엘라스틱서치

김종민
이메일 : jongmin.kim@elastic.co
블로그 : http://kimjmin.net

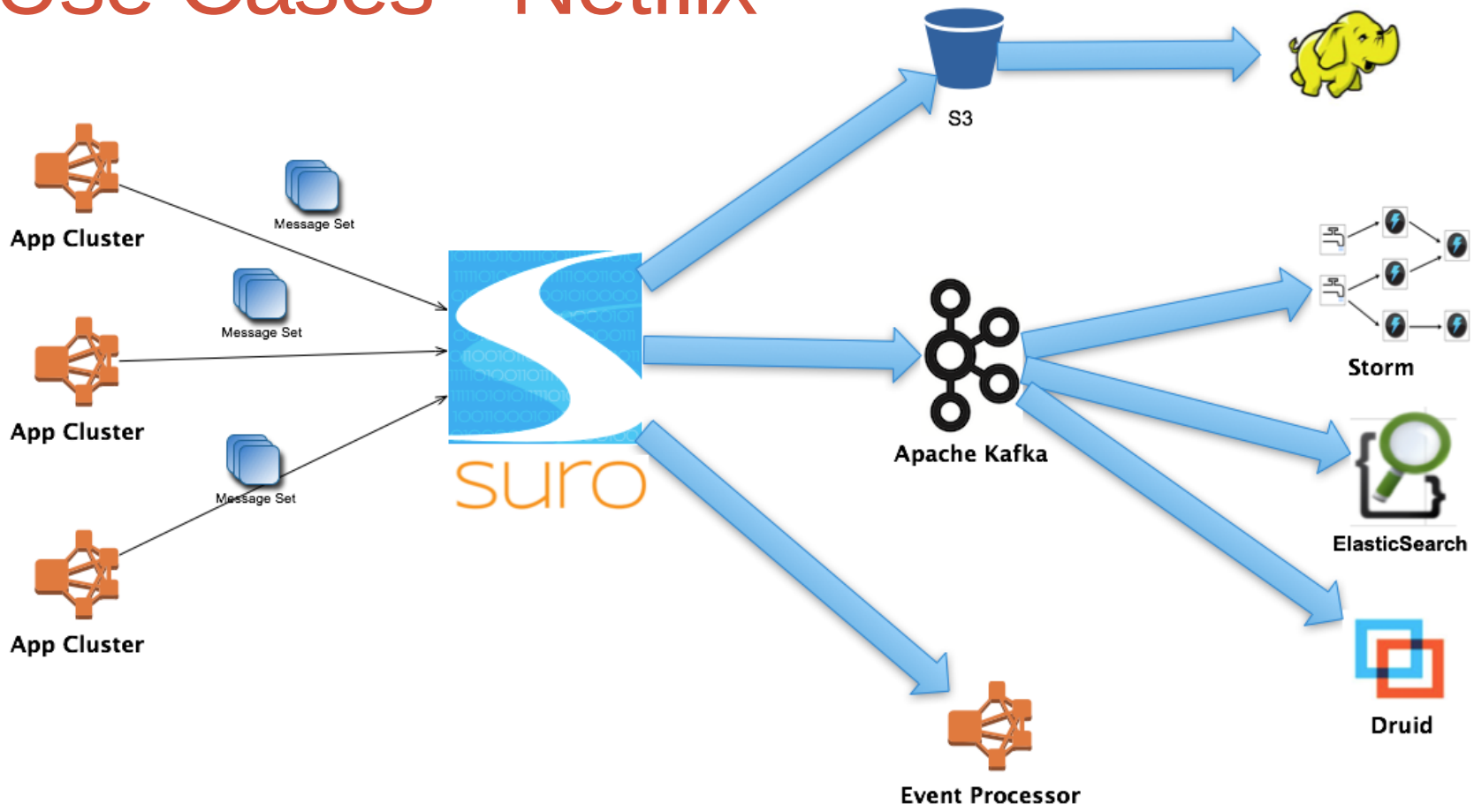
엘라스틱서치 - Elasticsearch

- http://elastic.co
- Open Source - https://github.com/elastic/elasticsearch
- Java
- Apache Lucene
- Restful
- JSON Document Based
- Real-time Search
- Full-text Search

Use Cases

- Github, Sourceforge…

Use Cases - Netflix



데이터 저장 – 관계 DB

PK	Text
Doc 1	blue sky green land red sun
Doc 2	blue ocean green land
Doc 3	red flower blue sky

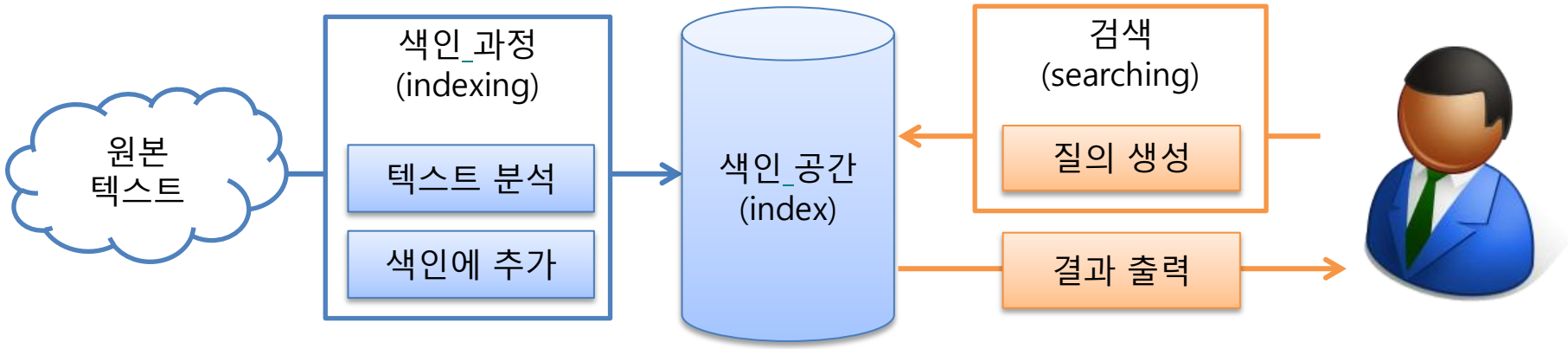
- PK, Index, 칼럼을 기준으로 순서대로 검색.

데이터 저장 – 역파일 색인

검색어 (term)	검색어가 가리키는 대상 문서	검색어 (term)	검색어가 가리키는 대상 문서
blue	Doc 1, Doc 2, Doc 3	red	Doc 1, Doc 3
sky	Doc 1, Doc 3	ocean	Doc 2
green	Doc 1, Doc 2	flower	Doc 3
land	Doc 1, Doc 2	sun	Doc 1

- 본문의 검색어를 먼저 추출한 뒤 검색어에 해당하는 문서를 찾음

데이터 저장 프로세스


원본
텍스트
색인_과정
(indexing)
텍스트 분석
색인에 추가
색인_공간
(index)
검색
(searching)
질의 생성
결과 출력

관계DB vs 엘라스틱서치

HTTP	CRUD	SQL
GET	Read	Select
PUT	Update	Update
POST	Create	Insert
DELETE	Delete	Delete

관계 DB	엘라스틱서치
데이터베이스 (Database)	인덱스 (index)
테이블(Table)	타입(Type)
열(Row)	도큐먼트 (Document)
행(Column)	필드(Field)
스키마(Schema)	매핑(Mapping)

Restful API

- 단일 URL를 통한 자원의 접근
- http 메소드를 이용해서 자원 처리
- Not Rest
  - 추가 : http://site.com/user.jsp?cmd=add&id=user1&name=kim
  - 조회 : http://site.com/user.jsp?id=user1
  - 수정 : http://site.com/user.jsp?cmd=modify&id=user1&name=lee
  - 삭제 : http://site.com/user.jsp?cmd=delete&id=user1
- Rest
  - 추가 : -POST http://site.com/user/user1 {name:kim}
  - 조회 : -GET http://site.com/user/user1
  - 수정 : -PUT http://site.com/user/user1 {name:lee}
  - 삭제 : -DELETE http://site.com/user/user1

엘라스틱서치 Rest API

- http://host:port/인덱스/타입/도큐먼트 id
- curl -X‘메서드’ http://host:port/인덱스/타입/도큐먼트 id -d ‘{데이터}’

```
$ curl -XPUT http://localhost:9200/books/book/1 -d '
{
  "title" : "Elasticsearch Guide",
  "author" : "Kim",
  "date" : "2014-05-01",
  "pages" : 250
}
'
{"_index":"books","_type":"book","_id":"1","_version":1,"created":true}
```

엘라스틱서치 Rest API

```
$ curl -XGET http://localhost:9200/books/book/1
{"_index":"books","_type":"book","_id":"1","_version":1,"found":true, "_source" :
  {
    "title" : "Elasticsearch Guide",
    "author" : "Kim",
    "date" : "2014-05-01",
    "pages" : 250
  }
}
```

클러스터(cluster)

- 엘라스틱서치 시스템의 가장 큰 단위
- 하나의 클러스터는 다수의 노드로 구성
- 하나의 클러스터를 다수의 서버로 바인딩 해서 운영, 또는 역으로 하나의 서버에서 다수의 클러스터 운용 가능

config/elasticsearch.yml

```
cluster.name: elasticsearch
```

```
$ bin/elasticsearch --cluster.name=elasticsearch
```

노드 (Node)

- 엘라스틱서치를 구성하는 하나의 단위 프로세스
- 다수의 샤드로 구성됨
- 같은 클러스터명을 가진 노드들은 자동으로 바인딩 됨

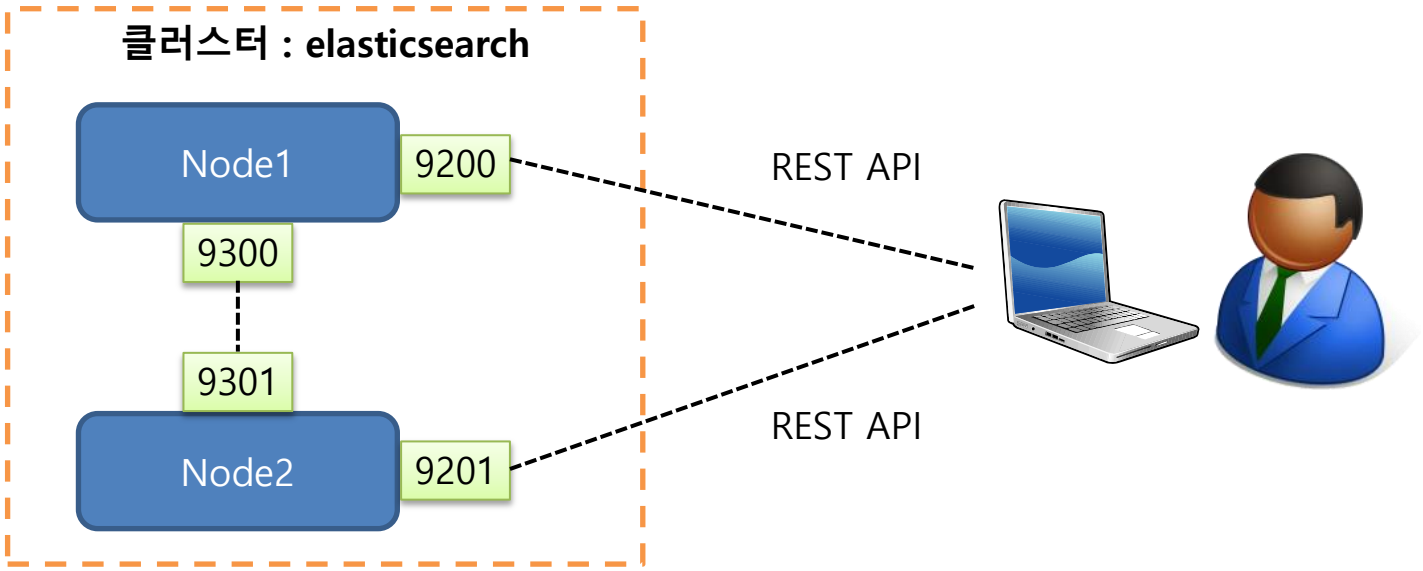
config/elasticsearch.yml

```
node.name: "Node1"
```

```
$ bin/elasticsearch --node.name=Node1
```

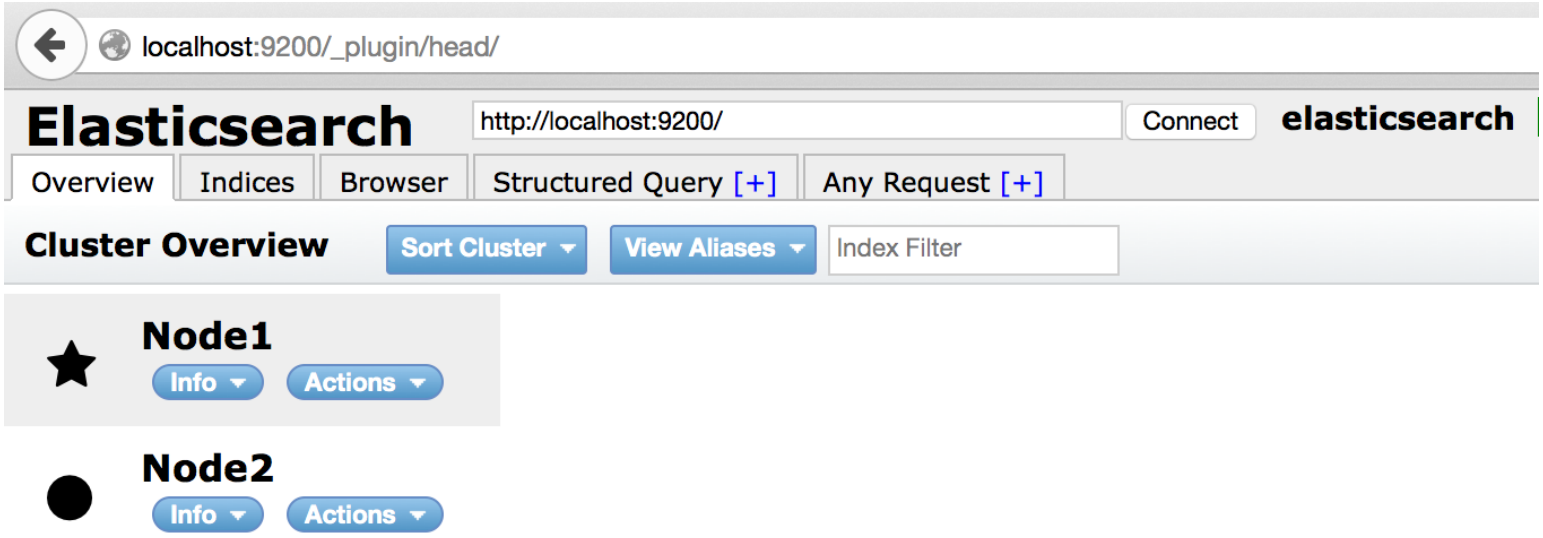
노드 (Node)

- http 통신 포트 : 9200~ 차례대로 증가
- 노드 간 데이터 교환 포트 : 9300~ 차례대로 증가

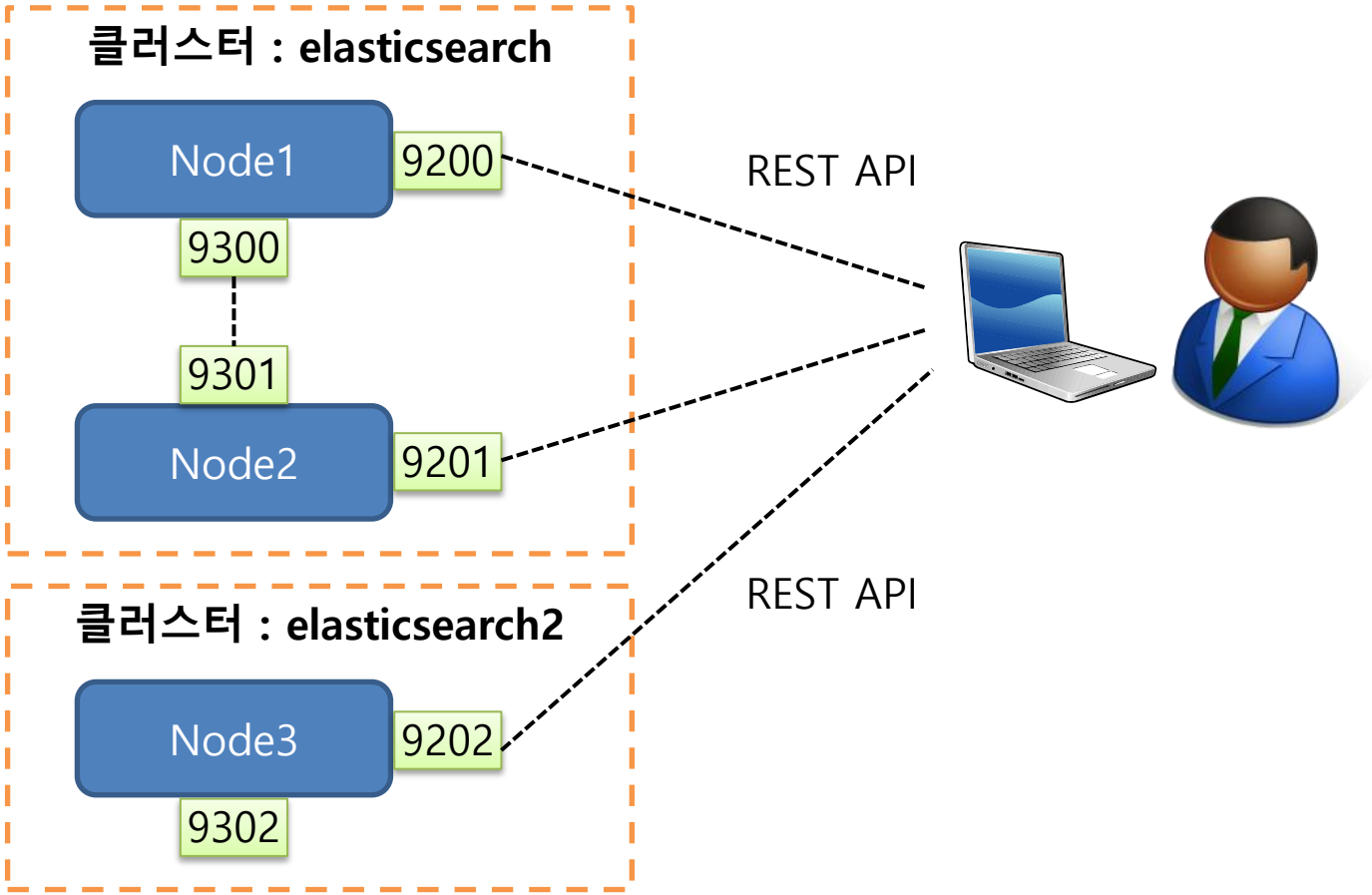


head 플러그인

```
$ bin/plugin --install mobz/elasticsearch-head
```



```
$ bin/elasticsearch --cluster.name=elasticsearch --node.name=Node1
$ bin/elasticsearch --cluster.name=elasticsearch --node.name=Node2
$ bin/elasticsearch --cluster.name=elasticsearch2 --node.name=Node3
```



master node & data node

- 마스터 노드 : 클러스터 상태 관리
- 데이터 노드 : 데이터 입/출력, 검색 수행

config/elasticsearch.yml

```
node.master: true
node.data: true
```

```
$ bin/elasticsearch --node.master=true --node.data=true
```

```
$ bin/elasticsearch --node.name=Node1 --node.master=true --node.data=false
$ bin/elasticsearch --node.name=Node2 --node.master=false --node.data=true
$ bin/elasticsearch --node.name=Node3 --node.master=false --node.data=true
$ bin/elasticsearch --node.name=Node4 --node.master=false --node.data=true
```

샤드 (shard) & 레플리카 (replica)

- 샤드 : 데이터 검색 단위 인스턴스
- 레플리카 : 샤드의 복사본

config/elasticsearch.yml

```
index.number_of_shards: 5
index.number_of_replicas: 1
```

```
$ curl -XPUT localhost:9200/books -d '
{
  "settings" : {
    "number_of_shards" : 5,
    "number_of_replicas" : 1
  }
}'
```

검색

```
$ curl -XPUT localhost:9200/books/book/1 -d '
{ "title": "Romeo and Juliet", "author": "William Shakespeare", "category":"Tragedies", "written": "1562-12-01T20:40:00", "pages" : 125 }'
```

```
$ curl -XPUT localhost:9200/books/book/2 -d '
{ "title" : "Hamlet", "author": "William Shakespeare", "category":"Tragedies", "written": "1599-06-01T12:34:00", "pages" : 172 }'
```

```
$ curl -XPUT localhost:9200/books/book/3 -d '
{ "title": "The Prince and the Pauper", "author": "Mark Twain", "category":"Children literature", "written": "1881-08-01T10:34:00", "pages" : 79}'
```

검색

```
$ curl localhost:9200/books/_search?pretty=true
{ … 중략 …
},
  "hits" : {
    "total" : 3,
    "max_score" : 1.0,
    "hits" : [ {
… 중략 …
      "_source":
{ "title": "Romeo and Juliet", "author": "William Shakespeare", "category":"Tragedies", "written
": "1562-12-01T20:40:00", "pages" : 125 }
    },
… 중략 …
```

검색 - URI 검색

```
$ curl 'http://localhost:9200/books/_search?q=william&pretty=true'
```

```
$ curl 'http://localhost:9200/books/_search?q=author:william&pretty=true'
```

```
$ curl 'http://localhost:9200/books/_search?q=author:william&fields=title,author,pages&pretty=true'
```

검색 - Request Body 검색

```
$ curl 'http://localhost:9200/books/_search?pretty=true' -d '
{
  "query" : {
    "match" : {
      "author" : "William"
    }
  }
}'
```

텀(Term) 확인 - facet

```
$ curl 'localhost:9200/books/_search
?pretty' -d '
{
  "facets" : {
    "author_terms" : {
      "terms" : { "field" : "author" }
    }
  }
}'
```

```
"facets" : {
    "author_terms" : {
… 중략 …
      "terms" : [ {
        "term" : "william",
        "count" : 2
      }, {
        "term" : "shakespeare",
        "count" : 2
      }, {
        "term" : "twain",
        "count" : 1
      }, {
        "term" : "mark",
        "count" : 1
      } ]
```

텀(Term)

검색어 (term)	검색어가 가리키는 대상 문서
william	books/book/1, books/book/2
shakespeare	books/book/1, books/book/2
twain	books/book/3
mark	books/book/3

검색 - Request Body 검색

```
$ curl 'http://localhost:9200/books/_search?pretty=true' -d '
{
  "query" : {
    "match" : {
      "author" : "William"
    }
  }
}'
```

```
$ curl 'http://localhost:9200/books/_search?pretty=true' -d '
{
  "query" : {
    "term" : {
      "author" : "William"
    }
  }
}'
```

매핑

```
$ curl -XPUT localhost:9200/books -d '
{
  "mappings" : {
    "book" : {
      "properties" : {
        "author" : {
          "type" : "string",
          "index" : "not_analyzed"
        }
      }
    }
  }
}'
```

```
$ curl -XPUT localhost:9200/books/book/1 -d '
{ "title": "Romeo and Juliet", "author": "William Shakespeare", "category":"Tragedies", "written": "1562-12-01T20:40:00", "pages" : 125 }'
```

```
$ curl -XPUT localhost:9200/books/book/2 -d '
{ "title" : "Hamlet", "author": "William Shakespeare", "category":"Tragedies", "written": "1599-06-01T12:34:00", "pages" : 172 }'
```

```
$ curl -XPUT localhost:9200/books/book/3 -d '
{ "title": "The Prince and the Pauper", "author": "Mark Twain", "category":"Children literature", "written": "1881-08-01T10:34:00", "pages" : 79}'
```

텀(Term) 확인 - facet

```
$ curl 'localhost:9200/books/_search
?pretty' -d '
{
  "facets" : {
    "author_terms" : {
      "terms" : { "field" : "author" }
    }
  }
}'
```

```
"facets" : {
    "author_terms" : {
      "_type" : "terms",
      "missing" : 0,
      "total" : 3,
      "other" : 0,
      "terms" : [ {
        "term" : "William Shakespeare",
        "count" : 2
      }, {
        "term" : "Mark Twain",
        "count" : 1
      } ]
    }
  }
```

검색 - Request Body 검색

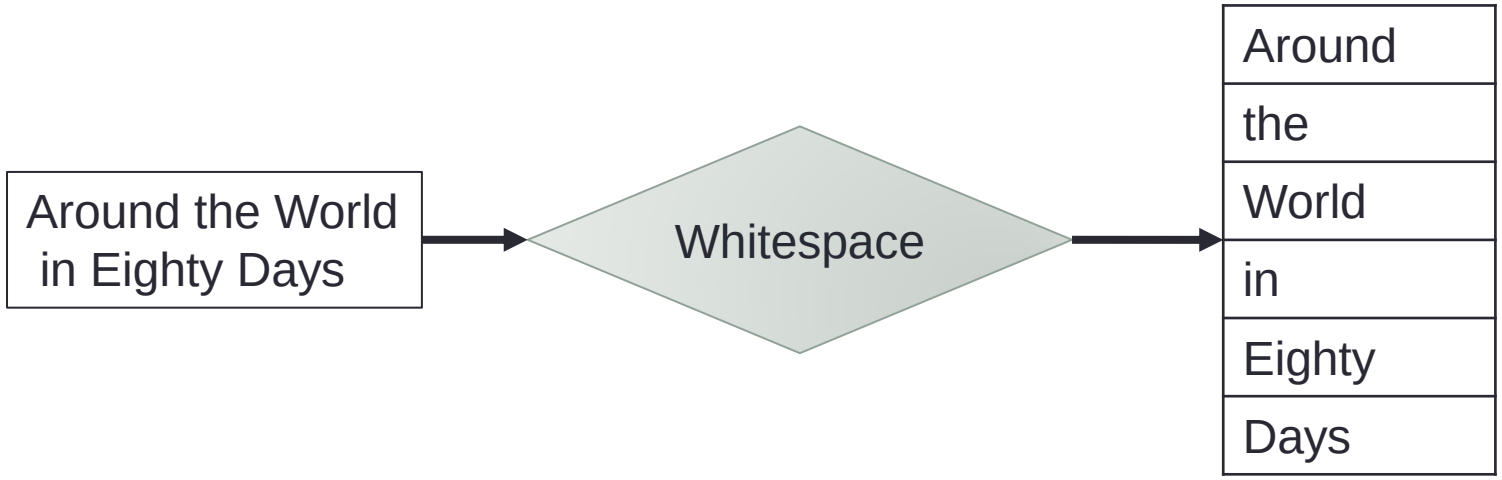
```
$ curl 'http://localhost:9200/books/_search?pretty=true' -d '
{
  "query" : {
    "term" : {
      "author" : "William Shakespeare"
    }
  }
}'
```

분석(Analyze)

- 애널라이저(Analyzer)를 이용해서 입력된 문장을 텀(term)으로 분해하는 과정
- 1 개의 토크나이저 (Tokenizer)
- 0~n 개의 토큰 필터 (Token Filter)

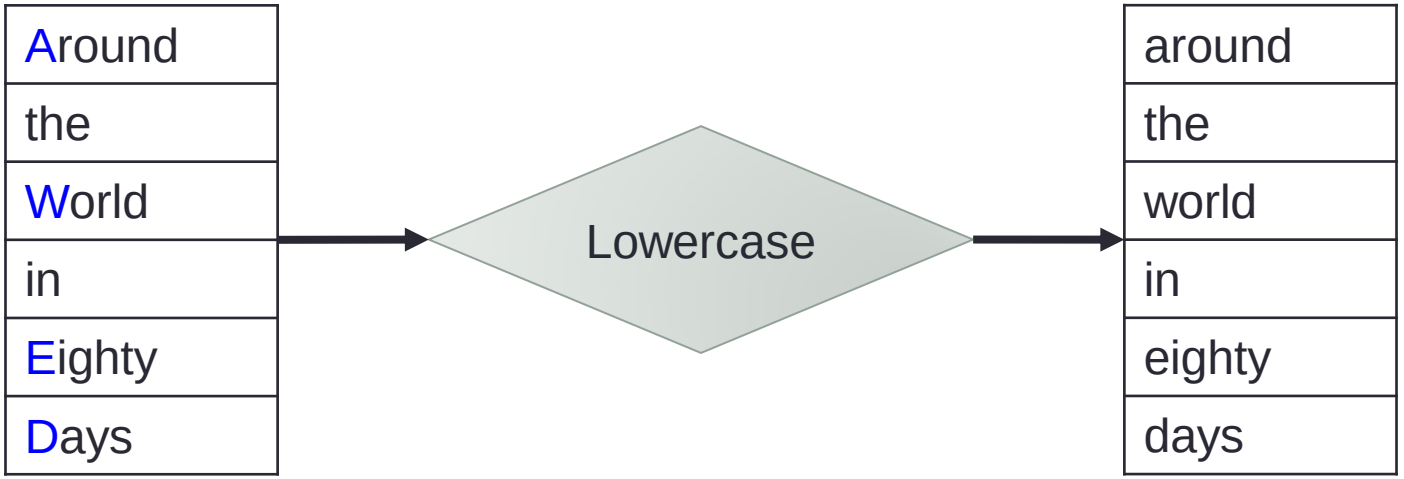
토크나이저(Toknizer)

- Whitespace 토크나이저 - 공백, 탭, 개행 문자 등을 기준으로 문장 분리.



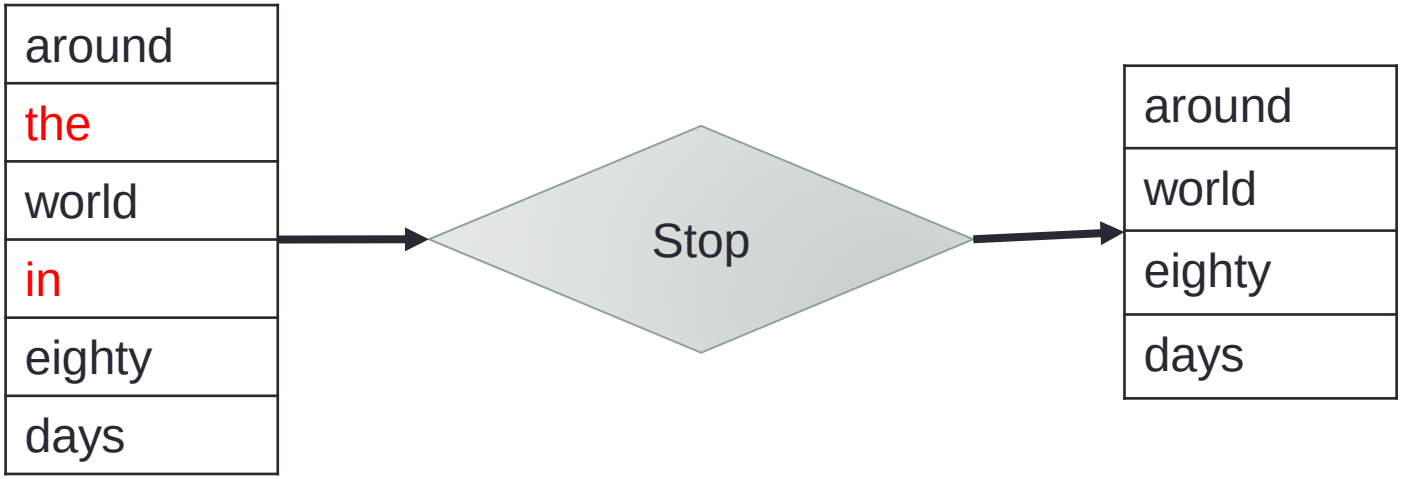
토큰필터(Token Filter)

- Lowercase 토큰필터 - 소문자로 변환



토큰필터(Token Filter)

- Stop 토큰 필터 - stopword 배제



애널라이저 API - _analyze

```
$ curl -XPOST 'http://localhost:9200/books/_analyze?tokenizer=whitespace&filters=lowercase,stop&pretty' -d 'Around the World in Eighty Days'
```

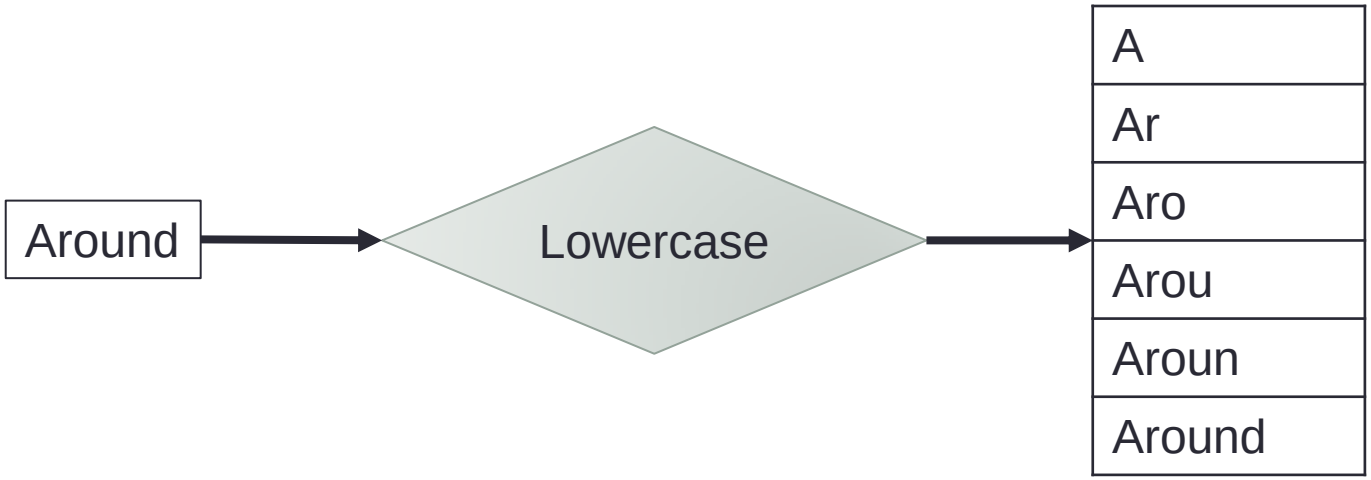
```
{
  "tokens" : [ {
    "token" : "around", "start_offset" : 0, "end_offset" : 6, "type" : "word", "position" : 1
  }, {
    "token" : "world", "start_offset" : 11, "end_offset" : 16, "type" : "word", "position" : 3
  }, {
    "token" : "eighty", "start_offset" : 20, "end_offset" : 26, "type" : "word", "position" : 5
  }, {
    "token" : "days", "start_offset" : 27, "end_offset" : 31, "type" : "word", "position" : 6
  } ]
}
```

사용자 정의 애널라이저

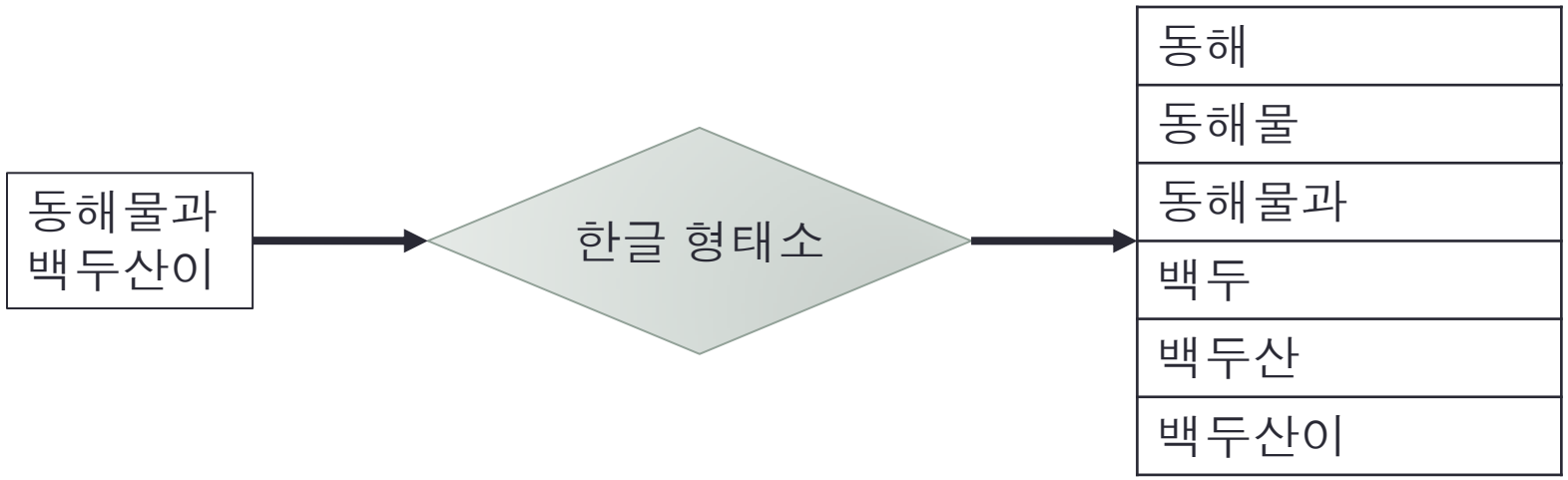
```
curl -XPUT 'http://localhost:9200/books' -d '
{
  "settings" : {
    "analysis" : {
      "analyzer" : {
        "my_analyzer" : {
          "tokenizer" : "whitespace",
          "filter" : [ "lowercase", "stop" ]
        }
      }
    }
  }
}'
```

```
$ curl -XPOST 'localhost:9200/books/_analyze?analyzer=my_analyzer&pretty' -d 'A
round the World in Eighty Days'
```

Ngram 토크나이저



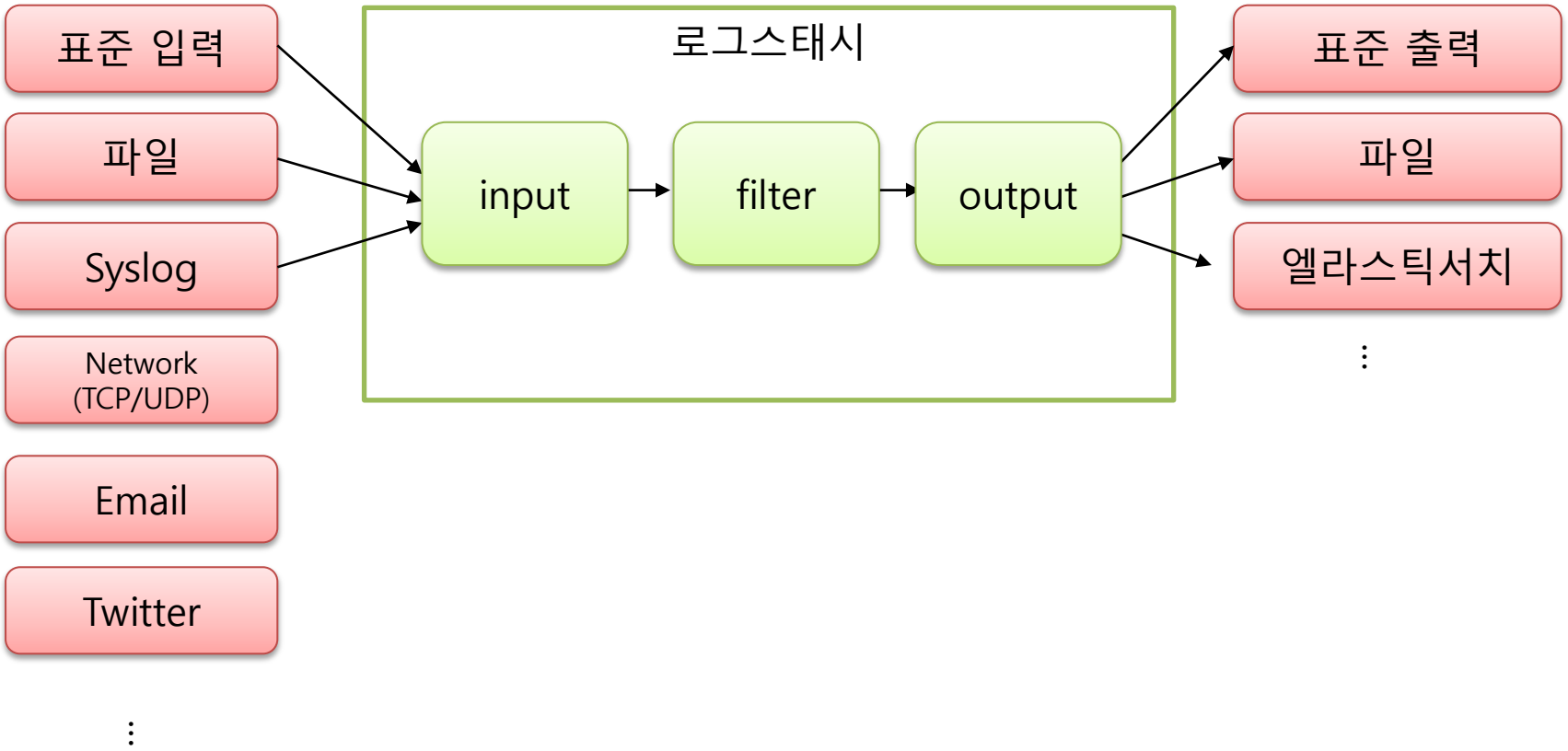
한글 형태소 분석기



엘라스틱서치 사용시 고려사항

- 저장할 데이터 형태와 검색 결과 설계
- 데이터 매핑 구조와 애널라이저 설계
- 저장할 데이터의 유효성 검증
- 원본 데이터 보관c

로그스태시 (Logstash)



로그스태시 (Logstash)

inputs	codecs	filters	outputs
collectd	cloudtrail	advisor	boundary
drupal_dblog	collectd	alter	circonus
elasticsearch	compress_spooler	anonymize	cloudwatch
eventlog	dots	checksum	csv
exec	edn	cidr	datadog
file	edn_lines	cipher	datadog_metrics
ganglia	fluent	clone	elasticsearch
gelf	graphite	collate	elasticsearch_http
gemfire	json	csv	elasticsearch_river
generator	json_lines	date	email
graphite	json_spooler	dns	exec
heroku	line	drop	file

로그스태시 (Logstash)

- 표준 입력 → 표준 출력

standard.conf

```
input {
  stdin { }
}

output {
  stdout { }
}
```

$ bin/logstash -f standard.conf

Hello World
2015-06-01T07:19:27.594Z Jongminui-MacBook-Pro.local Hello World

로그스태시 (Logstash)

- 표준 입력 → 표준 출력 {codec => json}

standard.conf

```
input {
  stdin { }
}

output {
  stdout { codec => json }
}
```

Hello World

{"message":"Hello World","@version":"1","@timestamp":"2015-06-01T07:21:33.876Z","host":"Jongminui-MacBook-Pro.local"}

로그스태시 (Logstash)

- 표준 입력 {codec => json} → 표준 출력 {codec => json}

standard.conf

```
input {
  stdin { codec => json }
}

output {
  stdout { codec => json }
}
```

{ "name":"Jongmin Kim", "age":35 }

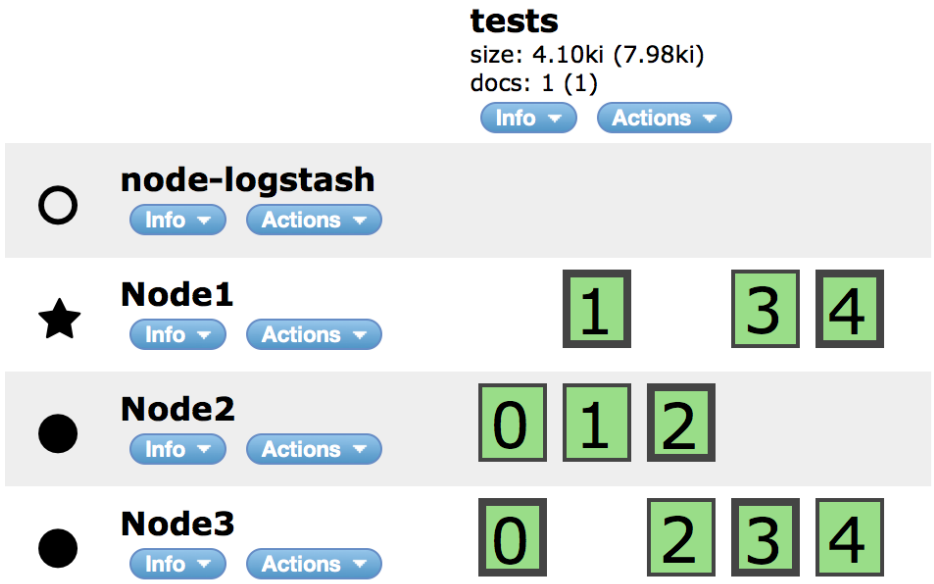
{"name":"Jongmin Kim","age":35,"@version":"1","@timestamp":"2015-06-01T07:25:52.784Z","host":"Jongminui-MacBook-Pro.local"}

로그스태시 (Logstash)

- 엘라스틱서치 출력

elasticsearch.conf

```
output {
  elasticsearch {
    cluster => "elasticsearch"
    node_name => "node-logstash"
    index => "tests"
    document_type => "test-
%{+YYYY.MM.dd}"
    id => "%{id}"
  }
}
```



로그스태시 (Logstash)

{ "id":"kimjmin", "name":"Jongmin Kim", "age":35 }

curl localhost:9200/tests/_search?pretty

…
"hits" : [{
"_index" : "tests",
"_type" : "test-2015.06.02",
"_id" : "kimjmin",
"_score" : 1.0,
"_source":{"id":"kimjmin","name":"Jongmin Kim","age":35,"@version":"1","@timestamp":"2015-06-02T08:44:47.877Z","host":"Jongminui-MacBook-Pro.local"}
}]

로그스태시 (Logstash)

- 파일 입력

standard.conf

```
input {
  file {
    codec => json
    path => "/Users/kimjmin/git/elastic-demo/data/*.log"
  }
}
```

로그스태시 (Logstash) - Filter

- 입력 데이터를 분해, 추가, 삭제, 변형 등의 과정을 거친 뒤 출력으로 전송
- grok, mutate, date …
- 입력한 순서 대로 위에서 부터 차례대로 적용됨

로그스태시 (Logstash)

- 공통
  - add_field => { "comment" => "My name is %{name}" }
  - remove_field => ["name", "age"]
- grok
  - match => { "message" => "Duration: %{NUMBER:duration}" }
- mutate
  - convert => { “age" => "integer" }
  - lowercase => ["name"]
  - split => { "fieldname" => "," }

로그스태시 (Logstash)

- grok

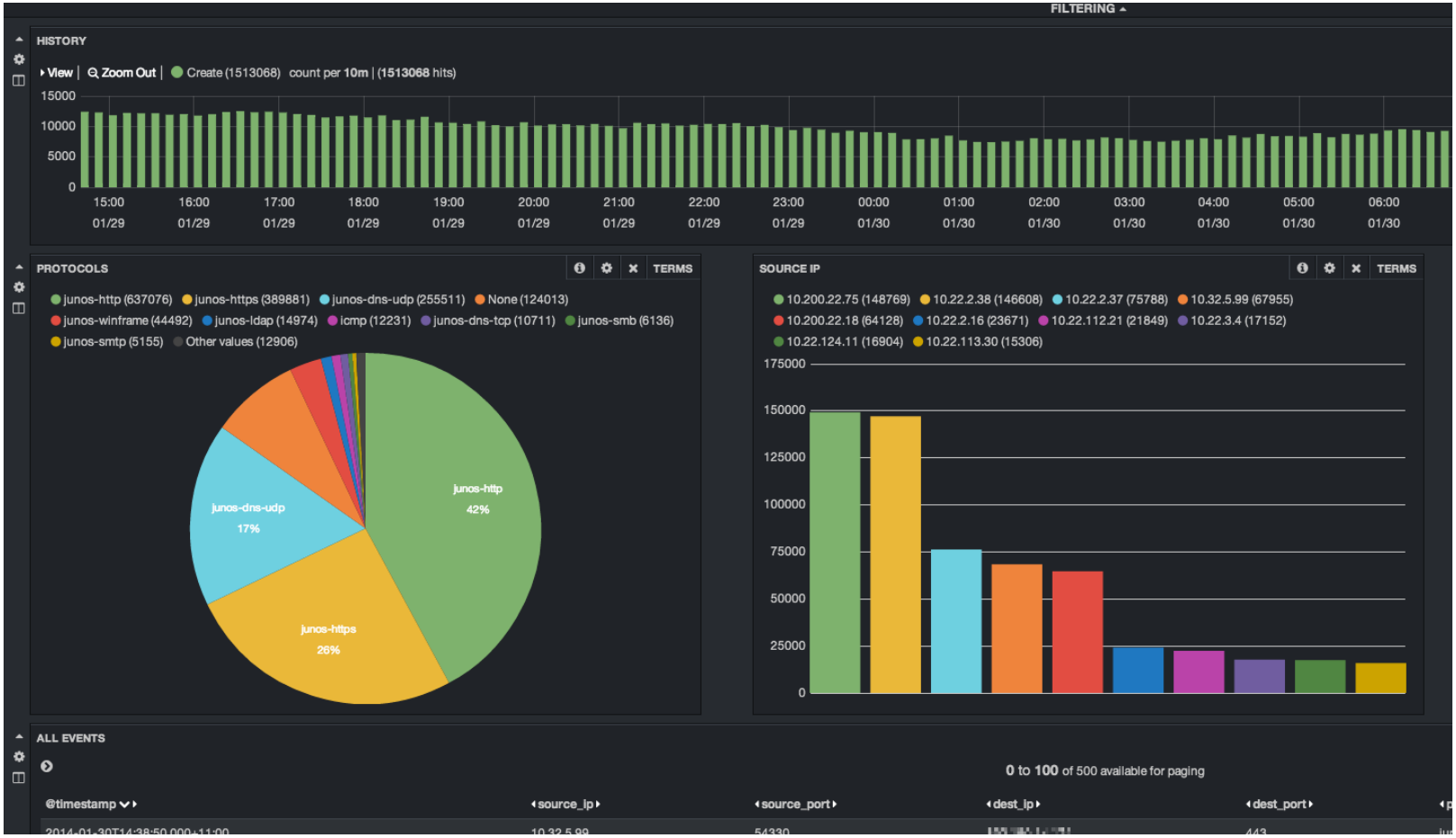
filter.conf

```
filter {
  grok {
    match => {
      "message" =>
"%{IP:client} %{WORD:method} %{
URIPATHPARAM:request} %{NUMB
ER:bytes} %{NUMBER:duration}"
    }
  }
}
```

55.3.244.1 GET /index.html 15824 0.0 43

{"message":"55.3.244.1 GET /index.html 15824 0.043","@version":"1","@timestamp":"2015-06-03T05:25:30.529Z","host":"Jongminui-MacBook-Pro.local","client":"55.3.244.1","method":"GET","request":"/index.html","bytes":"15824","duration":"0.043"}

키바나 (Kibana) - 3


FILTERING
HISTORY
View | Zoom Out | Create (1513068) count per 10m | (1513068 hits)
15000
10000
5000
0
15:00 01/29
16:00 01/29
17:00 01/29
18:00 01/29
19:00 01/29
20:00 01/29
21:00 01/29
22:00 01/29
23:00 01/29
00:00 01/30
01:00 01/30
02:00 01/30
03:00 01/30
04:00 01/30
05:00 01/30
06:00 01/30
PROTOCOLS
TERMS
junos-http (637076)
junos-https (389881)
junos-dns-udp (255511)
None (124013)
junos-winframe (44492)
junos-ldap (14974)
icmp (12231)
junos-dns-tcp (10711)
junos-smb (6136)
junos-smtp (5155)
Other values (12906)
junos-http 42%
junos-dns-udp 17%
junos-https 26%
SOURCE IP
TERMS
10.200.22.75 (148769)
10.22.2.38 (146608)
10.22.2.37 (75788)
10.32.5.99 (67955)
10.200.22.18 (64128)
10.22.2.16 (23671)
10.22.112.21 (21849)
10.22.3.4 (17152)
10.22.124.11 (16904)
10.22.113.30 (15306)
175000
150000
125000
100000
75000
50000
25000
0
ALL EVENTS
0 to 100 of 500 available for paging
@timestamp
source_ip
source_port
dest_ip
dest_port

키바나 (Kibana) - 3

- Only HTML, Javascript (AngularJS)
  - 클라이언트에서 실행
  - 9200 포트 개방 필요 - 보안에 취약
- 별도 웹서버 필요. Tomcat, Nginx 등.
- Facet based

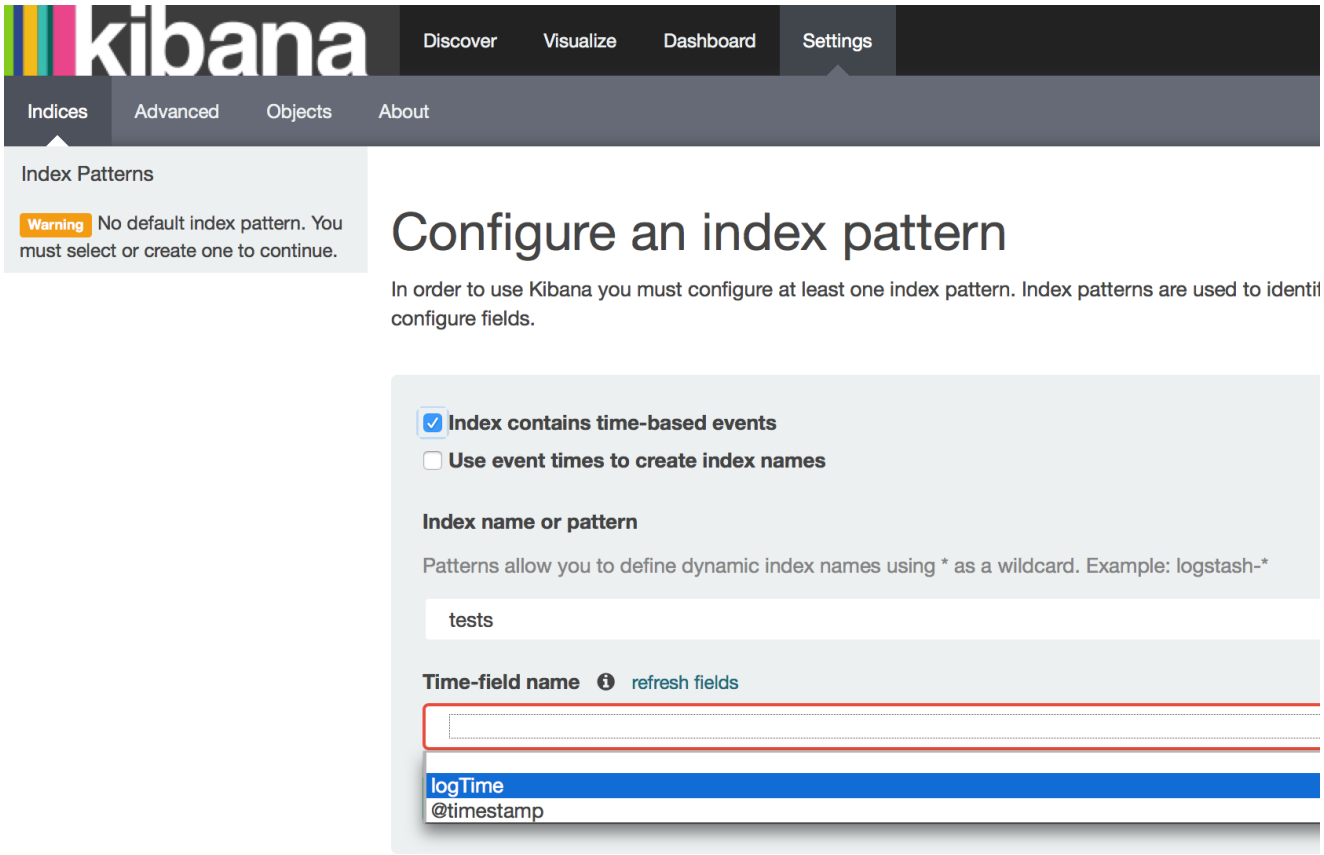
키바나 (Kibana) - 4

- Elasticsearch 1.4.4 이상 필요.
- NodeJS 서버 사용 - port : 5601
- Aggregation based.

bin/kibana

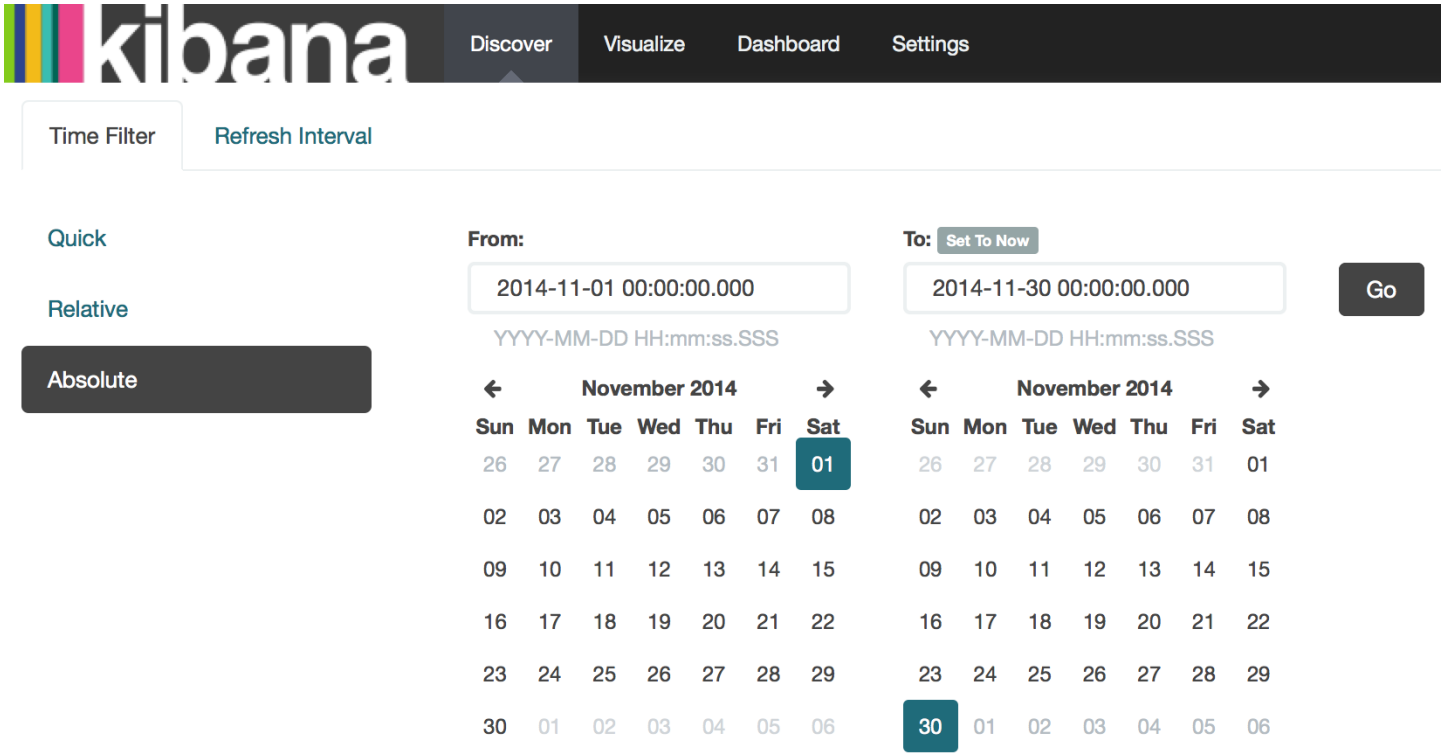
키바나 (Kibana) - 4

- 기준 index, time-field 설정.

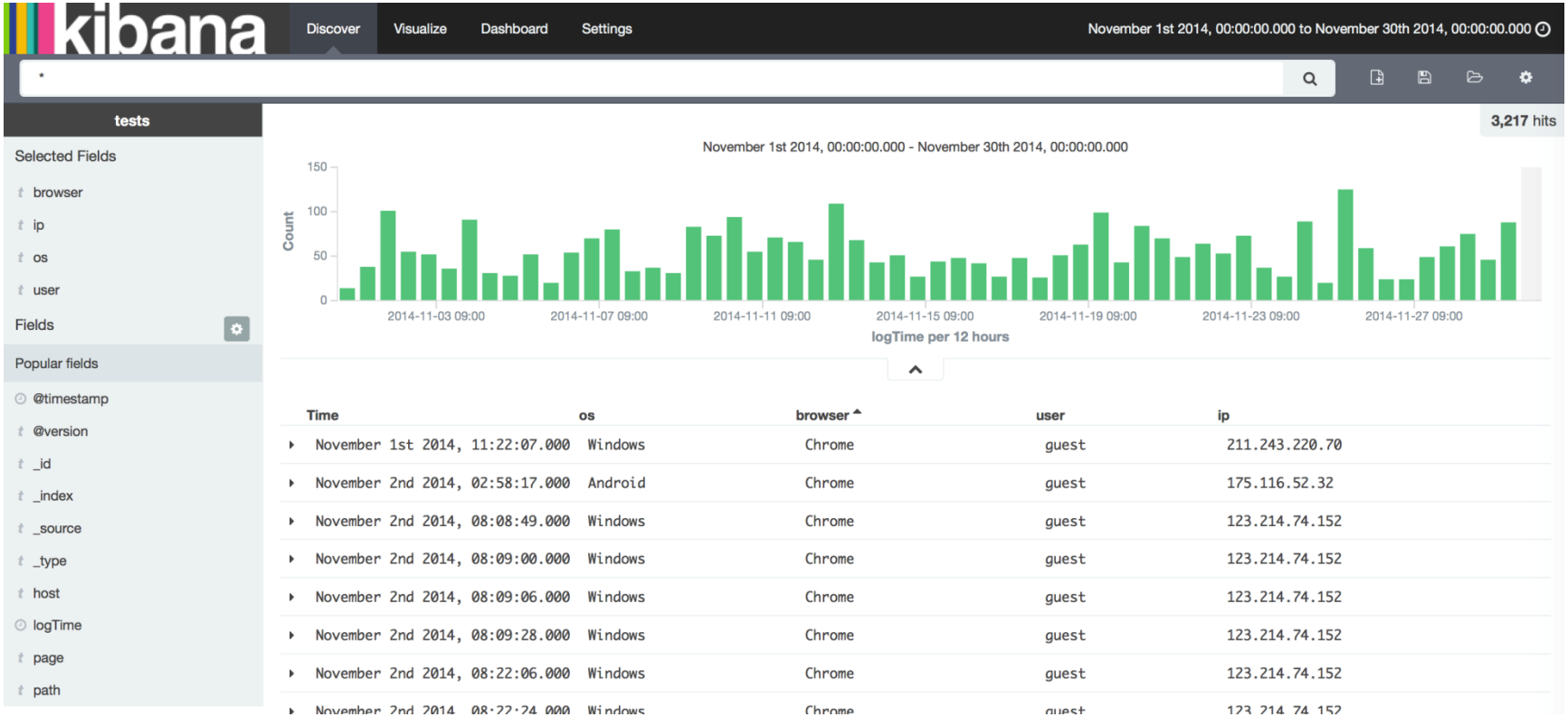


키바나 (Kibana) - 4

- Discover → Time Filter 설정



키바나 (Kibana) - 4


kibana
Discover
Visualize
Dashboard
Settings
November 1st 2014, 00:00:00.000 to November 30th 2014, 00:00:00.000
*
tests
Selected Fields
browser
ip
os
user
Fields
Popular fields
@timestamp
@version
_id
_index
_source
_type
host
logTime
page
path
3,217 hits
November 1st 2014, 00:00:00.000 - November 30th 2014, 00:00:00.000
Count
150
100
50
0
2014-11-03 09:00
2014-11-07 09:00
2014-11-11 09:00
2014-11-15 09:00
2014-11-19 09:00
2014-11-23 09:00
2014-11-27 09:00
logTime per 12 hours
Time
os
browser
user
ip
November 1st 2014, 11:22:07.000 Windows Chrome guest 211.243.220.70
November 2nd 2014, 02:58:17.000 Android Chrome guest 175.116.52.32
November 2nd 2014, 08:08:49.000 Windows Chrome guest 123.214.74.152
November 2nd 2014, 08:09:00.000 Windows Chrome guest 123.214.74.152
November 2nd 2014, 08:09:06.000 Windows Chrome guest 123.214.74.152
November 2nd 2014, 08:09:28.000 Windows Chrome guest 123.214.74.152
November 2nd 2014, 08:22:06.000 Windows Chrome guest 123.214.74.152

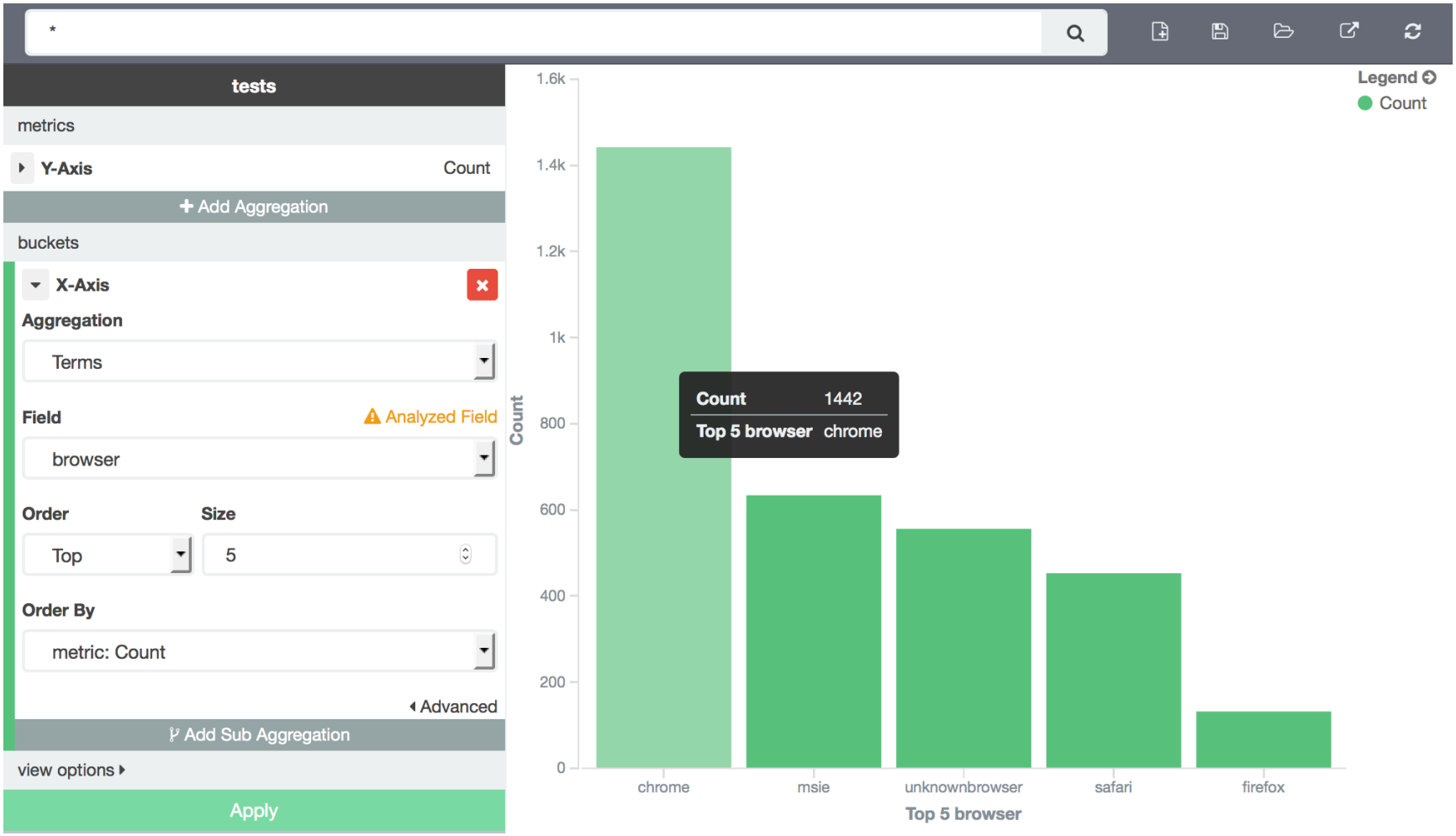
키바나 (Kibana) - 4

Create a new visualization

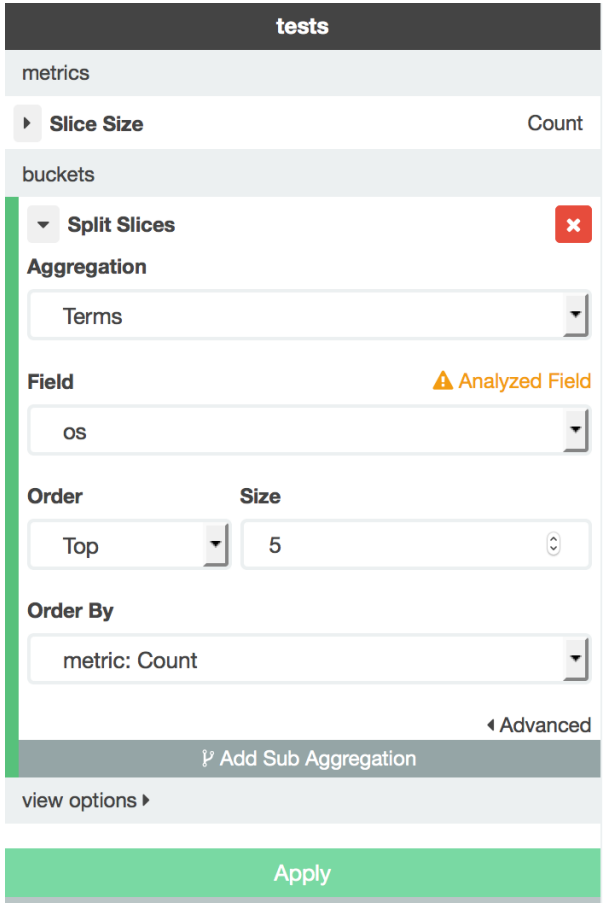
Step 1

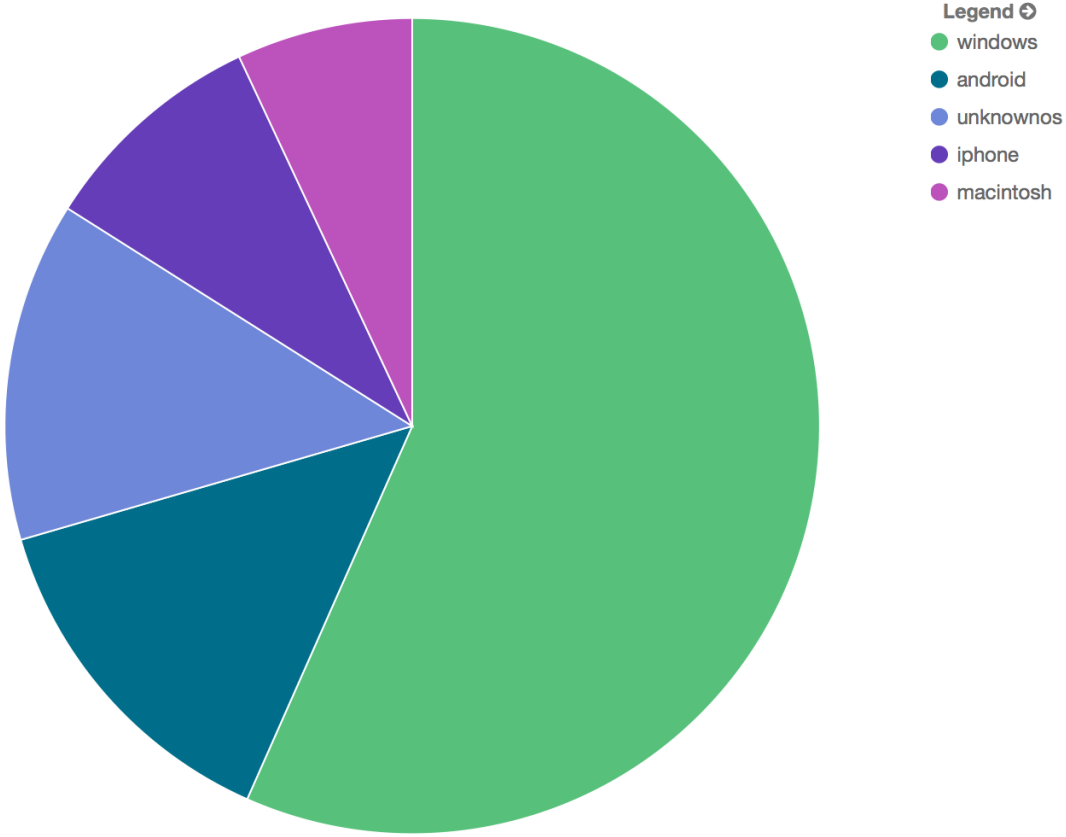
Area chart	Great for stacked timelines in which the total of all series is more important than comparing any two or more series. Less useful for assessing the relative change of unrelated data points as changes in a series lower down the stack will have a difficult to gauge effect on the series above it.
Data table	The data table provides a detailed breakdown, in tabular format, of the results of a composed aggregation. Tip, a data table is available from many other charts by clicking grey bar at the bottom of the chart.
Line chart	Often the best chart for high density time series. Great for comparing one series to another. Be careful with sparse sets as the connection between points can be misleading.
Markdown widget	Useful for displaying explanations or instructions for dashboards.
Metric	One big number for all of your one big number needs. Perfect for show a count of hits, or the exact average a numeric field.
Pie chart	Pie charts are ideal for displaying the parts of some whole. For example, sales percentages by department.Pro Tip: Pie charts are best used sparingly, and with no more than 7 slices per pie.
Tile map	Your source for geographic maps. Requires an elasticsearch geo_point field. More specifically, a field that is mapped as type:geo_point with latitude and longitude coordinates.
Vertical bar chart	The goto chart for oh-so-many needs. Great for time and non-time data. Stacked or grouped, exact numbers or percentages. If you are not sure which chart your need, you could do worse than to start here.

키바나 (Kibana) - 4

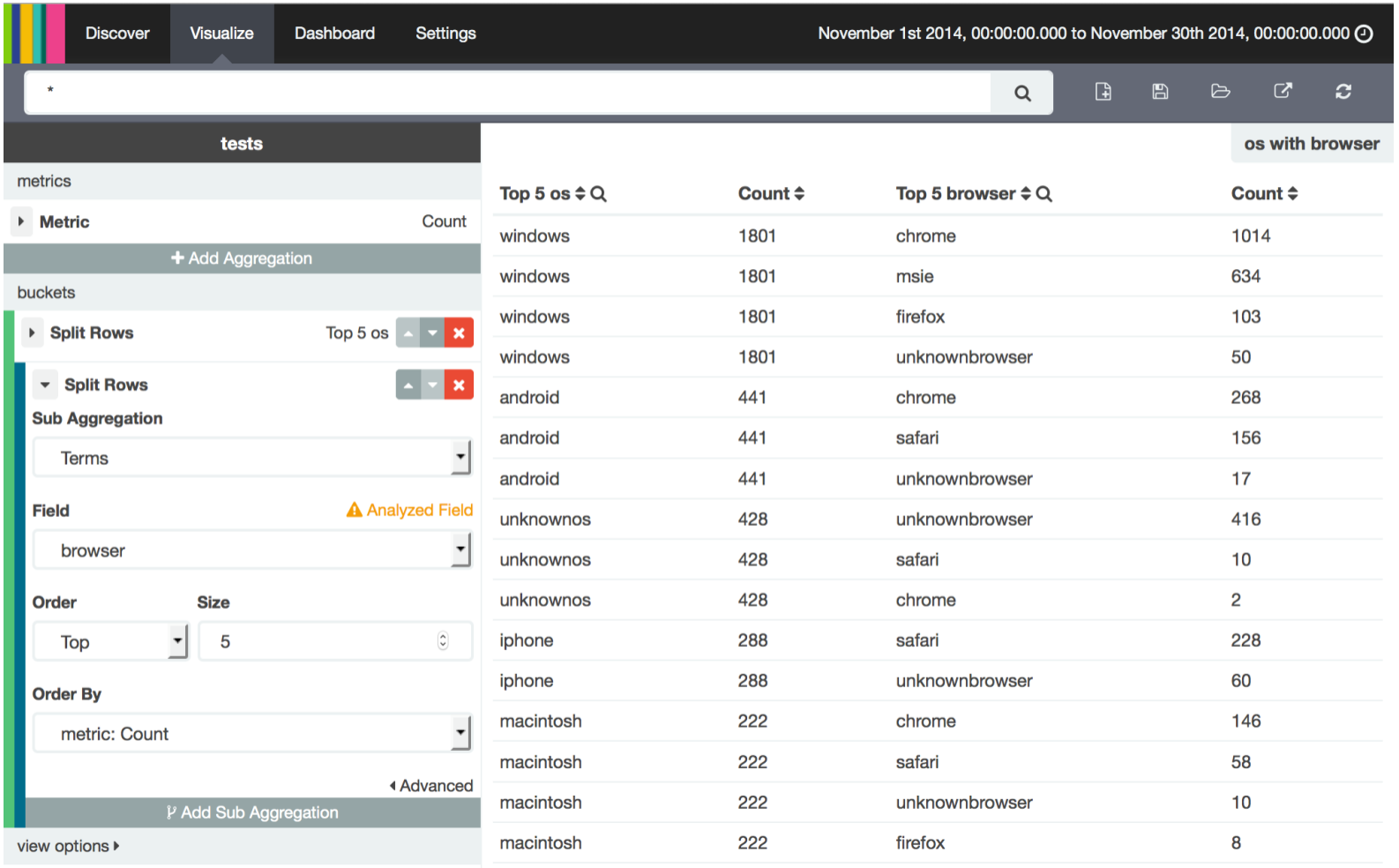

*
tests
metrics
Y-Axis
Count
+ Add Aggregation
buckets
X-Axis
Aggregation
Terms
Field
Analyzed Field
browser
Order
Size
Top
5
Order By
metric: Count
Advanced
Add Sub Aggregation
view options
Apply
Legend
Count
1.6k
1.4k
1.2k
1k
800
600
400
200
0
Count
Count 1442
Top 5 browser chrome
chrome
msie
unknownbrowser
safari
firefox
Top 5 browser

키바나 (Kibana) - 4


tests
metrics
Slice Size
Count
buckets
Split Slices
Aggregation
Terms
Field
Analyzed Field
os
Order
Size
Top
5
Order By
metric: Count
Advanced
Add Sub Aggregation
view options
Apply


os pie chart
Legend
windows
android
unknownos
iphone
macintosh

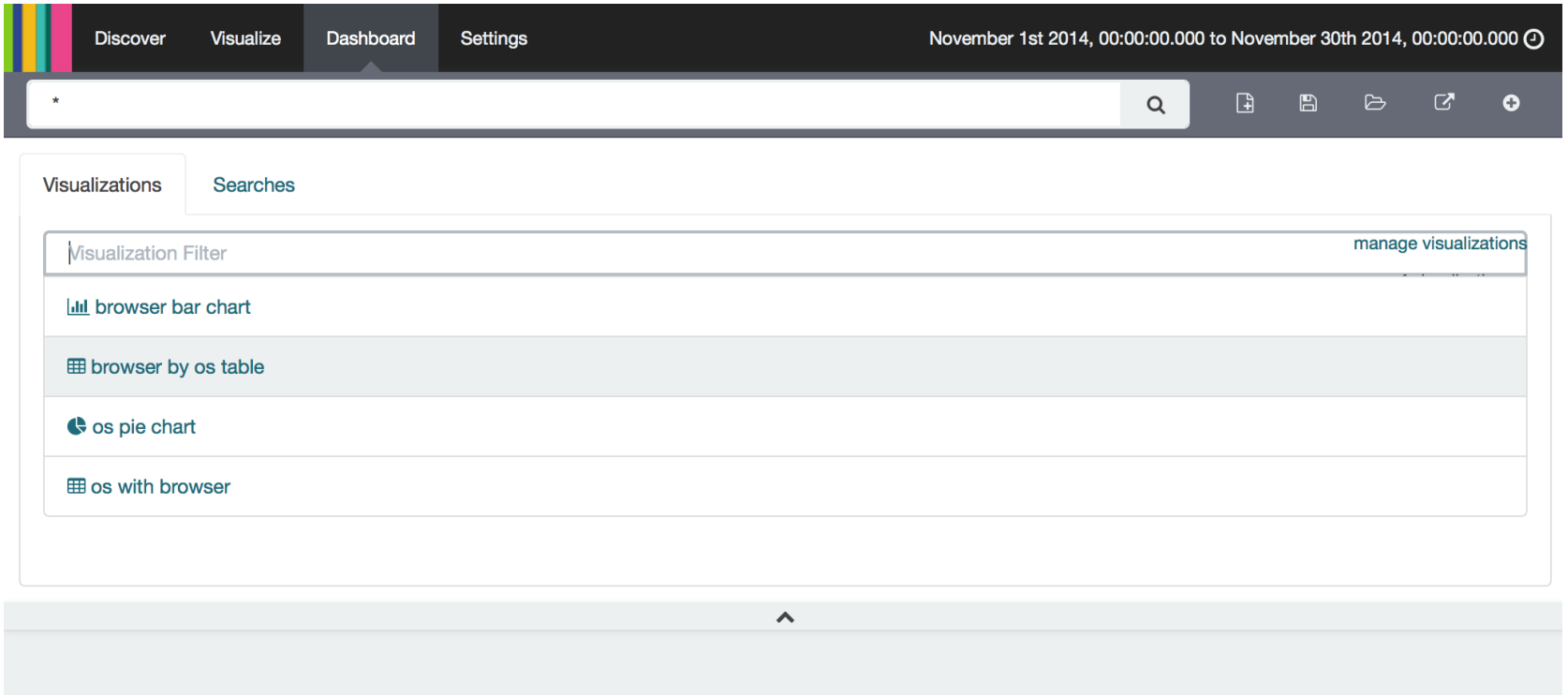
키바나 (Kibana) - 4



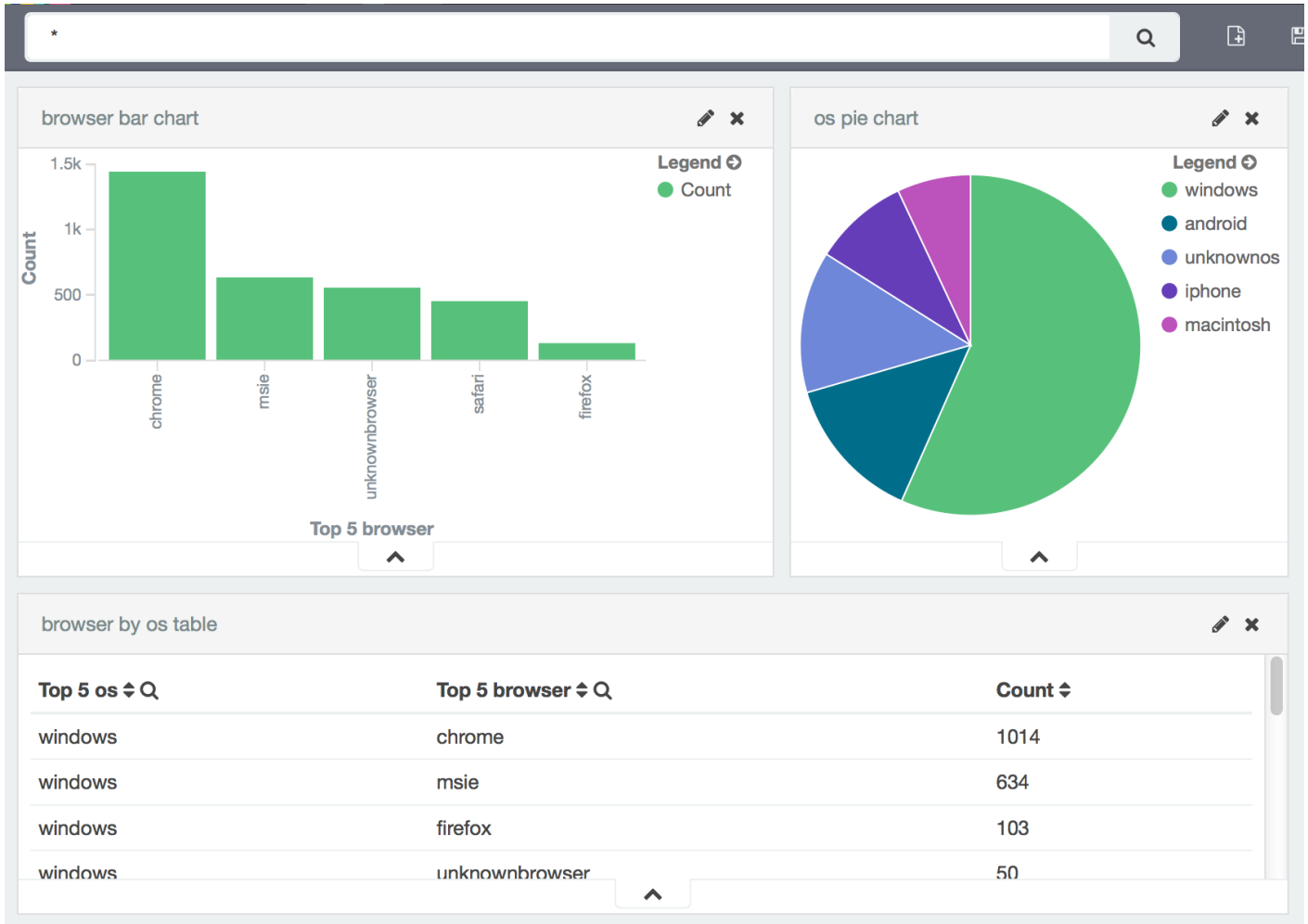
Top 5 os	Count	Top 5 browser	Count
windows	1801	chrome	1014
windows	1801	msie	634
windows	1801	firefox	103
windows	1801	unknownbrowser	50
android	441	chrome	268
android	441	safari	156
android	441	unknownbrowser	17
unknownos	428	unknownbrowser	416
unknownos	428	safari	10
unknownos	428	chrome	2
iphone	288	safari	228
iphone	288	unknownbrowser	60
macintosh	222	chrome	146
macintosh	222	safari	58
macintosh	222	unknownbrowser	10
macintosh	222	firefox	8

키바나 (Kibana) - 4

- Visualize 탭에서 미리 저장한 시각 도구를 가지고 Dashboard 탭에서 대시보드 작성.



키바나 (Kibana) - 4


*
browser bar chart
1.5k
1k
500
0
Count
chrome
msie
unknownbrowser
safari
firefox
Top 5 browser
Legend
Count
os pie chart
Legend
windows
android
unknownos
iphone
macintosh
browser by os table
Top 5 os
Top 5 browser
Count
windows chrome 1014
windows msie 634
windows firefox 103
windows unknownbrowser 50

감사합니다