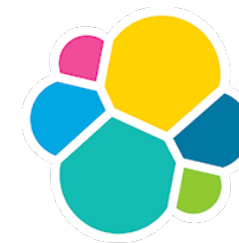


# Elastic Stack을 활용한 Data Dashboard 만들기

Week 6 - 전체 Review



Fast Campus

지금까지 Elastic Stack에 대해 배우긴 했는데  
실제로 처음부터 끝까지 어떻게 어떤 순서로 작업해야 되는지 모르겠다.  
지금까지 배운 걸 어떻게 이용하면 되는지 **순서대로** 살펴보자.

목차	페이지
Server 준비	4
Elastic Stack 설치 및 실행	6
Elastic Stack Workflow	7
Mapping	9
Logstash 실행	11
Index 등록	12
Discover	13
Visualize	14
Aggregation	16
Dashboard	20
Filter	21
Search	25
Query DSL	28

# Elastic Stack을 실행할 Server가 필요하다

(Elasticsearch node 1대 기준)

	권장 👑	수업 실습 서버
Memory	8GB - 64GB	8GB
CPUs	2 - 8 core	2 core
Disk	SSD	SSD

위 권장을 참고해서 운영 환경에 맞게 Elastic Stack Architecture를 설계하고 운영하자

## 적절한 EC2 Instance를 선택하고 생성한 후에 접속하자 🏰

```
__|  __|_ )  
_| (  /  Amazon Linux AMI  
__|\__|__|
```

```
https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/  
15 package(s) needed for security, out of 21 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-21-251 ~]$
```

# Elastic Stack을 설치/실행하자 👑

1. tar.gz 방법 👑

2. docker 방법 👑

⋮

---

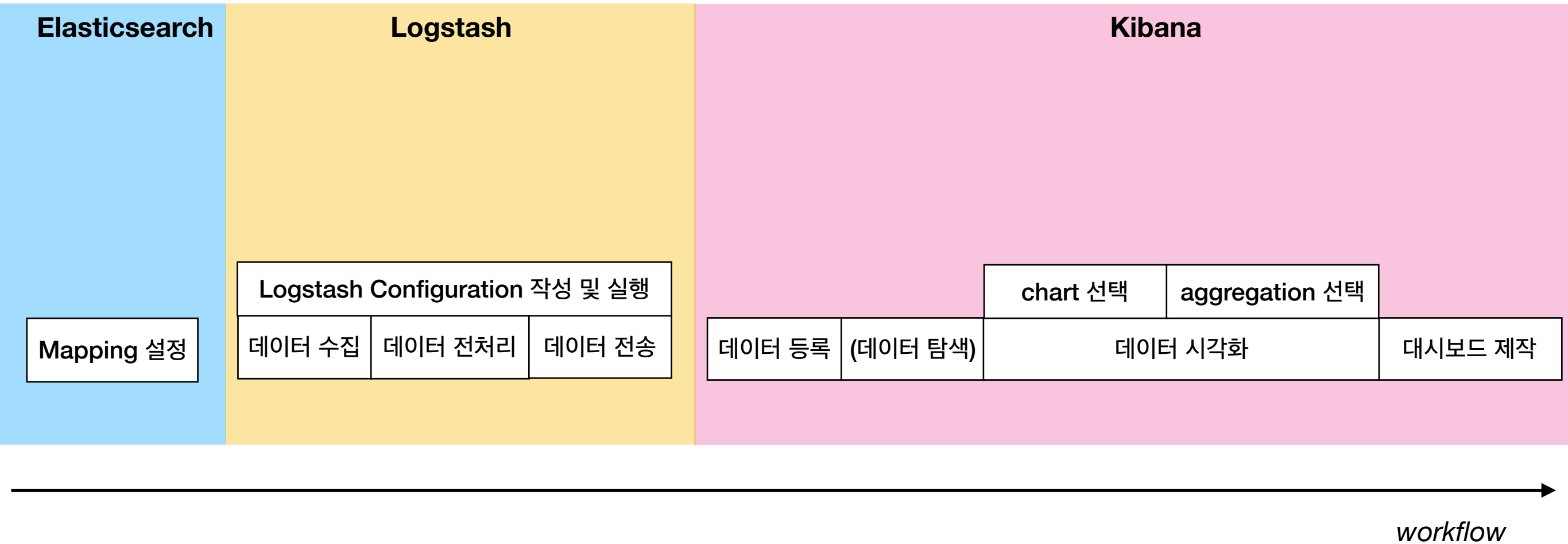
여러가지 설치법 중 환경에 맞는 걸 선택해서 사용하자

**지금까지기 Elastic Stack을 활용하기 위한 준비단계였다면,**

**이제는 본격적으로 Elastic Stack을 활용할 차례다.**

**담당자는 다음장의 workflow 대로 작업 할 수 있다.**

# Elastic Stack Workflow





# 1. Mapping 설정

## 1. Index 구성

- 단발성 Index : shopping
- 정기적 Index : shopping-2018.02.13, shopping-2018.02.14, ...

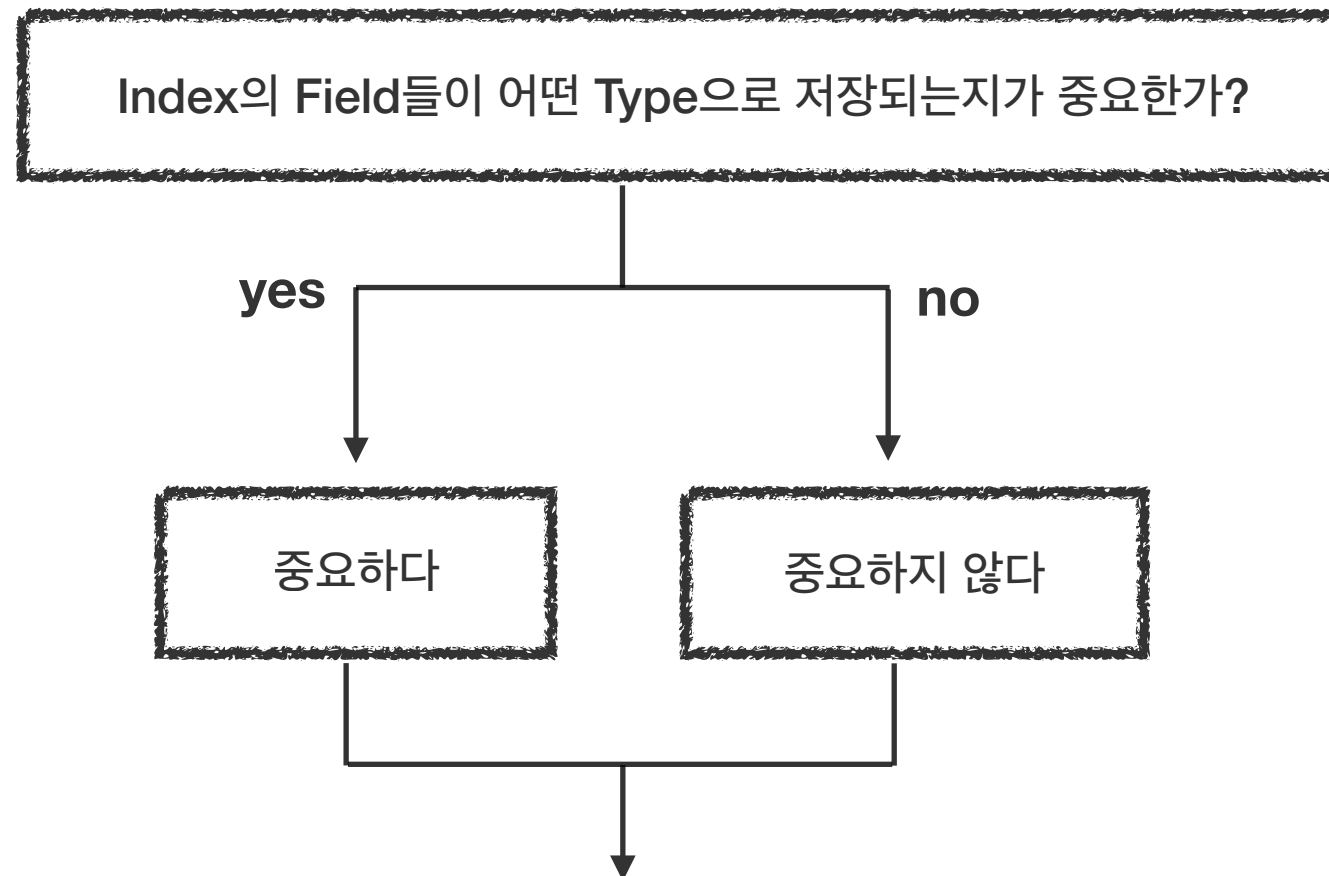
## 2. Document 구성

- 각 Document를 구성할 Field(s) 정하기
- 예시 : 접수날짜, 상품가격, 상품분류, 결제카드, ...

## 3. Field Data Type 설정

- 각 Field에 적절한 Data Type으로 설정
- 예시
  - 접수날짜 : date
  - 상품가격 : integer
  - 상품분류 : keyword

# 1. Mapping 설정은 꼭 필요한가?




- Mapping이 없어도 에러가 발생하지는 않는다
- 다만 사용자가 원하는 Data Type으로 데이터가 저장된다는 보장이 없다. 예) “2017-01-01 13:00:00”
- 그러므로 (색인 전에) 가급적 Mapping을 설정하는 걸 권장한다

## 2. Logstash Configuration 작성 및 실행 👑

- |       |   |                                    |
|-------|---|------------------------------------|
| 문제 정의 | [ | 1. 데이터가 어디에 있는지 (=input) 명확히 한다.   |
|       | — | 2. 데이터를 어디로 전송할지 (=output) 명확히 한다. |
|       | ] | 3. 데이터를 어떻게 변형할지 (=filter) 명확히 한다. |
| 코드 작성 | [ | 4. 적절한 input plugin 선택             |
|       | — | 5. 적절한 output plugin 선택            |
|       | ] | 6. 적절한 filter plugin 선택            |

### 3. Index 등록 👑



Discover  
Visualize  
Dashboard  
Timelion  
Dev Tools  
Management

Collapse

Management / Kibana  
Index Patterns   Saved Objects   Advanced Settings

#### Configure an index pattern

In order to use Kibana you must configure at least one index pattern. Index patterns are used to identify the Elasticsearch index to run search and analytics against. They are also used to configure fields.

**Index name or pattern**

**Time Filter field name** ⓘ [refresh fields](#)

☐ Expand index pattern when searching [DEPRECATED]

With this option selected, searches against any time-based index pattern that contains a wildcard will automatically be expanded to query only the indices that contain data within the currently selected time range.

Searching against the index pattern `logstash-*` will actually query Elasticsearch for the specific matching indices (e.g. `logstash-2015.12.21`) that fall within the current time range.

With recent changes to Elasticsearch, this option should no longer be necessary and will likely be removed in future versions of Kibana.

☐ Use event times to create index names [DEPRECATED]

Time Filter field name is required

## 4. 데이터 탐색 🏰

The screenshot displays the Kibana search interface. On the left is a blue sidebar with navigation links: Discover, Visualize, Dashboard, Timelion, Management, and Dev Tools. The main area is divided into several sections. At the top, the 'Index Pattern' is set to 'logstash-\*' and the 'Query bar' contains an asterisk '\*'. The 'Time Picker' shows a range from 'May 17th 2015, 04:00:41.685 to May 20th 2015, 18:32:51.964'. Below the query bar, a 'Toolbar' contains buttons for 'New', 'Save', 'Open', and 'Share'. The main visualization area shows a 'Histogram' of 'Count' vs 'utc\_time per hour'. Below the histogram is a 'Document Table' with two rows of search results. The first row shows a document from May 18th 2015, 02:03:25.877 with fields like @timestamp, ip, extension, response, geo.coordinates, geo.src, geo.dest, geo.srcdest, @tags, utc\_time, and referer. The second row shows a document from May 18th 2015, 05:28:25.013 with similar fields.

Index Pattern

Query bar

Time Picker

14,005 hits

New Save Open Share

May 17th 2015, 04:00:41.685 to May 20th 2015, 18:32:51.964

Discover

Visualize

Dashboard

Timelion

Management

Dev Tools

logstash-\*

Selected Fields

? \_source

Available Fields

Popular

t @message

t extension

ip

t machine.os

t response

t url

t @tags

@timestamp

? @version

t \_id

t \_index

# \_score

t \_type

t agent

Count

May 17th 2015, 04:00:41.685 - May 20th 2015, 18:32:51.964 — Hourly

2015-05-17 17:00 2015-05-18 17:00 2015-05-19 17:00

utc\_time per hour

Time

\_source

May 18th 2015, 02:03:25.877

@timestamp: May 18th 2015, 02:03:25.877 ip: 185.124.182.126 extension: gif response: 404 geo.coordinates: { "lat": 36.518375, "lon": -86.05828083 } geo.src: PH geo.dest: MM geo.srcdest: PH:MM @tags: success, info utc\_time: May 18th 2015, 02:03:25.877 referer: http://twitter.com/error/will

May 18th 2015, 05:28:25.013

@timestamp: May 18th 2015, 05:28:25.013 ip: 79.1.14.87 extension: gif response: 200 geo.coordinates: { "lat": 35.16531472, "lon": -107.9006142 } geo.src: GN geo.dest: US geo.srcdest: GN:US @tags: success, info utc\_time: May 18th 2015, 05:28:25.013 referer: http://www.slate.com/warning/

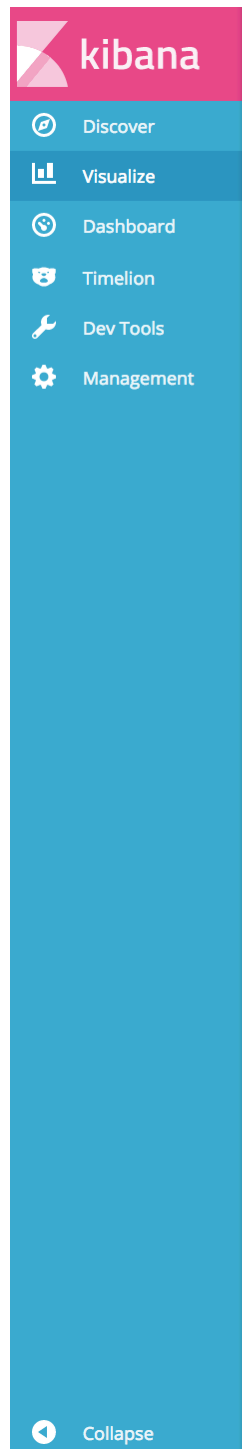
Side Navigation

Toolbar

Histogram

Document Table

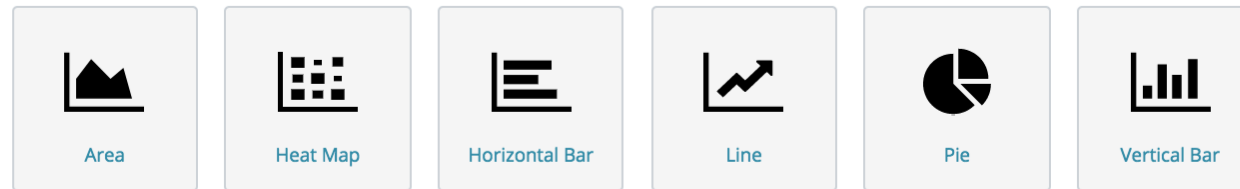
## 5. Visualize - Chart 선택 👑



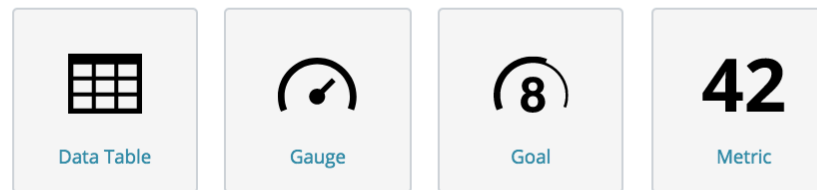
Select visualization type

🔍 Search visualization types...

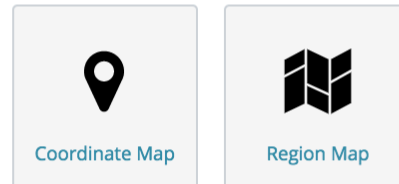
Basic Charts



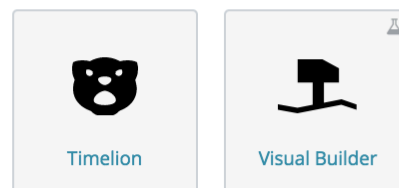
Data



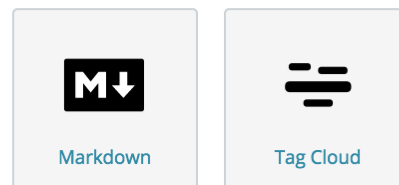
Maps



Time Series



Other



## 5. Visualize - Aggregation 선택 👑










1. 문제에서 **metrics** 영역과 **buckets** 영역으로 구분한다
2. **metrics**와 **buckets** 내에서 사용할 aggregation을 선택한다
3. term aggregation으로 **bucket**을 나눌 경우 sorting을 위한 aggregation을 정의한다

# Metric Aggregation




종류	적용 가능 Type	상세
Avg 	Number	(Bucket 내) Document의 특정 Field의 평균 계산
Sum 	Number	(Bucket 내) Document의 특정 Field의 합 계산
Min/Max 	Number	(Bucket 내) Document의 특정 Field의 최소/최대 계산
Extended Stats 	Number	(Bucket 내) Document의 특정 Field의 기초 통계값 계산
Cardinality 	Number	(Bucket 내) Document의 특정 Field의 고유한 개수 계산
Percentiles 	Number	(Bucket 내) Document의 특정 Field의 백분위수 계산
Percentiles Ranks 	Number	(Bucket 내) Document의 특정 Field의 백분위 계산
Top Hits 	All	(Bucket 내) 특정 조건을 만족하는 Documents의 특정 Field의 Agg 반환
Value Count 	All	(Bucket 내) Document의 개수 계산, Kibana 상에서 default metric aggregation



# Bucket Aggregation

종류	적용 가능 Type	상세
Date 	Date	날짜/시간을 일정하게 지정하여 구간 내의 Documents로 Bucket 형성
Date Range 	Date	날짜/시간을 임의로 지정하여 구간 내의 Documents로 Bucket 형성
Histogram 	Number	범위를 일정하게 지정하여 구간 내의 Documents로 Bucket 형성
Range 	Number	범위를 임의로 지정하여 구간 내의 Documents로 Bucket 형성
Terms 	Date, IP, Number, String	특정 Field 값을 기준으로 Bucket 생성 (카테고리 데이터에 유용)
Significant Terms 	String	Background 대비 Foreground에서 특별한 값으로 Bucket 생성
Filters 	Date, IP, Number, String	직접 조건을 입력하여 Bucket 생성 (조건 개수만큼 Bucket 생성)
Geo Hash 	Geo Point	특정 지점 (Centroid) 근처에 있는 값들을 모아서 Bucket 생성
IPv4 Range 	IP	IP 주소의 범위로 Bucket 생성

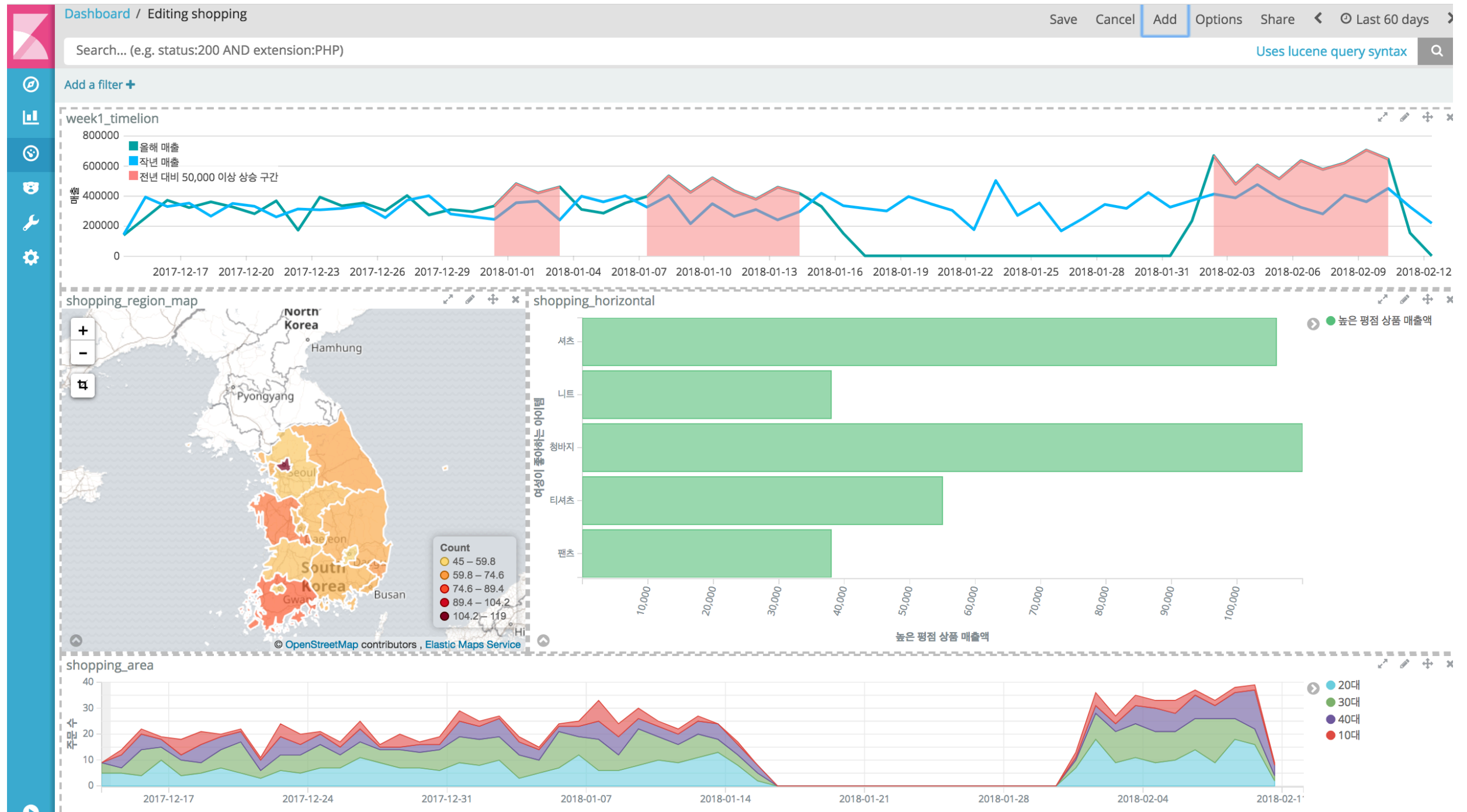
# Parent Pipeline Aggregation

종류	설명
Derivative 	Date Bucket Agg 후, Bucket 내 Metric Agg 하고 난 후, 연속한 Bucket 값들의 차이를 구함
Cumulative Sum 	Date Bucket Agg 후, Bucket 내 Metric Agg 하고 난 후, Bucket 값들의 누적합을 구함
Moving Average 	Date Bucket Agg 후, Bucket 내 Metric Agg 하고 난 후, 연속한 {n개} Bucket 값들의 평균을 구함
Serial Diff 	Date Bucket Agg 후, Bucket 내 Metric Agg 하고 난 후, {n번째 이전} Bucket 과의 차이를 구함

# Sibling Pipeline Aggregation

종류	설명
Min Bucket 	Bucket Agg 후, Bucket 내 Metric Agg 하고 난 후, Min Aggregation 적용
Max Bucket 	Bucket Agg 후, Bucket 내 Metric Agg 하고 난 후, Max Aggregation 적용
Sum Bucket 	Bucket Agg 후, Bucket 내 Metric Agg 하고 난 후, Sum Aggregation 적용
Average Bucket 	Bucket Agg 후, Bucket 내 Metric Agg 하고 난 후, Average Aggregation 적용

## 6. Dashboard



**Dashboard는 만들었는데**  
**원하는 조건의 데이터만 보고 싶으면?**

# Filter를 실행하자

Discover
Visualize
Dashboard
Timelion
Dev Tools
Management

Dashboard / week2
Share Clone Edit < Last 6 months >

Search... (e.g. status:200 AND extension:PHP)
[Uses lucene query syntax](#)

Add a filter +

Filter

Fields...

Label

Optional

Edit Query DSL

Cancel Save

주요

주요

주소

week2\_datatable

날짜	인기 Top 3	매출	매출 증감	매출 누적
07월22일 00시00분	스커트	25,000	-	25,000
07월23일 00시00분	청바지, 팬츠, 티셔츠	399,000	374,000	424,000
07월24일 00시00분	니트, 티셔츠, 코트	284,000	-115,000	708,000
07월25일 00시00분	스웨터, 자켓, 니트	217,000	-67,000	925,000
07월26일 00시00분	자켓, 스웨터, 가디건	211,000	-6,000	1,136,000
07월27일 00시00분	자켓, 니트, 가디건	320,000	109,000	1,456,000
07월28일 00시00분	니트, 니트, 점퍼	330,000	10,000	1,786,000
07월29일 00시00분	셔츠, 스웨터, 가디건	312,000	-18,000	2,098,000

## Filter 사용법을 익히자 👑

The screenshot shows a 'Add filter' dialog box with a close button (X) in the top right corner. The dialog is divided into two main sections: 'Filter' and 'Label'. In the 'Filter' section, there are three input fields: a dropdown menu showing '\_id' (annotated with ①), a dropdown menu showing 'is' (annotated with ②), and a text input field containing 'Value...' (annotated with ③). To the right of these fields is a link that says 'Edit Query DSL'. In the 'Label' section, there is a text input field containing 'Optional' (annotated with ④). At the bottom right of the dialog are two buttons: 'Cancel' and 'Save'.

- ① Filter 적용할 Field 선택
- ② 적용할 Operator 선택 (다음 페이지 참조)
- ③ Filter에 적용하려는 Value 입력
- ④ (여러 Filter 구분하기 위한) 이름 입력

## Operator 설명

Operator	역할
is	Field의 Value가 입력한 값과 일치하는 Documents 선택
is not	Field의 Value가 입력한 값과 일치하지 않는 Documents 선택
is one of	Field의 Value가 입력한 값 중에 존재하는 Documents 선택
is not one of	Field의 Value가 입력한 값 중에 존재하지 않는 Documents 선택
exists	Field가 적어도 한 개의 non-null 값을 가지는 Documents 선택
does not exist	Field가 존재하지 않거나 null 값만 가지는 Documents 선택
is between	Field의 Value가 입력한 값 사이에 존재하는 Documents 검색
is not between	Field의 Value가 입력한 값 사이에 존재하지 않는 Documents 검색

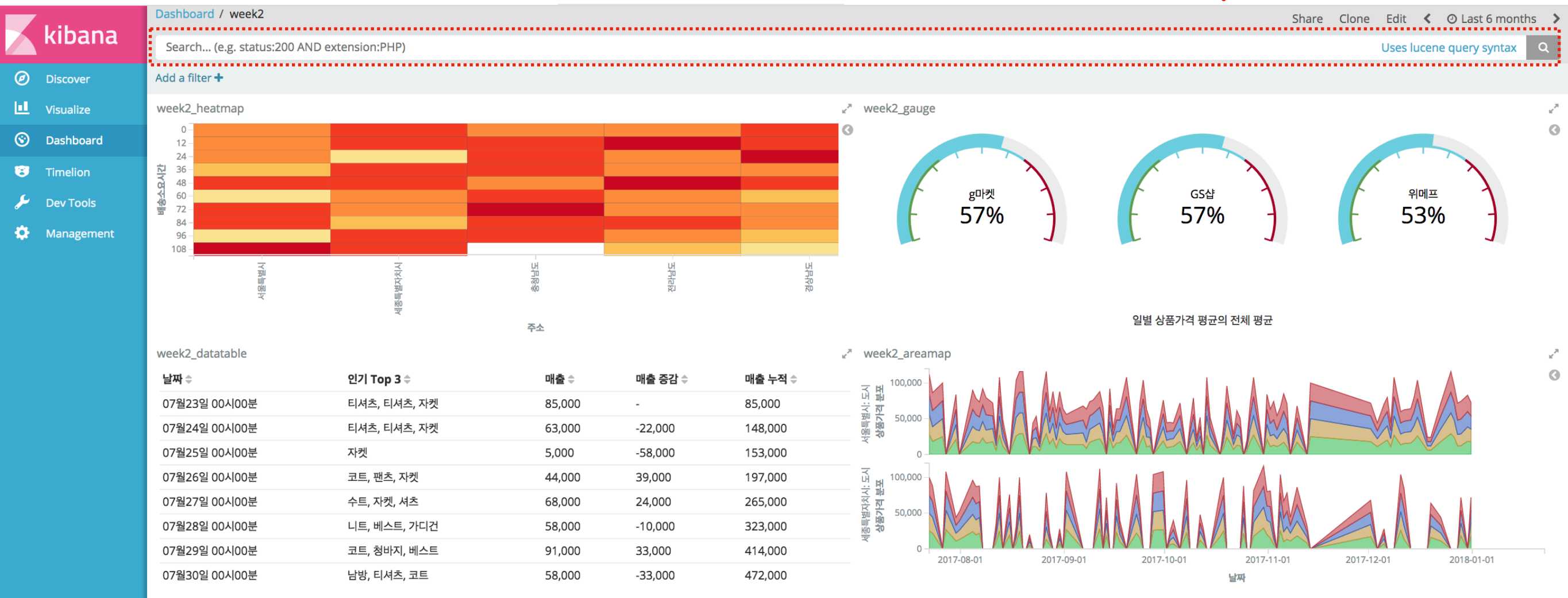


**구글 검색처럼 검색할 수는 없나?**

# Query Bar를 확인하자



Query Bar



# Lucene Query의 사용법을 익히자 👑

종류	기능	Query 예시
Keyword 검색	Field에 상관없이 Value 일치하는 Documents 검색	여성
Field Match 검색	특정 Field의 Value가 일치하는 Documents 검색	고객성별:여성
Exact Field Match 검색	특정 Value가 정확히 모두 일치하는 Documents 검색	배송메모:"상품 이상"
Must be 검색	특정 Field가 존재하는 Documents 검색	_exists_:구매사이트
Must not be present 검색	특정 Field가 존재하지 않는 Documents 검색	_missing_:구매사이트
AND 검색	특정 조건들을 모두 만족하는 Documents 검색	고객성별:여성 AND 상품분류:셔츠
OR 검색	특정 조건들 중 적어도 1개를 만족하는 Documents 검색	고객성별:남성 OR 상품분류:셔츠
NOT 검색	특정 조건을 만족하지 않는 Documents 검색	NOT 구매사이트:옥션
Term 검색	조건 중 적어도 하나라도 만족하는 Documents 검색	상품분류: (니트 코트)
Fuzzy 검색	검색어와 유사한 Documents 검색	경상북도~
Proximity 검색	검색어의 순서를 변경해서 찾을 수 있는 Documents 검색	배송메모:"내에 시간 배송 못함"~2
Numeric Value 검색	Numeric Field Value로 Documents 검색	상품가격:>5000
Range 검색	Field의 Value가 입력한 값 사이에 존재하는 Documents 검색	고객나이 : [10 TO 30]
Wildcard ? 검색	Wildcard ?(한글자)를 활용해서 Documents 검색	서?특별시
Wildcard * 검색	Wildcard *(모든글자)를 활용해서 Documents 검색	쿠*

**Filter는 편하지만 기능이 제한적이고,  
Search는 Scripted Field가 검색이 안된다.**

**두 개를 아우르고 싶다면?**

# Filter + Query DSL을 이용하면

		Filter + Query DSL	Filter	Search
“고객성별”이 여성인 Data		✓	✓	✓
“결제카드”가 우리 또는 국민인 Data		✓	✓	✓
“고객성별”이 남성이면서 “연령대”가 20대	SCRIPTED FIELD	✓	✓	
“구매사이트”가 쿠팡이거나 “상품개수”가 1~3인 Data	OR 연산	✓		✓
“결제카드”가 “우”로 시작하는 모든 Data	WILDCARD 검색	✓		✓
“구매사이트”가 22번가(오타 아니에요)와 유사한 Data	FUZZY / PROXIMITY 검색	✓		✓

# Query DSL로 무얼 할 수 있을까?

## Match All Query

match-all

## Full Text Queries

match  
query-string  
⋮

## Term Level Queries

exists  
fuzzy  
prefix  
range  
term  
terms  
wildcard  
⋮

## Specialized Queries

script  
⋮

## Compound Queries

bool  
⋮

## Bool Query로 여러가지 Query를 함께 사용할 수 있다 🏰

A : 고객주소\_시도 = 서울특별시

B : 구매사이트 = 11로 시작

C : 고객나이 < 30

D : 주문날짜 = 일요일

Term Query

Wildcard Query

Range Query

Script Query



위의 Query를 아래와 같은 조건으로 검색 가능

- A AND B
- A AND NOT B
- A OR B
- A AND (B OR C)
- A AND (B OR C OR D 중 2개 이상 만족)

⋮

```
GET {Index 이름}/{Type 이름}/_search
```

```
{
  "query": {
    "bool": {
      "must": [
        {
          "range": {
            "고객나이": {
              "gt": 25
            }
          }
        },
        {
          "must_not": [
            {
              "wildcard": {
                "서울주소_시도": "경?도"
              }
            }
          ]
        }
      ],
      "should": [
        {
          "term": {
            "결제카드": "우리"
          }
        },
        {
          "script": {
            "script": {
              "source": "doc['주문시간'].date.hourOfDay > 18"
            }
          }
        }
      ],
      "minimum_should_match": 1
    }
  }
}
```



반드시 만족해야 한다



반드시 만족하면 안된다



{minimum\_should\_match}개 이상  
만족해야 한다



should clause 내의 query가 1개 이상 참이어야 한다



# Filter에 Query DSL을 적용하자 👑

**Dashboard / week2** Share Clone Edit < Last 6 months >

Search... (e.g. status:200 AND extension:PHP) Uses lucene query syntax 🔍

**Add a filter +** 선택

**Add filter** ✕

**Filter**

Fields... ▼

**Label**

Optional

Cancel Save

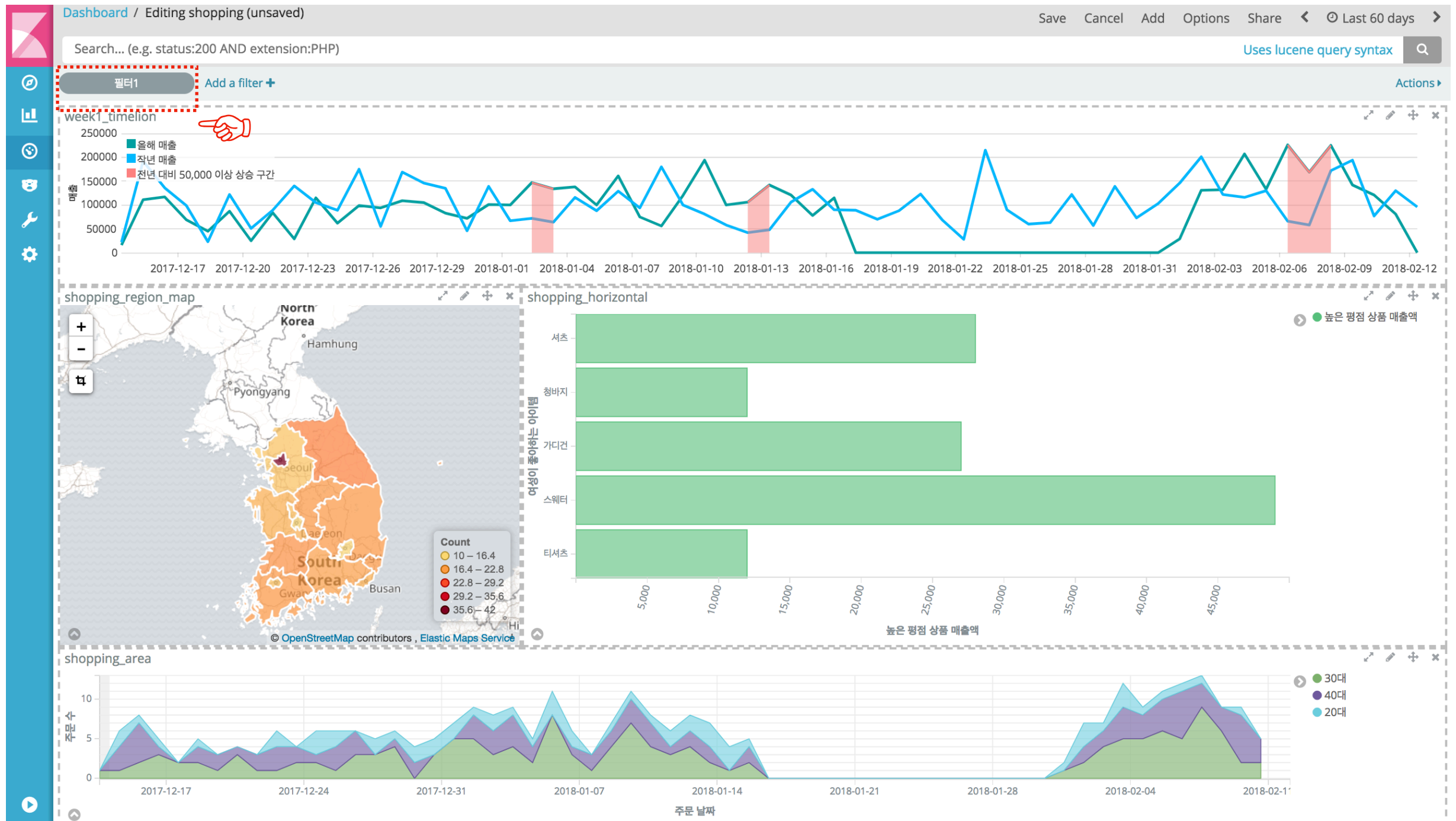
**선택** Edit Query DSL

주소

week2\_datatable

날짜 ⚡	인기 Top 3 ⚡	매출 ⚡	매출 증감 ⚡	매출 누적 ⚡
07월22일 00시00분	스커트	25,000	-	25,000
07월23일 00시00분	청바지, 팬츠, 티셔츠	399,000	374,000	424,000
07월24일 00시00분	니트, 티셔츠, 코트	284,000	-115,000	708,000
07월25일 00시00분	스웨터, 자켓, 니트	217,000	-67,000	925,000
07월26일 00시00분	자켓, 스웨터, 가디건	211,000	-6,000	1,136,000
07월27일 00시00분	자켓, 니트, 가디건	320,000	109,000	1,456,000
07월28일 00시00분	니트, 니트, 점퍼	330,000	10,000	1,786,000
07월29일 00시00분	셔츠, 스웨터, 가디건	312,000	-18,000	2,098,000

# Filter+Query 반영 결과를 보자



**질문 및 Feedback은**  
**gshock94@gmail.com로 주세요**