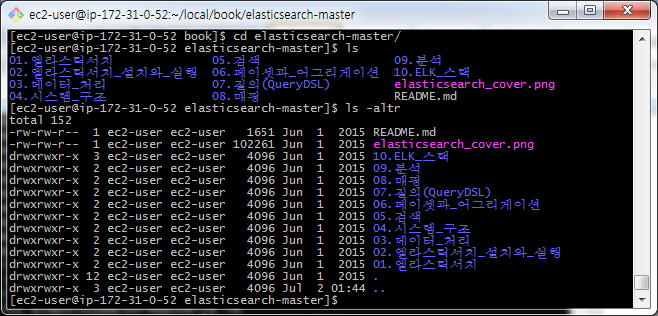


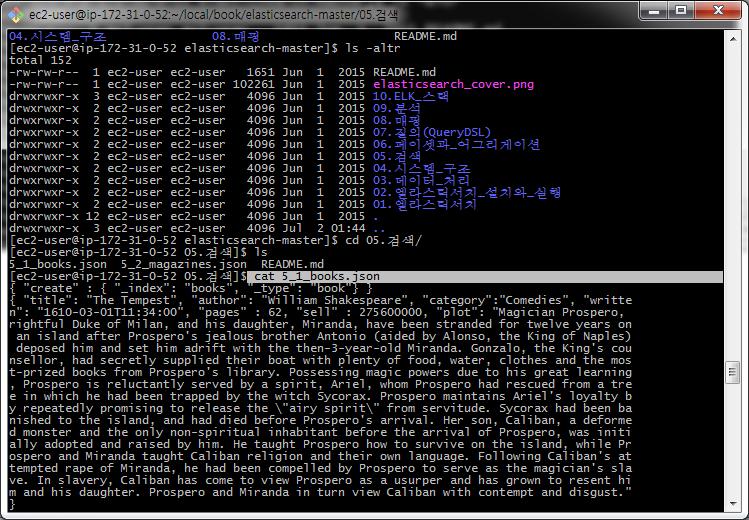




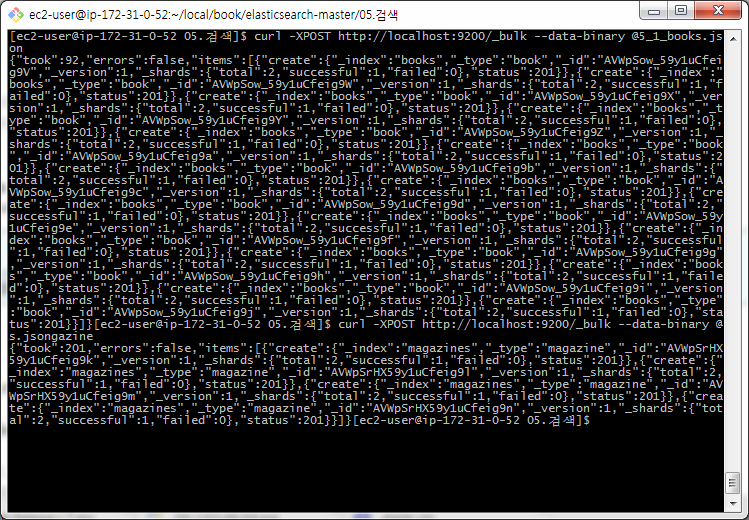


ls -altr

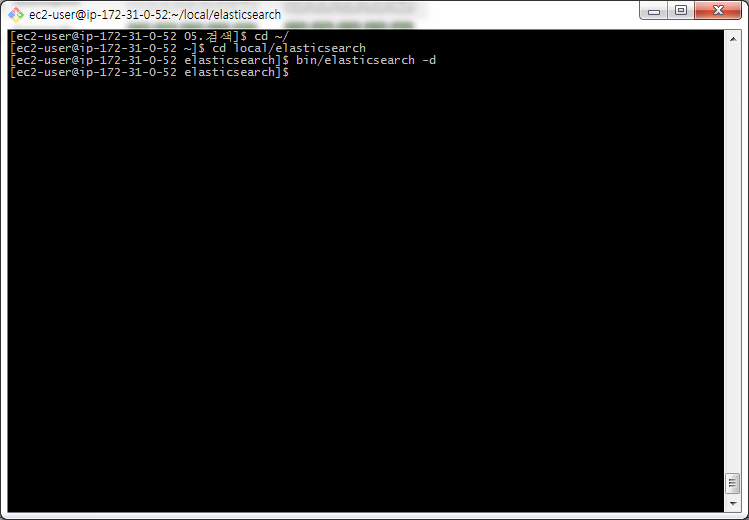




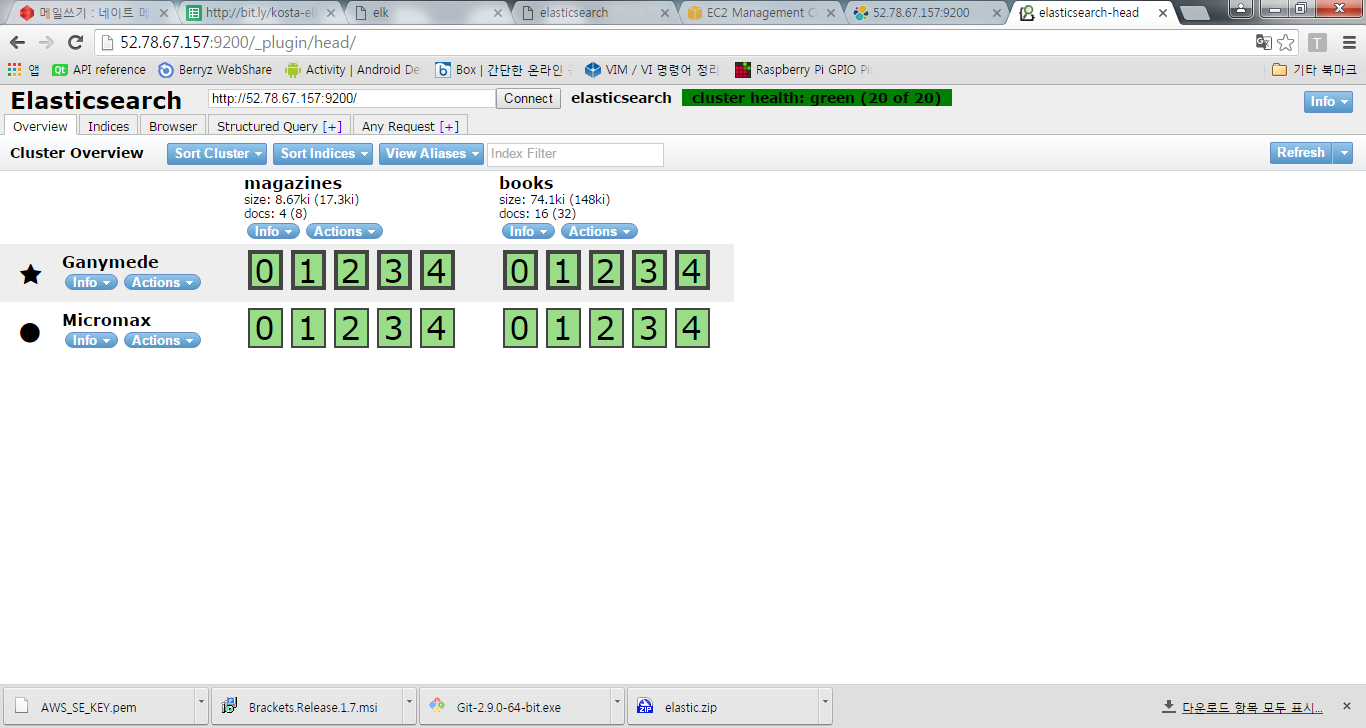
데이터 적재

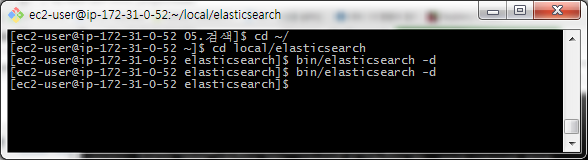




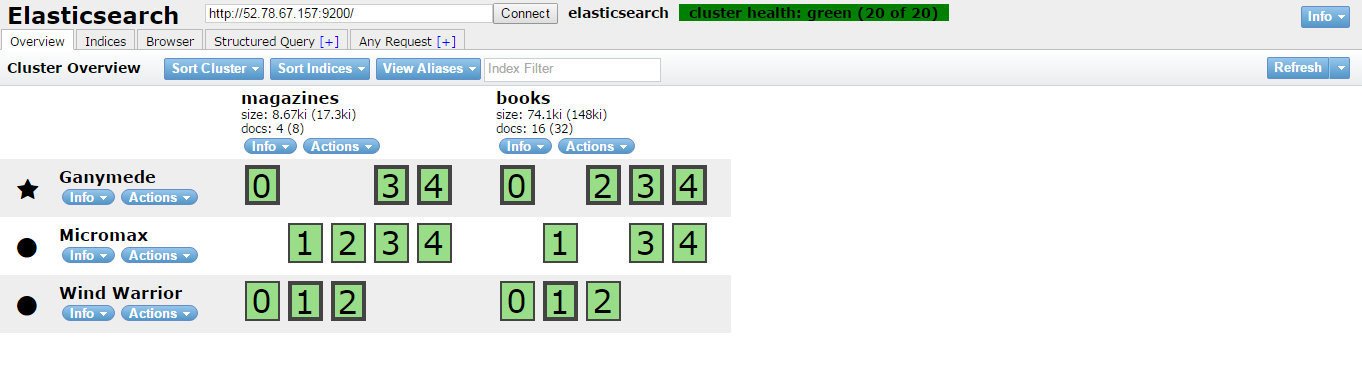


**띠운 상태에서 또 뛰우면 클러스트링처럼 또 뜹니다.**

****

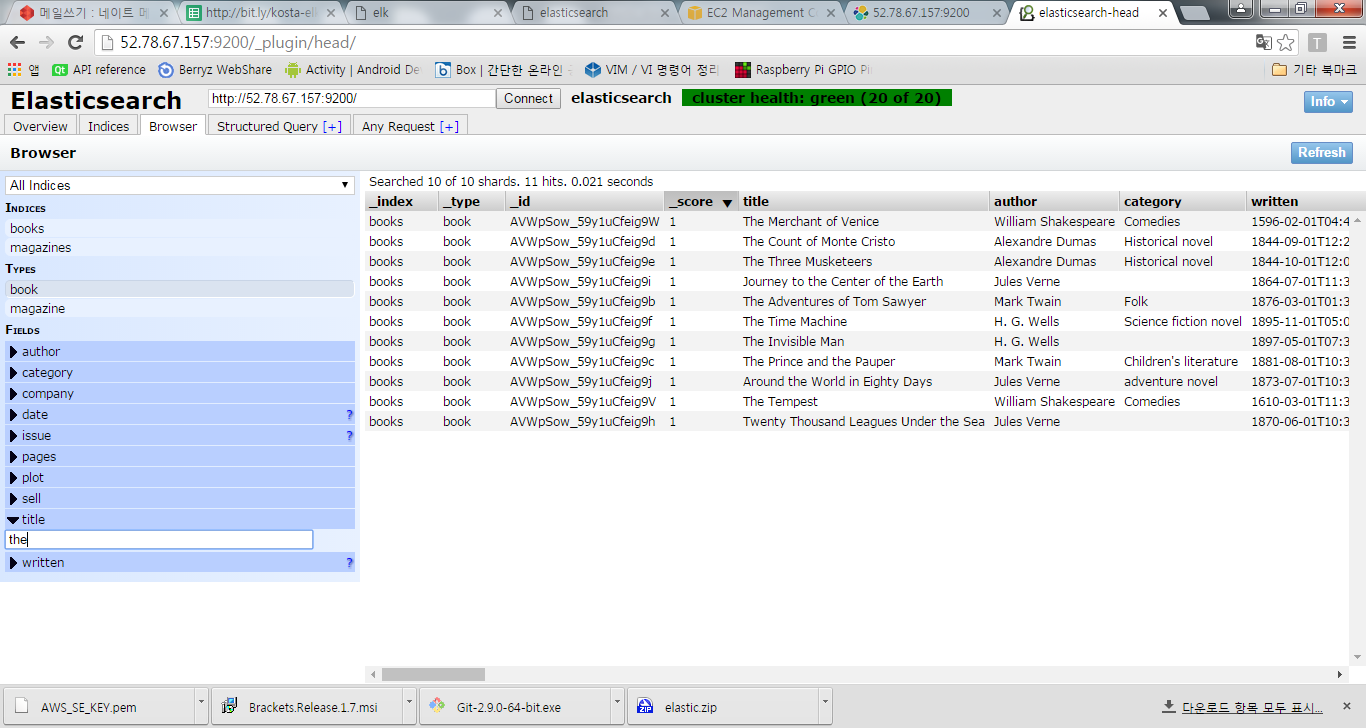
****

**한번 더 실행 ( 프로세스 3개째)**

****

**분산처리~!! 노드가 많아 지면 많아 질 수록 속도가 빨라집니다.**

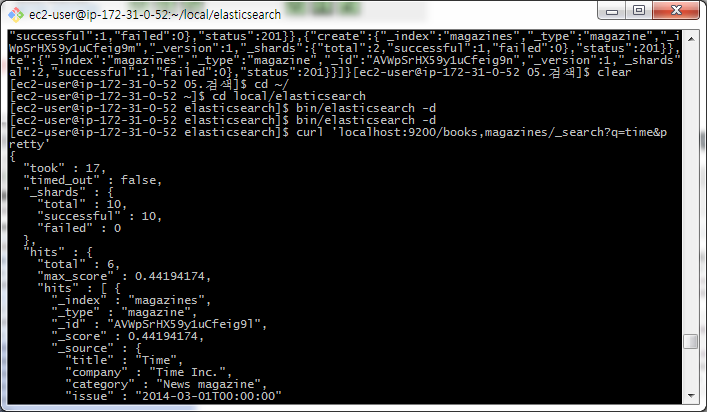
**검색을 이렇게 할 수 있습니다. TITLE로**

****

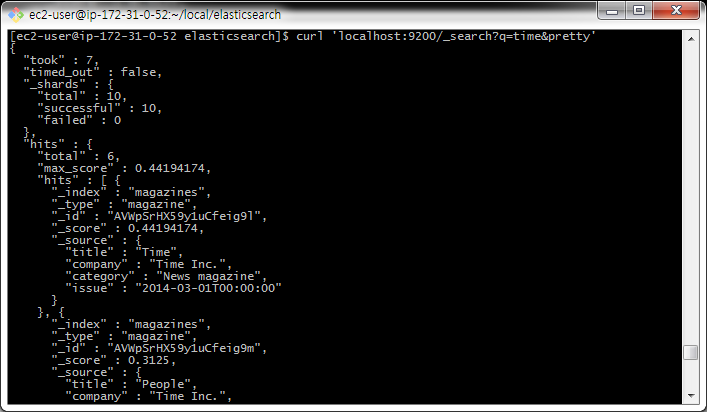
**검색 API**

* curl localhost:9200/books/book/\_search?q=keyword
* q 파라미터
* index/type 단위로 검색, 또는 index로 검색 가능
* 결과는 hits 필드에 배열로 표시
* url 자체를 문자열처럼 '로 감싸서 조회 가능
  + "took" : 검색 소요시간 밀리초 단위
* multi tenancy
  + curl 'localhost:9200/books,magazines/\_search?q=time&pretty'
  + curl 'localhost:9200/\_all/\_search?q=time&pretty'
  + curl 'localhost:9200/\_search?q=time&pretty'
* URI 검색
  + q
    - 필드명:질의어
      * curl 'localhost:9200/\_search?q=title:time&pretty'
    - 공백 처리
      * curl 'localhost:9200/\_search?q=title:time%20AND%20machine&pretty'
  + df(default field)
    - curl 'localhost:9200/\_search?q=time&df=title&pretty'
  + default\_operator
    - curl 'localhost:9200/\_search?q=time%20machine&default\_operator=AND&pretty'
  + explain
    - 상세 점수(score) 표시
    - score : 검색어에 해당하는 데이터의 정확도
    - 점수가 높을수록 상위에 표시
    - curl 'localhost:9200/\_search?q=title:time&explain&pretty'
  + \_source
    - 기본값은 true
    - false로 설정한 경우 hit와 score같은 메타 정보만 출력
    - curl 'localhost:9200/\_search?q=title:time&\_source=false&pretty'
  + fields
    - 출력 결과에 해당 지정된 필드만 표시
    - curl 'localhost:9200/\_search?q=title:time&fields=title,author,category&pretty'
  + sort
    - curl 'localhost:9200/\_search?q=author:jules&sort=pages&pretty'
    - curl 'localhost:9200/\_search?q=author:jules&sort=pages:desc&pretty'
    - curl 'localhost:9200/\_search?q=author:jules&fields=author,title&sort=title&pretty'
    - curl 'localhost:9200/\_search?q=author:jules&fields=author,title&sort=title:desc&pretty'
    - 값 전체로 정렬하려면 데이터 색인 전에 title 필드를 not\_analyzed로 매핑(mapping)해야 함(8장 참고)
  + from
    - 몇 번째부터 출력할지 지정, 기본값 0
    - curl 'localhost:9200/\_search?q=author:jules&fields=author,title&from=1&pretty'

**mutil tenancy 테스트 해보기**

****

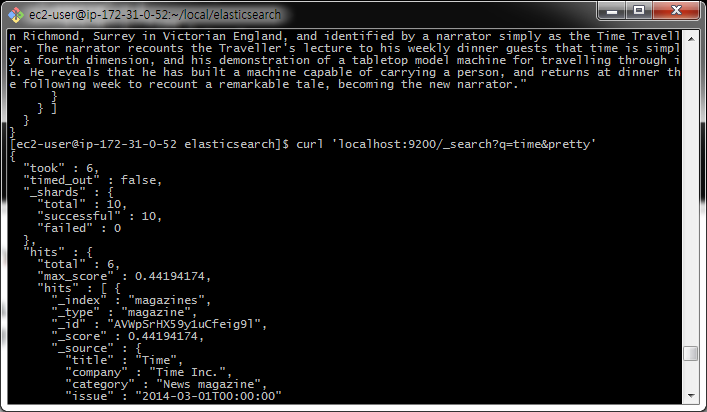
****

****

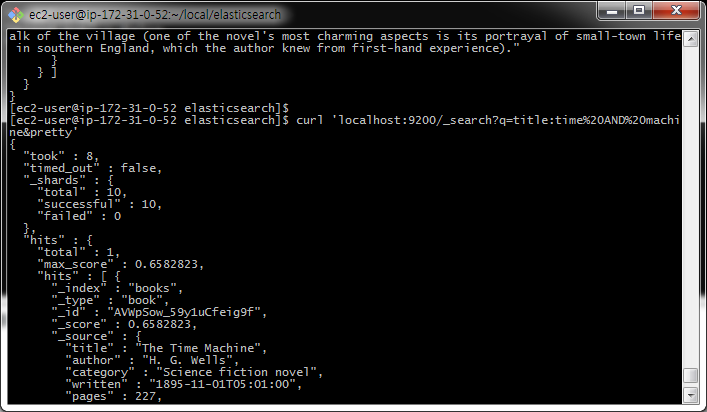
**TIP youtobe Y앞에 SS 붙이면 다운로드 받을 수 있습니다.**

**URL 검색**

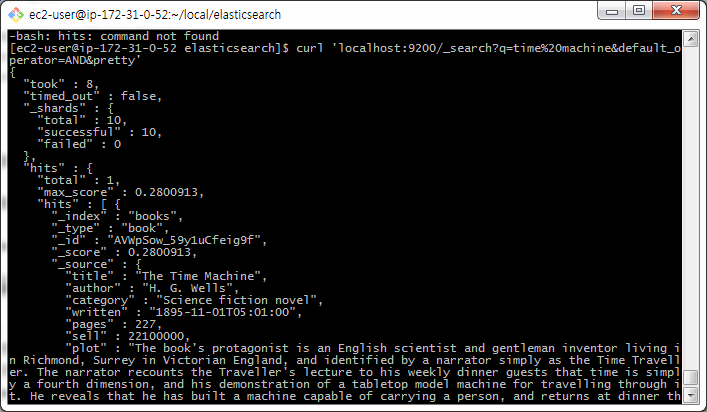
**대문자를 써도 소문자로 변환함 옵션이 있긴있음 대문자도 허용하도록**

****

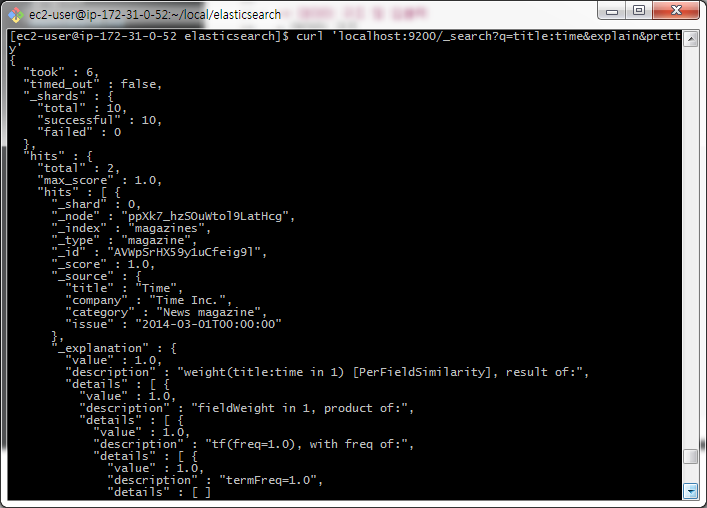
**%20 은 공백**

****

**논리연산을 넣은 방식**

****

**좀더 자세하게**

****

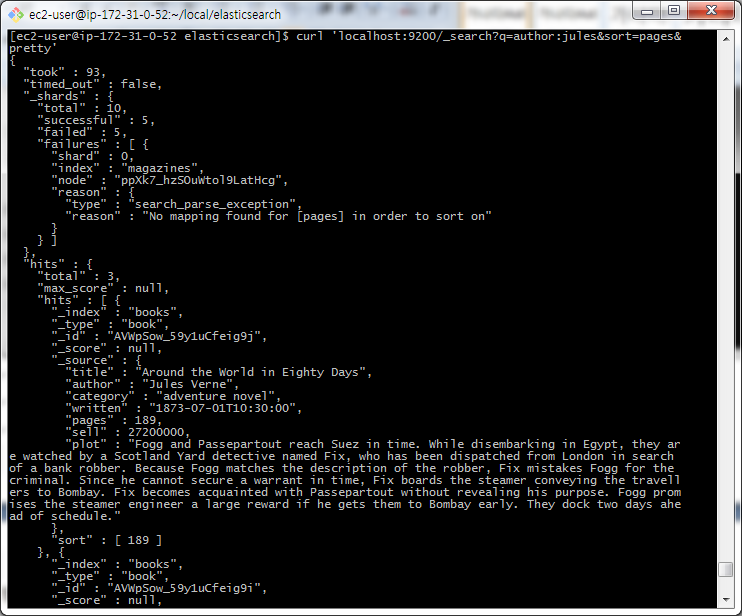
**간단하게**

****

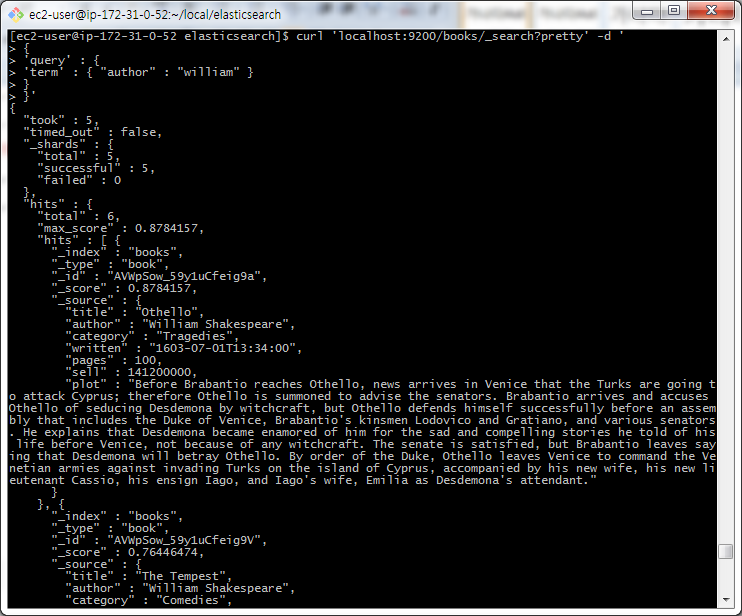
**원하는것만 보고 싶다.**

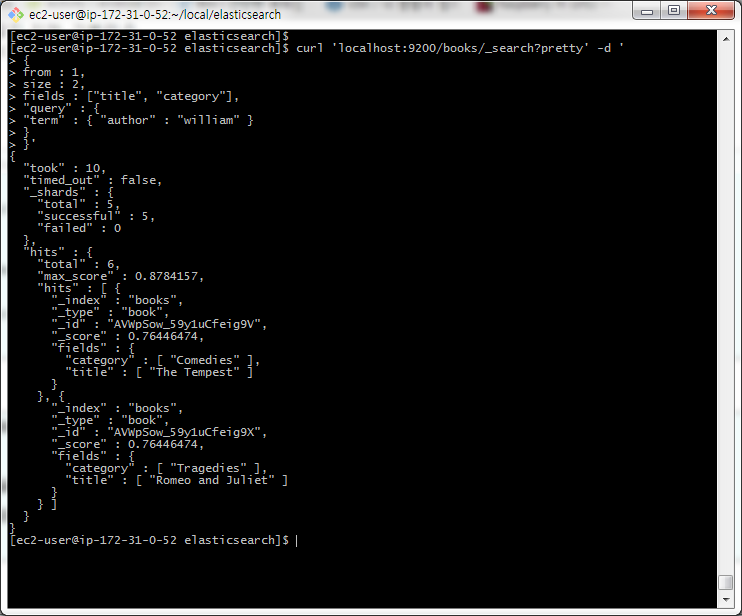
****

**정렬해서 보고 싶다.**

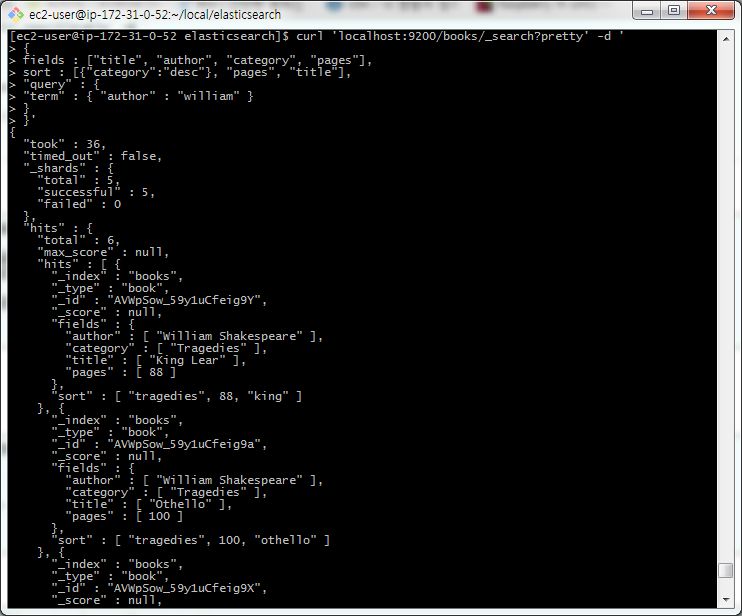
****

**Request BODY 방식으로 조회해 보기**

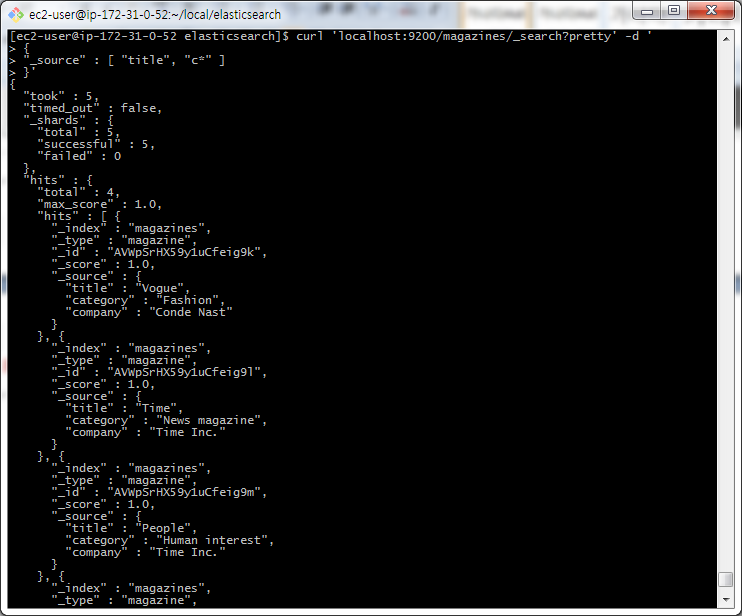
****

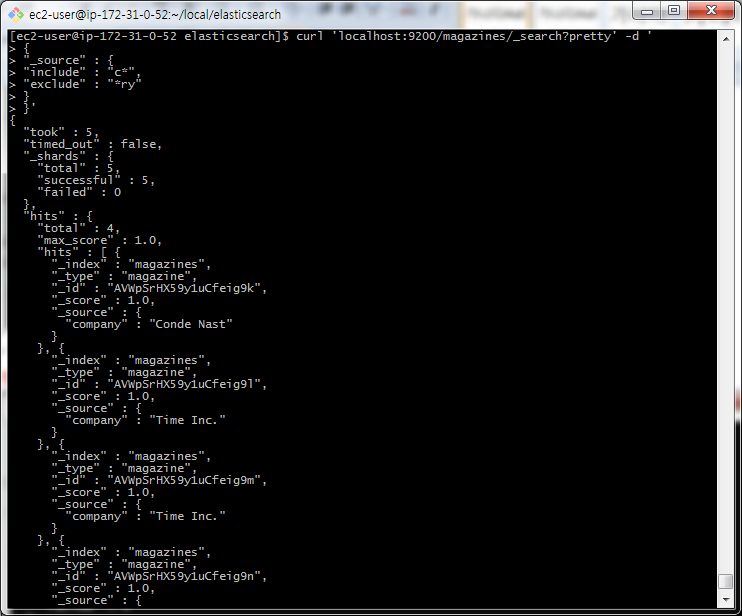
****

**정렬**

****

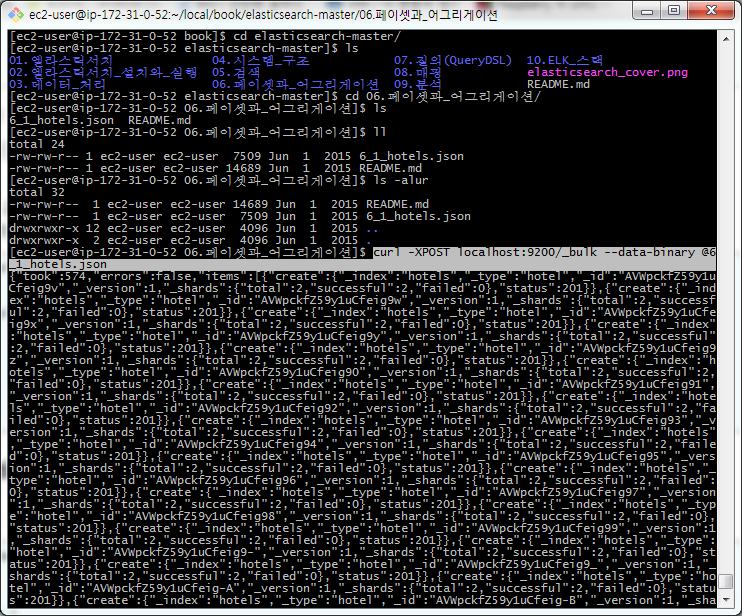
**조건에 따라 조회**

****

****

**어그리게이션 [통계등..]**

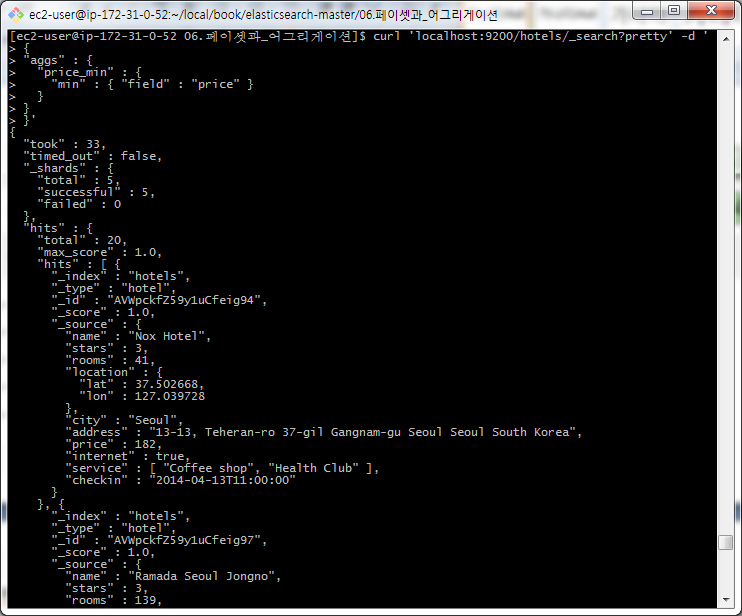
**선 데이터 넣기 작업**

****

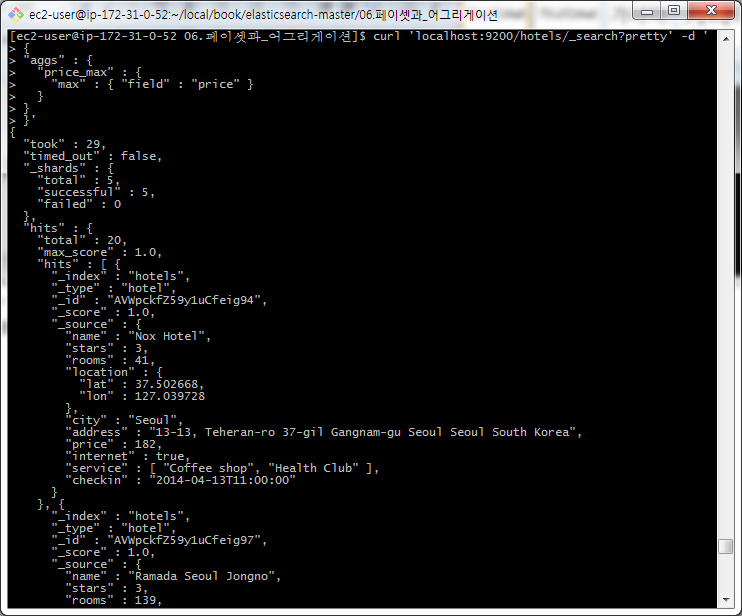
****

**어글리게이션 테스트 해보기**

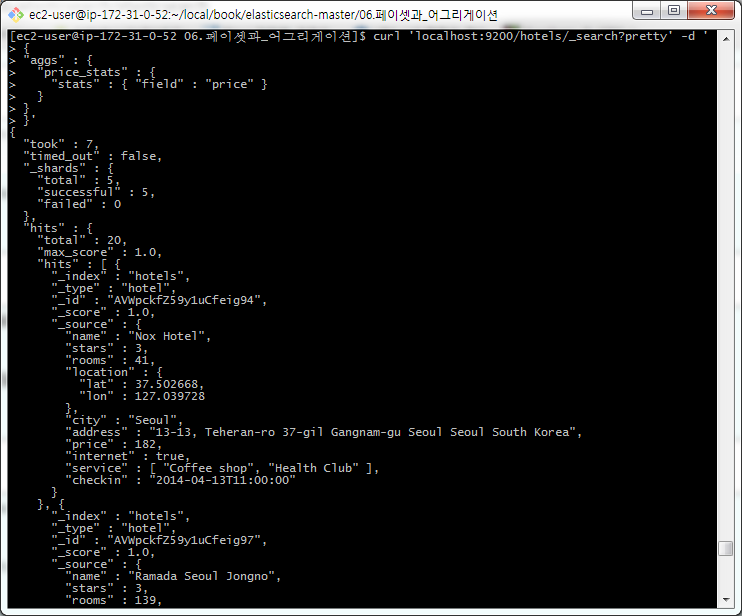
**가장 싼거**

****

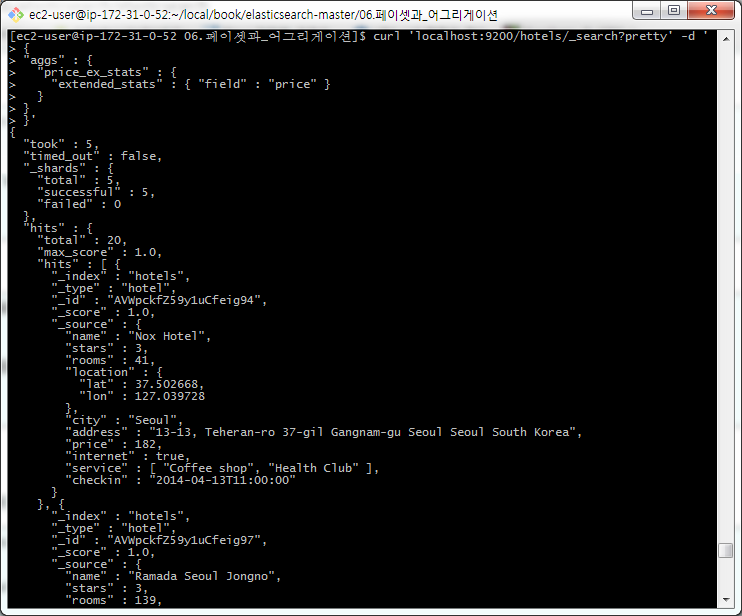
**최대값**

****

기본 통계 정보

****

확장된 통계 정보

****

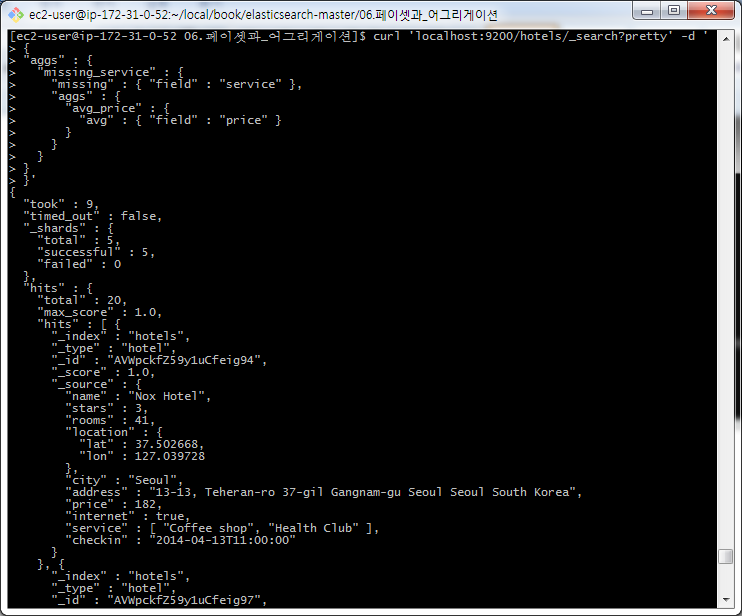
**글로벌 어그리게이션**

글로벌 and 하위

****

**누락(missing) 어그리게이션**

지정한 필드가 없거나 필드 값이 null인 도큐먼트를 담는 버킷 생성

****

오름차순, 내림차순 "\_term" : "asc" 할경우 오름차순

****

**범위, 날짜 범위 어그리게이션**

curl 'localhost:9200/hotels/\_search?pretty' -d '

{

"aggs" : {

"range\_room" : {

"range" : {

"field" : "rooms",

"ranges" : [{"to":500}, {"from":500, "to":1000}, {"from":1000}]

},

"aggs" : {

"avg\_price" : {

"avg" : { "field" : "price" }

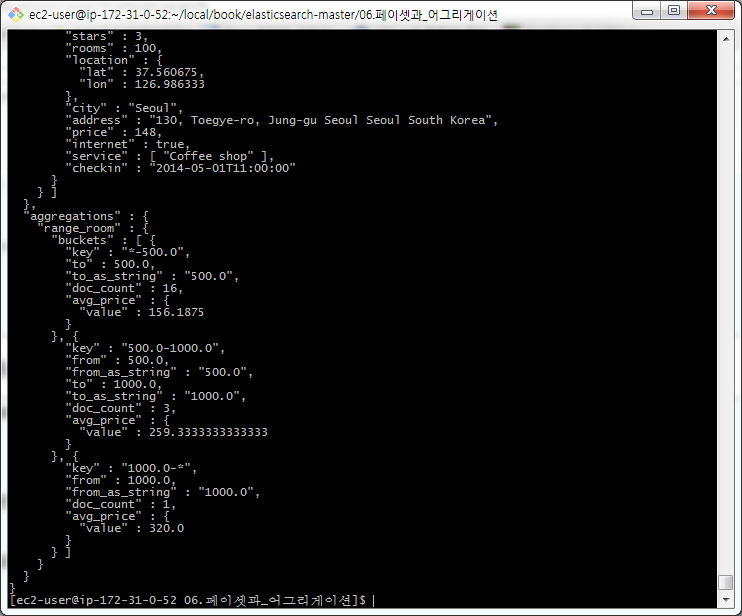
}

}

}

}

}'



**위치거리**

curl 'localhost:9200/hotels/\_search?pretty' -d '

{

"aggs" : {

"geo\_location" : {

"geo\_distance" : {

"field" : "location",

"origin" : "37.52, 126.98",

"distance\_type" : "plane",

"unit" : "km",

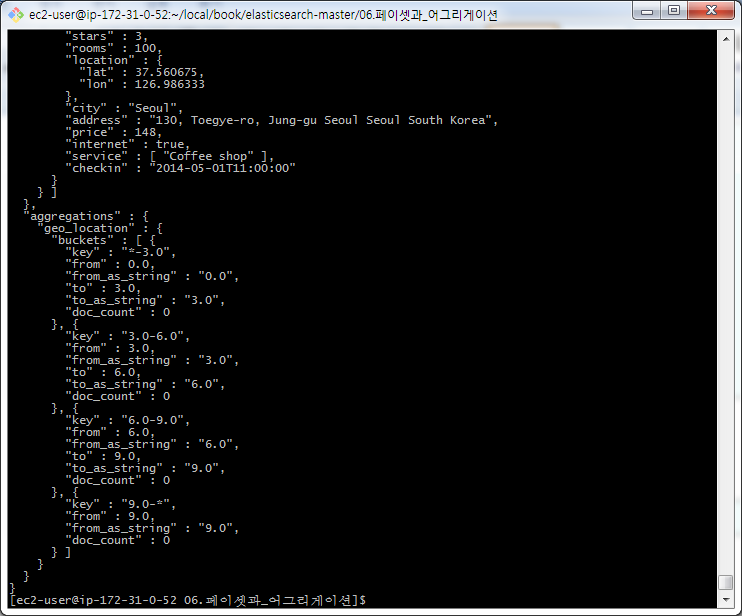
"ranges" : [{"to":3},{"from":3, "to":6},{"from":6, "to":9},{"from":9}]

}

}

}

}'

****

기타 사항은 http://okdevtv.com/mib/elasticsearch/elasticsearch 여기에 있는 명령어를 따라해 보자.

-------------이벤트 정보---------------

엘라스틱서치 취업회사 스토리

https://www.dropbox.com/s/6ao012omk03ierg/elastic-job.pdf?dl=0

로그스테쉬 (logstash)

정규 표현식

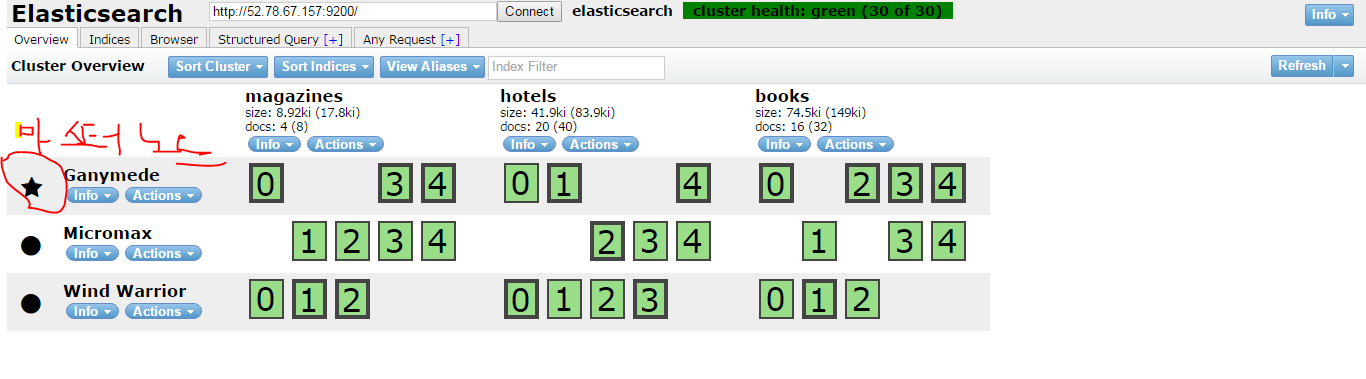
https://github.com/logstash-plugins/logstash-patterns-core/blob/master/patterns/grok-patterns

정규표현식 확인 사이트

https://regex101.com/

okjsp logstash 정보

ubuntu@ip-172-31-4-140:~/local/logstash-2.3.0$ cat logconf/okky.conf  
input {  
beats {  
port => 5044  
codec => plain {  
# iconv -f cp949 -t utf8 < foo.log > foo.log.utf8  
# iconv -f utf8 < foo.log  
charset => "UTF-8"  
# charset => "CP949"  
}  
}  
file {  
path => "/var/log/nginx/access.log"  
start\_position => beginning  
}  
}  
filter {  
if [message] =~ "^#|\.(css|js|ico|png|xml|jpg|JPG|gif|jpeg|eot\?) " {  
drop {}  
}  
grok {  
match => { "message" => "%{COMBINEDAPACHELOG}"}  
}  
date {  
match => [ "timestamp", "dd/MMM/yyyy:HH:mm:ss Z" ]  
}  
geoip {  
source => "clientip"  
}  
useragent {  
source => "agent"  
}  
  
# field added  
if [request] {  
mutate {  
add\_field => {  
"reqs" => "%{request}"  
}  
}  
}  
mutate {  
split => ["reqs", "?"]  
add\_field => { "uri" => "%{reqs[0]}" }  
add\_field => { "req\_uri" => "%{reqs[0]}" }  
}  
if [reqs][1] {  
mutate {  
add\_field => { "querystring" => "%{reqs[1]}" }  
}  
}  
if ![querystring] {  
mutate {  
add\_field => { "querystring" => "-" }  
}  
}  
if [uri] =~ "articles" {  
mutate {  
split => ["uri", "/"]  
add\_field => { "p\_category" => "%{uri[2]}" }  
}  
}  
urldecode {  
field => "querystring"  
}  
  
  
# params  
if [request] =~ "\?" {  
kv {  
field\_split => "&"  
source => "querystring"  
include\_keys => [ "query", "redirectUrl" ]  
prefix => "param\_"  
}  
}  
  
  
mutate {  
remove\_field => [  
"reqs",  
"uri"  
]  
}  
  
}  
output {  
elasticsearch {}  
# stdout {}  
}



kibana는 마스터만 바라보면 됩니다. 연결할때에~ 엘라스틱 서치랑

https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elk-stack-on-ubuntu-14-04

로그인 기능이 필요하면

/etc/nginx/sites-available/default

* server {
* listen 80;
* server\_name example.com;
* auth\_basic "Restricted Access";
* auth\_basic\_user\_file /etc/nginx/htpasswd.users;
* location / {
* proxy\_pass http://localhost:5601;
* proxy\_http\_version 1.1;
* proxy\_set\_header Upgrade $http\_upgrade;
* proxy\_set\_header Connection 'upgrade';
* proxy\_set\_header Host $host;
* proxy\_cache\_bypass $http\_upgrade;
* }
* }

페이스북 그룹

https://www.facebook.com/groups/elasticsearch.kr/

ELK 성능을 올리고 싶다면

https://www.elastic.co/kr/blog/performance-indexing-2-0

인덱싱 최대화

https://www.elastic.co/kr/blog/performance-considerations-elasticsearch-indexing

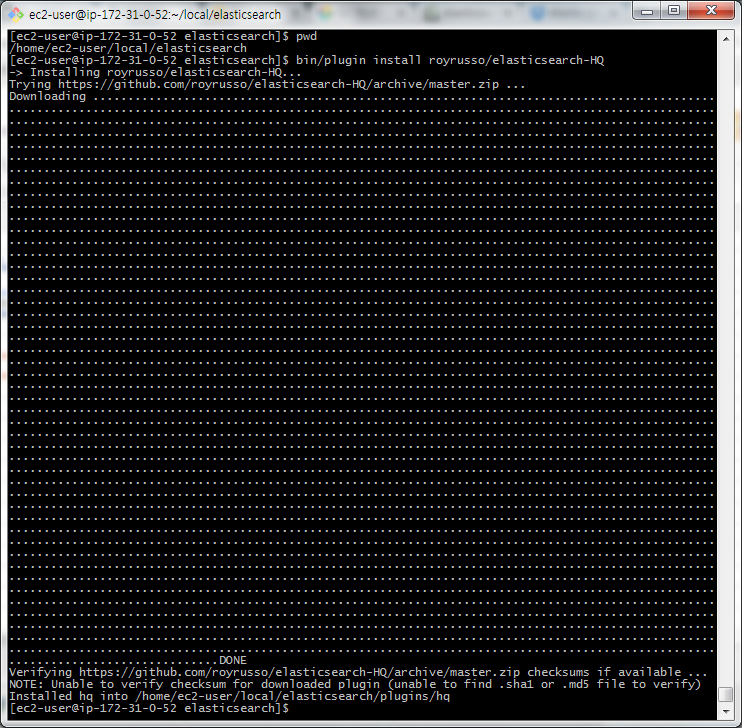
-------------이벤트 정보 종료---------------

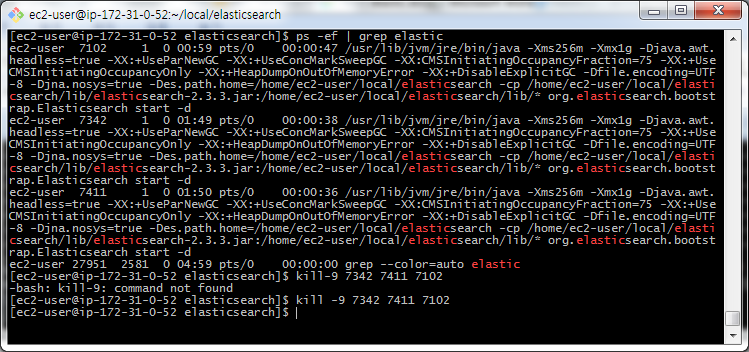
부분삭제

-------- elastic plugin 설치

http://www.elastichq.org/

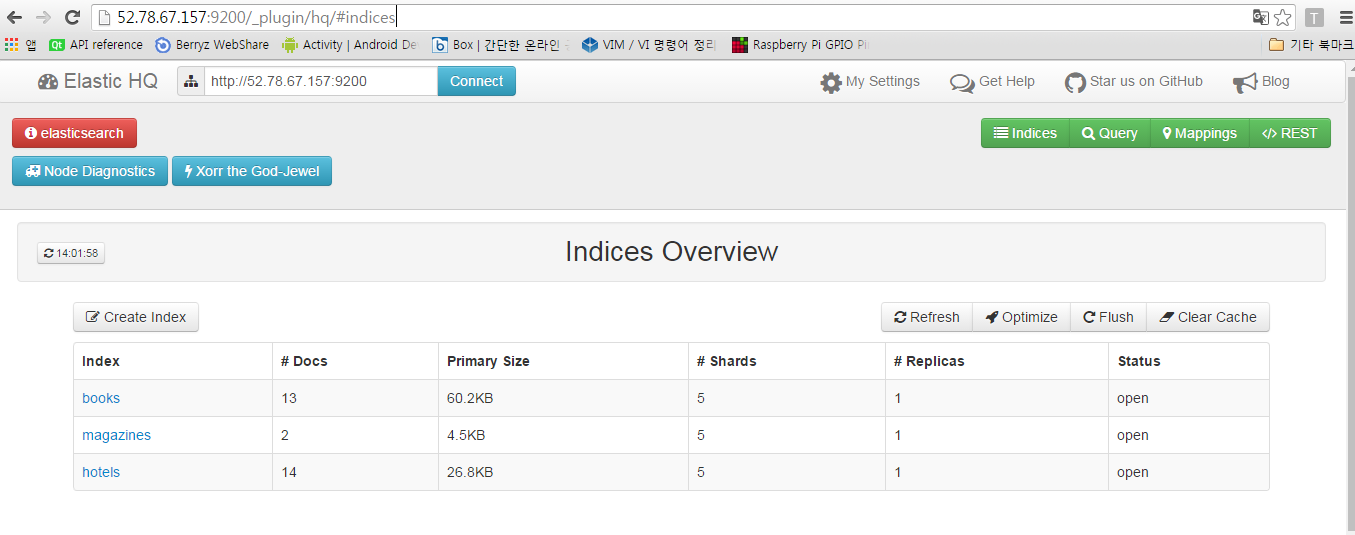
http://www.elastichq.org/support.html





/bin/elasticsearch -d 다시 실행후 아래 URL로 접속을 해보자

http://52.78.67.157:9200/\_plugin/hq/



**질의(QueryDSL)**

* Query
  + 전문 검색(full text search)
  + scoring
  + 결과 캐싱 안함
  + 응답속도 느림
* Filter
  + Y/N조건의 바이너리 구분
  + no scoring
  + 결과 캐싱됨
  + 응답속도 빠름

Query

형태소 분석

* The Prince and the Pauper → the, prince, and, pauper
* 모두 소문자로, 중복 삭제
* the, prince, and, pauper 같은 토큰을 텀term 이라고 함

대문자 소문자 했을 때 결과 비교

|  |
| --- |
| **curl 'localhost:9200/books/\_search?pretty' -d '**  **> {**  **> "query" : {**  **> "term" : {**  **> "title" : "prince"**  **> }**  **> }**  **> }'**  {  "took" : 6,  "timed\_out" : false,  "\_shards" : {  "total" : 5,  "successful" : 4,  "failed" : 0  },  "hits" : {  "total" : 1,  "max\_score" : 0.614891,  "hits" : [ {  "\_index" : "books",  "\_type" : "book",  "\_id" : "AVWpSow\_59y1uCfeig9c",  "\_score" : 0.614891,  "\_source" : {  "title" : "The Prince and the Pauper",  "author" : "Mark Twain",  "category" : "Children's literature",  "written" : "1881-08-01T10:34:00",  "pages" : 79,  "sell" : 112100000,  "plot" : "Tom Canty (youngest son of a family living with the dregs of society in Offal Court) has always aspired to a better life, encouraged by the local priest (who has taught him to read and write). Loitering around the palace gates one day, he sees a prince (the Prince of Wales – Edward VI). Tom is nearly caught and beaten by the Royal Guards; however, Edward stops them and invites Tom into his palace chamber. There the two boys get to know one another, fascinated by each other's life and their uncanny resemblance. They decide to switch clothes \"temporarily\". Edward leaves in a hurry before the boys are caught at their game, snatching up an article of national importance (which the reader later learns is the Great Seal of England). Soon Prince Edward is trying to escape the brutality of Tom's abusive drunken father. Tom, posing as the prince, tries to cope with court customs and manners. His fellow nobles and palace staff think \"the prince\" has an illness which has caused memory loss and fear he will go mad. They repeatedly ask him about the missing \"Great Seal\", but he knows nothing about it; however, when Tom is asked to sit in on judgments, his common-sense observations reassure them his mind is sound."  }  } ]  }  }  [ec2-user@ip-172-31-0-52 elasticsearch]$ **curl 'localhost:9200/books/\_search?pretty' -d '**  **> {**  **> "query" : {**  **> "term" : {**  **> "title" : "Prince"**  **> }**  **> }**  **> }'**  {  "took" : 1,  "timed\_out" : false,  "\_shards" : {  "total" : 5,  "successful" : 4,  "failed" : 0  },  "hits" : {  "total" : 0,  "max\_score" : null,  "hits" : [ ]  }  }  [ec2-user@ip-172-31-0-52 elasticsearch]$ |

매핑 ----- DDL ------ 과 같은 효과

[ec2-user@ip-172-31-0-52 elasticsearch]$ curl 'localhost:9200/books/\_mapping?pretty'

{

"books" : {

"mappings" : {

"book" : {

"properties" : {

"author" : {

"type" : "string"

},

"category" : {

"type" : "string"

},

"date" : {

"type" : "date",

"format" : "strict\_date\_optional\_time||epoch\_millis"

},

"pages" : {

"type" : "long"

},

"plot" : {

"type" : "string"

},

"sell" : {

"type" : "long"

},

"title" : {

"type" : "string"

},

"written" : {

"type" : "date",

"format" : "strict\_date\_optional\_time||epoch\_millis"

}

}

}

}

}

}

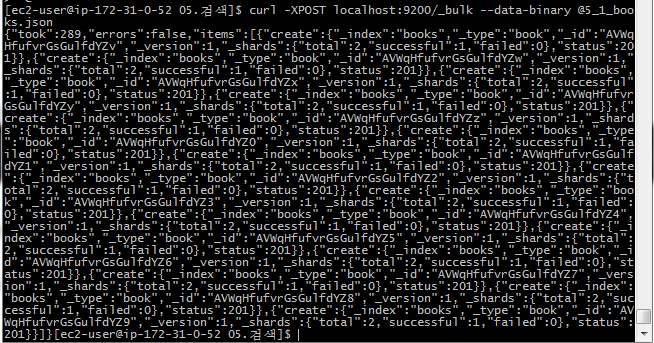
북인덱스 삭제



지워진 상태에서 매핑 생성해야됨

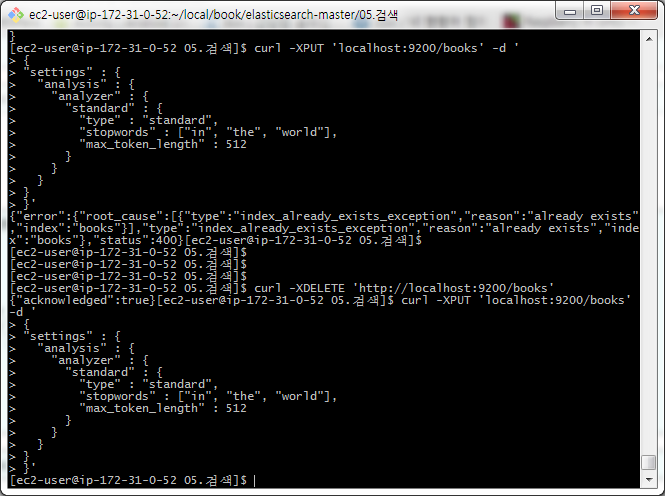
다시 적제

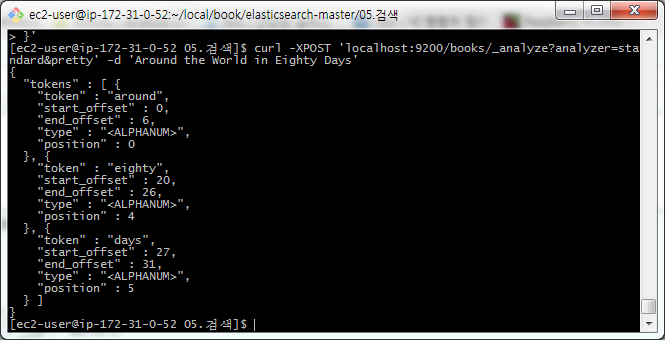
curl -XPOST localhost:9200/\_bulk --data-binary @5\_1\_books.json



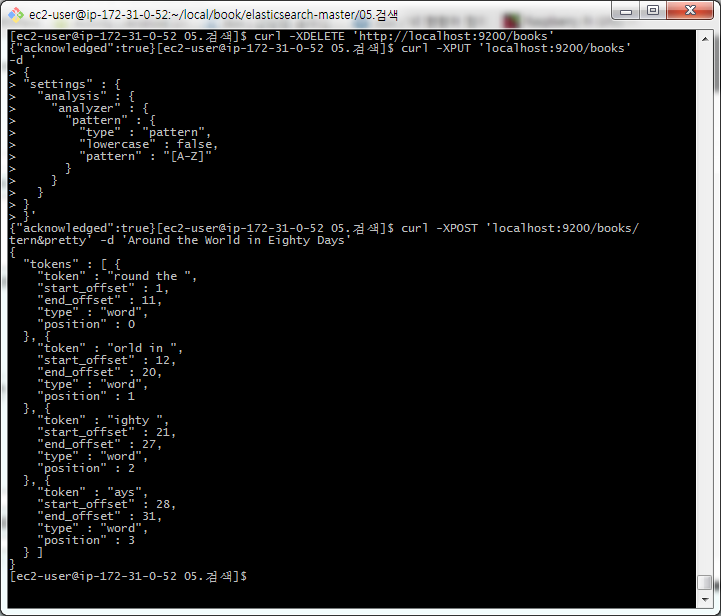
-------- 분석 --------

books 인덱스 삭제 후 분석기 설정

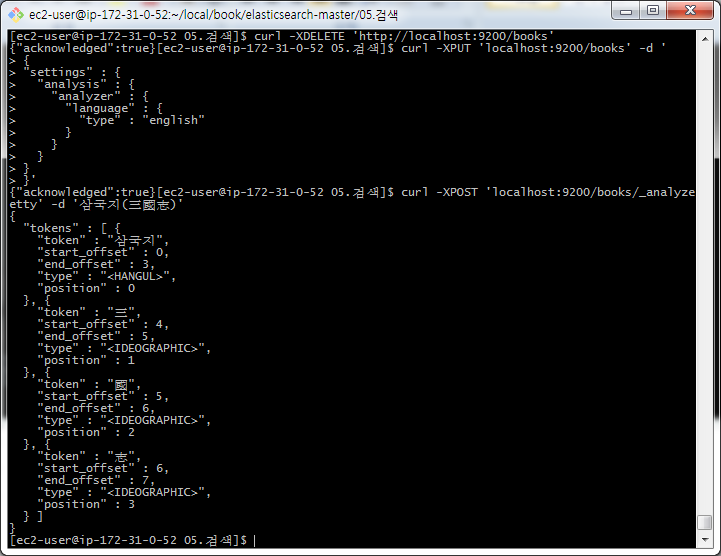


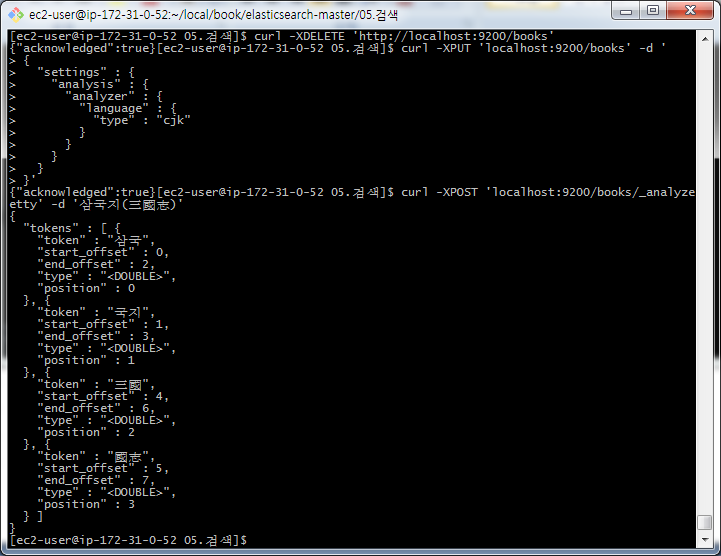


패턴분석



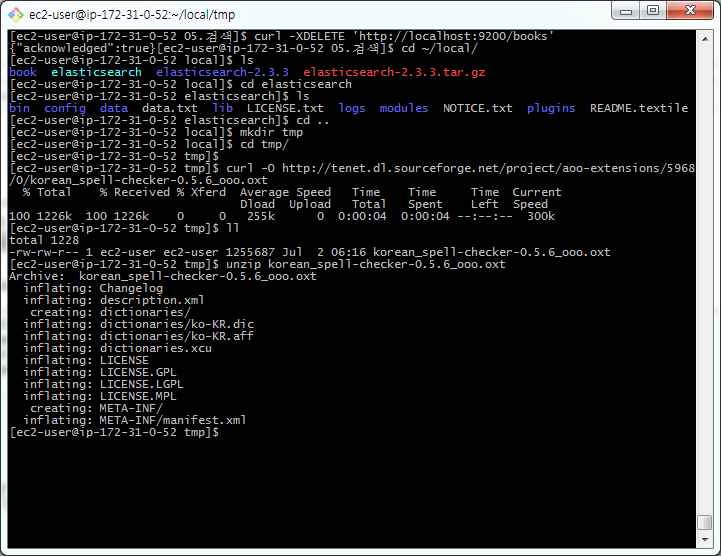
다국어





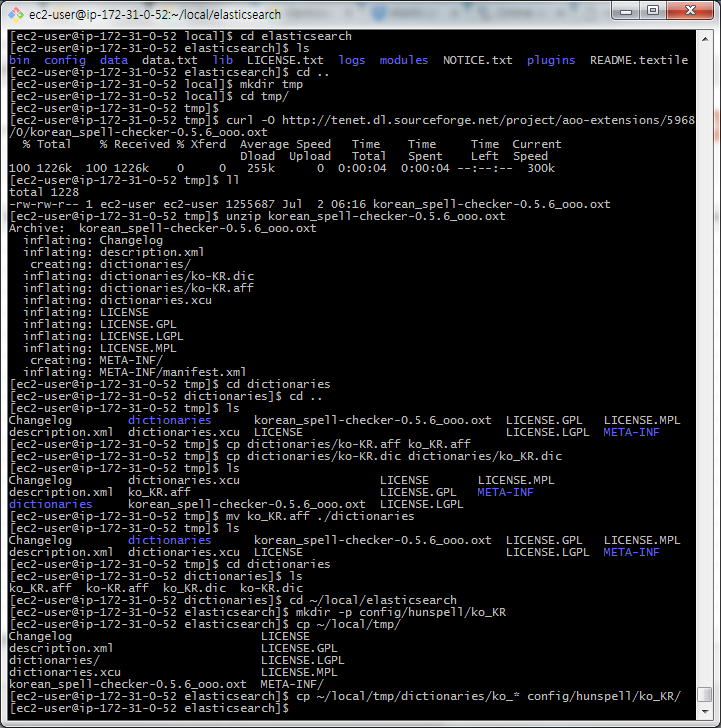
hunspell 토큰필터

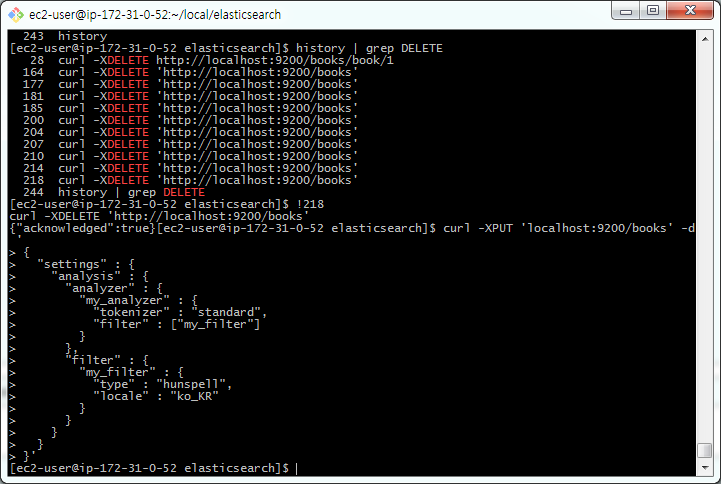
다운로드



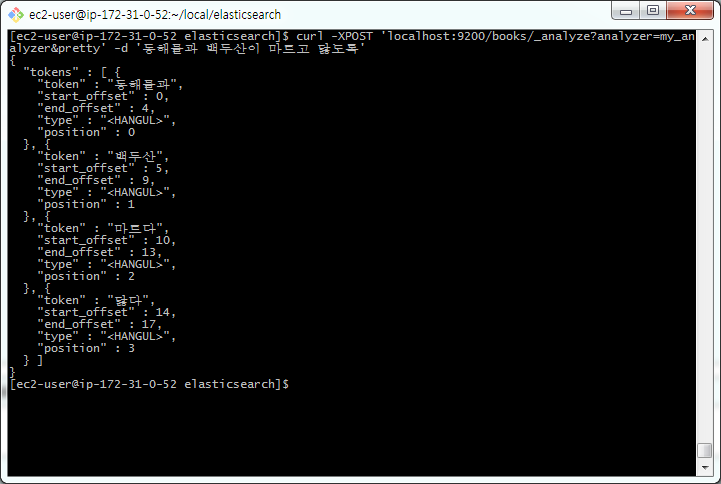
압축풀고 파일이름 바꺼주고

위치 옴겨주기

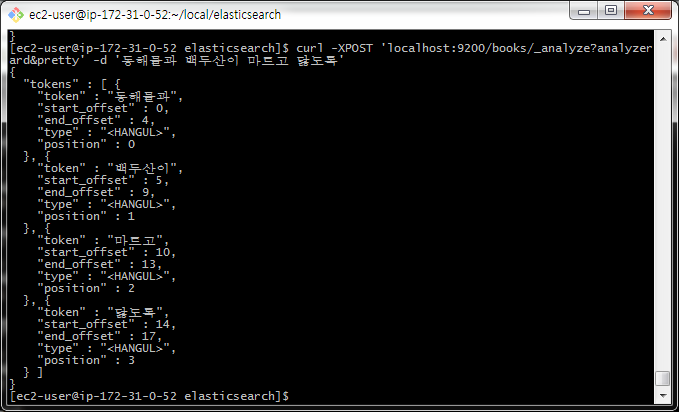




my\_analyzer



standard



https://github.com/chanil1218/elasticsearch-analysis-korean