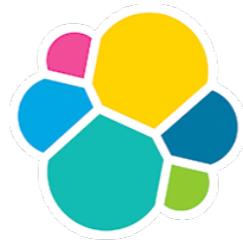


Elastic Stack 을 활용한 Data Dashboard 만들기

Week 5 - Dashboard 만들기 최종실습



Fast Campus

목차

- 설치
 - EC2 Instance 생성
 - EC2 Instance 접속
 - Elastic Stack 설치
 - Elasticsearch 환경설정
 - Logstash 환경설정
 - Kibana 환경설정
 - Elasticsearch
 - Kibana 실행
- Elasticsearch
 - Index 생성
 - Mapping 설정
- Logstash
 - elasticsearch -> elasticsearch
 - jdbc -> elasticsearch
 - file -> elasticsearch
- Kibana
 - Index Pattern 등록
 - Discover 사용
 - Data Format 변경
 - Scripted Field 사용
 - Visualize 구현
 - Timelion 구현
 - Dashboard 구현
 - csv export 하기
 - Filter/Lucene Query 사용
- Elasticserach
 - Document 추가/수정/삭제
 - Search API (Agg) 출력

설치

https://aws.amazon.com/에서 최소한 t2.micro 이상으로 EC2 Instance를 생성하자.

The screenshot shows the AWS EC2 Instances starting page. The top navigation bar includes the AWS logo, service dropdown, resource group dropdown, and a bell icon. The main menu on the left has 'EC2 대시보드' as the active item, followed by '이벤트', '태그', '보고서', '제한', '인스턴스' (selected), 'Launch Templates', '스팟 요청', '예약 인스턴스', and '전용 호스트'. Under '인스턴스', there are sections for 'AMI' and '번들 작업'. On the right, a search bar says '태그 및 속성별 필터 또는 키워드별 검색'. Below it, a message says '현재 리전에는 실행 중인 인스턴스가 없습니다.' and 'EC2 사용이 처음이십니까? [시작 안내서](#) 을(를) 확인해 보십시오.'. A large blue button labeled '인스턴스 시작' is prominent. At the bottom, a footer bar contains icons for feedback, Korean language, copyright information ('© 2008 - 2017, Amazon Web Services, Inc. 또는 자회사. All rights reserved.'), and links for '개인 정보 보호 정책' and '이용 약관'.

생성한 EC2 Instance로 접속하자

```
Last login: Mon Dec 11 11:50:15 2017 from 118.221.38.242
 _ _| _ |_
 _| (   /   Amazon Linux AMI
 __| \__|__|  
  
https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/
6 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-21-251 ~]$
```

Elastic Stack (5.6.4) 을 설치하자

```
[ec2-user@ip-172-31-21-251 fc]$ ls  
elasticsearch-5.6.4 kibana-5.6.4-linux-x86_64 logstash-5.6.4  
[ec2-user@ip-172-31-21-251 fc]$ █
```

Elasticsearch 환경 설정을 하자 - Bootstrap Checks, JVM Options, elasticsearch.yml

```
1 # ===== Elasticsearch Configuration =====
2 #
3 # NOTE: Elasticsearch comes with reasonable defaults for most settings.
4 #       Before you set out to tweak and tune the configuration, make sure you
5 #       understand what are you trying to accomplish and the consequences.
6 #
7 # The primary way of configuring a node is via this file. This template lists
8 # the most important settings you may want to configure for a production cluster.
9 #
10 # Please consult the documentation for further information on configuration options:
11 # https://www.elastic.co/guide/en/elasticsearch/reference/index.html
12 #
13 # ----- Cluster -----
14 #
15 # Use a descriptive name for your cluster:
16 #
17 #cluster.name: my-application
18 #
19 # ----- Node -----
20 #
21 # Use a descriptive name for the node:
22 #
23 #node.name: node-1
24 #
25 # Add custom attributes to the node:
26 #
27 #node.attr.rack: r1
28 #
29 # ----- Paths -----
30 #
31 # Path to directory where to store the data (separate multiple locations by comma):
32 #
33 #path.data: /path/to/data
34 #
35 # Path to log files:
36 #
37 #path.logs: /path/to/logs
38 #
39 # ----- Memory -----
40 #
"elasticsearch-5.6.4/config/elasticsearch.yml" 88L, 2854C
```

Logstash 환경 설정을 하자 - JVM.options

```
1 # JVM configuration
2
3 # Xms represents the initial size of total heap space
4 # Xmx represents the maximum size of total heap space
5
6 -Xms1g
7 -Xmx1g
8
9 #####
10 ## Expert settings
11 #####
12 ##
13 ## All settings below this section are considered
14 ## expert settings. Don't tamper with them unless
15 ## you understand what you are doing
16 ##
17 #####
18
19 ## GC configuration
20 -XX:+UseParNewGC
21 -XX:+UseConcMarkSweepGC
22 -XX:CMSInitiatingOccupancyFraction=75
23 -XX:+UseCMSInitiatingOccupancyOnly
24
25 ## optimizations
26
27 # disable calls to System#gc
28 -XX:+DisableExplicitGC
29
30 ## Locale
31 # Set the locale language
32 #-Duser.language=en
33
34 # Set the locale country
35 #-Duser.country=US
36
37 # Set the locale variant, if any
38 #-Duser.variant=
39
40 ## basic
"config/jvm.options" 80L, 1875C
```

Kibana 환경 설정을 하자 - kibana.yml

```
1 # Kibana is served by a back end server. This setting specifies the port to use.
2 #server.port: 5601
3
4 # Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.
5 # The default is 'localhost', which usually means remote machines will not be able to connect.
6 # To allow connections from remote users, set this parameter to a non-loopback address.
7 #server.host: "localhost"
8
9 # Enables you to specify a path to mount Kibana at if you are running behind a proxy. This only affects
10 # the URLs generated by Kibana, your proxy is expected to remove the basePath value before forwarding requests
11 # to Kibana. This setting cannot end in a slash.
12 #server.basePath: ""
13
14 # The maximum payload size in bytes for incoming server requests.
15 #server.maxPayloadBytes: 1048576
16
17 # The Kibana server's name. This is used for display purposes.
18 #server.name: "your-hostname"
19
20 # The URL of the Elasticsearch instance to use for all your queries.
21 #elasticsearch.url: "http://localhost:9200"
22
23 # When this setting's value is true Kibana uses the hostname specified in the server.host
24 # setting. When the value of this setting is false, Kibana uses the hostname of the host
25 # that connects to this Kibana instance.
26 #elasticsearch.preserveHost: true
27
28 # Kibana uses an index in Elasticsearch to store saved searches, visualizations and
29 # dashboards. Kibana creates a new index if the index doesn't already exist.
30 #kibana.index: ".kibana"
31
32 # The default application to load.
33 #kibana.defaultAppId: "discover"
34
35 # If your Elasticsearch is protected with basic authentication, these settings provide
36 # the username and password that the Kibana server uses to perform maintenance on the Kibana
37 # index at startup. Your Kibana users still need to authenticate with Elasticsearch, which
38 # is proxied through the Kibana server.
39 #elasticsearch.username: "user"
40 #elasticsearch.password: "pass"
"kibana-5.6.4-linux-x86_64/config/kibana.yml" 104L, 4649C
```

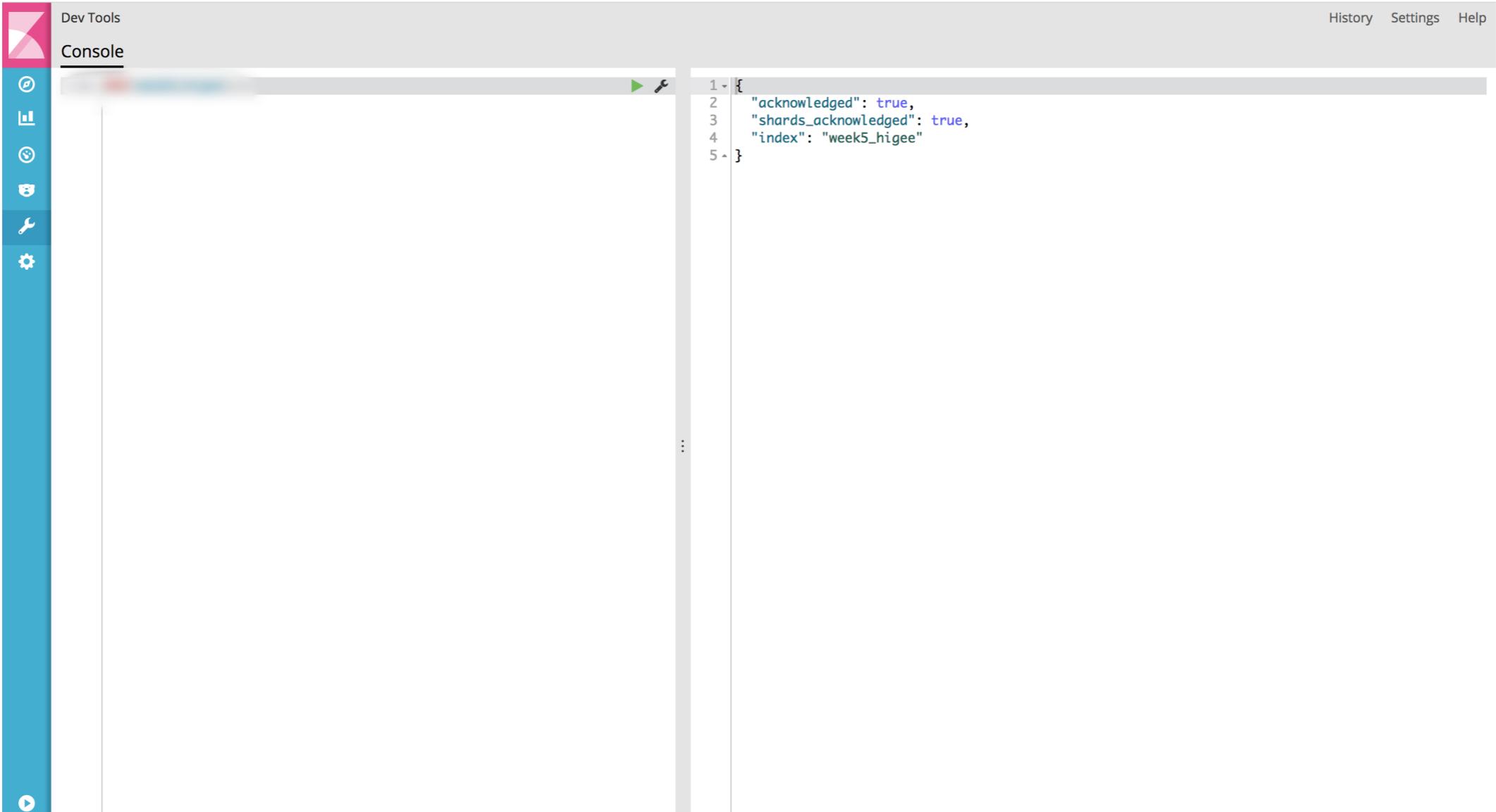
Elasticsearch를 실행하자

```
[2017-12-11T17:46:54,463][INFO ][o.e.n.Node      ] [ ] initializing ...
[2017-12-11T17:46:54,538][INFO ][o.e.e.NodeEnvironment  ] [ ] [ZcJS5Jd] using [1] data paths, mounts [[/ (/dev/xvda1)]], net usable_space [7.2gb], net total_spa
ce [15.6gb], spins? [no], types [ext4]
[2017-12-11T17:46:539][INFO ][o.e.e.NodeEnvironment  ] [ ] [ZcJS5Jd] heap size [3.9gb], compressed ordinary object pointers [true]
[2017-12-11T17:46:54,942][INFO ][o.e.n.Node      ] [ ] node name [ZcJS5Jd] derived from node ID [ZcJS5JdIR8mgPgM74IxwDw]; set [node.name] to override
[2017-12-11T17:46:54,943][INFO ][o.e.n.Node      ] [ ] version[5.6.4], pid[8857], build[8bbbedf5/2017-10-31T18:55:38.105Z], OS[Linux/4.9.62-21.56.amzn1.x8
6_64/amd64], JVM[Oracle Corporation/Java HotSpot(TM) 64-Bit Server VM/1.8.0_141/25.141-b15]
[2017-12-11T17:46:54,943][INFO ][o.e.n.Node      ] [ ] JVM arguments [-Xms4g, -Xmx4g, -Xms4g, -Xmx4g, -XX:+UseConcMarkSweepGC, -XX:CMSInitiatingOccupancy
Fraction=75, -XX:+UseCMSInitiatingOccupancyOnly, -XX:+AlwaysPreTouch, -Xss1m, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -Djdk.io.perm
issionsUseCanonicalPath=true, -Dio.netty.noUnsafe=true, -Dio.netty.noKeySetOptimization=true, -Dio.netty.recycler.maxCapacityPerThread=0, -Dlog4j.shutdownHook
Enabled=false, -Dlog4j2.disable.jmx=true, -Dlog4j.skipJansi=true, -XX:+HeapDumpOnOutOfMemoryError, -Des.path.home=/home/ec2-user/fc/elasticsearch-5.6.4]
[2017-12-11T17:46:55,792][INFO ][o.e.p.PluginsService  ] [ ] [ZcJS5Jd] loaded module [aggs-matrix-stats]
[2017-12-11T17:46:55,793][INFO ][o.e.p.PluginsService  ] [ ] [ZcJS5Jd] loaded module [ingest-common]
[2017-12-11T17:46:55,793][INFO ][o.e.p.PluginsService  ] [ ] [ZcJS5Jd] loaded module [lang-expression]
[2017-12-11T17:46:55,793][INFO ][o.e.p.PluginsService  ] [ ] [ZcJS5Jd] loaded module [lang-groovy]
[2017-12-11T17:46:55,793][INFO ][o.e.p.PluginsService  ] [ ] [ZcJS5Jd] loaded module [lang-mustache]
[2017-12-11T17:46:55,793][INFO ][o.e.p.PluginsService  ] [ ] [ZcJS5Jd] loaded module [lang-painless]
[2017-12-11T17:46:55,793][INFO ][o.e.p.PluginsService  ] [ ] [ZcJS5Jd] loaded module [parent-join]
[2017-12-11T17:46:55,793][INFO ][o.e.p.PluginsService  ] [ ] [ZcJS5Jd] loaded module [percolator]
[2017-12-11T17:46:55,793][INFO ][o.e.p.PluginsService  ] [ ] [ZcJS5Jd] loaded module [reindex]
[2017-12-11T17:46:55,794][INFO ][o.e.p.PluginsService  ] [ ] [ZcJS5Jd] loaded module [transport-netty3]
[2017-12-11T17:46:55,794][INFO ][o.e.p.PluginsService  ] [ ] [ZcJS5Jd] loaded module [transport-netty4]
[2017-12-11T17:46:55,794][INFO ][o.e.p.PluginsService  ] [ ] [ZcJS5Jd] no plugins loaded
[2017-12-11T17:46:57,634][INFO ][o.e.d.DiscoveryModule  ] [ ] [ZcJS5Jd] using discovery type [zen]
[2017-12-11T17:46:58,629][INFO ][o.e.n.Node      ] [ ] initialized
[2017-12-11T17:46:58,629][INFO ][o.e.n.Node      ] [ ] [ZcJS5Jd] starting ...
[2017-12-11T17:46:58,764][INFO ][o.e.t.TransportService  ] [ ] [ZcJS5Jd] publish_address {172.31.21.251:9300}, bound_addresses {172.31.21.251:9300}
[2017-12-11T17:46:58,774][INFO ][o.e.b.BootstrapChecks  ] [ ] [ZcJS5Jd] bound or publishing to a non-loopback or non-link-local address, enforcing bootstrap che
cks
[2017-12-11T17:47:01,826][INFO ][o.e.c.s.ClusterService  ] [ ] [ZcJS5Jd] new_master {ZcJS5Jd}{ZcJS5JdIR8mgPgM74IxwDw}{{PGXkJjSSLqDIRbS0zFGbw}{ec2-13-125-21-52.ap
-northeast-2.compute.amazonaws.com}{172.31.21.251:9300}}, reason: zen-disco-elected-as-master ([0] nodes joined)
[2017-12-11T17:47:01,868][INFO ][o.e.h.n.Netty4HttpServerTransport  ] [ ] [ZcJS5Jd] publish_address {172.31.21.251:9200}, bound_addresses {172.31.21.251:9200}
[2017-12-11T17:47:01,868][INFO ][o.e.n.Node      ] [ ] [ZcJS5Jd] started
[2017-12-11T17:47:03,623][INFO ][o.e.g.GatewayService  ] [ ] [ZcJS5Jd] recovered [121] indices into cluster_state
```

Kibana를 실행하자

```
log [17:49:53.949] [info][status][plugin:kibana@5.6.4] Status changed from uninitialized to green - Ready
log [17:49:54.019] [info][status][plugin:elasticsearch@5.6.4] Status changed from uninitialized to yellow - Waiting for Elasticsearch
log [17:49:54.050] [info][status][plugin:console@5.6.4] Status changed from uninitialized to green - Ready
log [17:49:54.065] [info][status][plugin:metrics@5.6.4] Status changed from uninitialized to green - Ready
log [17:49:54.269] [info][status][plugin:timelion@5.6.4] Status changed from uninitialized to green - Ready
log [17:49:54.279] [info][listening] Server running at http://ec2-13-125-21-52.ap-northeast-2.compute.amazonaws.com:5601
log [17:49:54.287] [info][status][ui settings] Status changed from uninitialized to yellow - Elasticsearch plugin is yellow
log [17:49:54.572] [info][status][plugin:elasticsearch@5.6.4] Status changed from yellow to green - Kibana index ready
log [17:49:54.573] [info][status][ui settings] Status changed from yellow to green - Ready
```

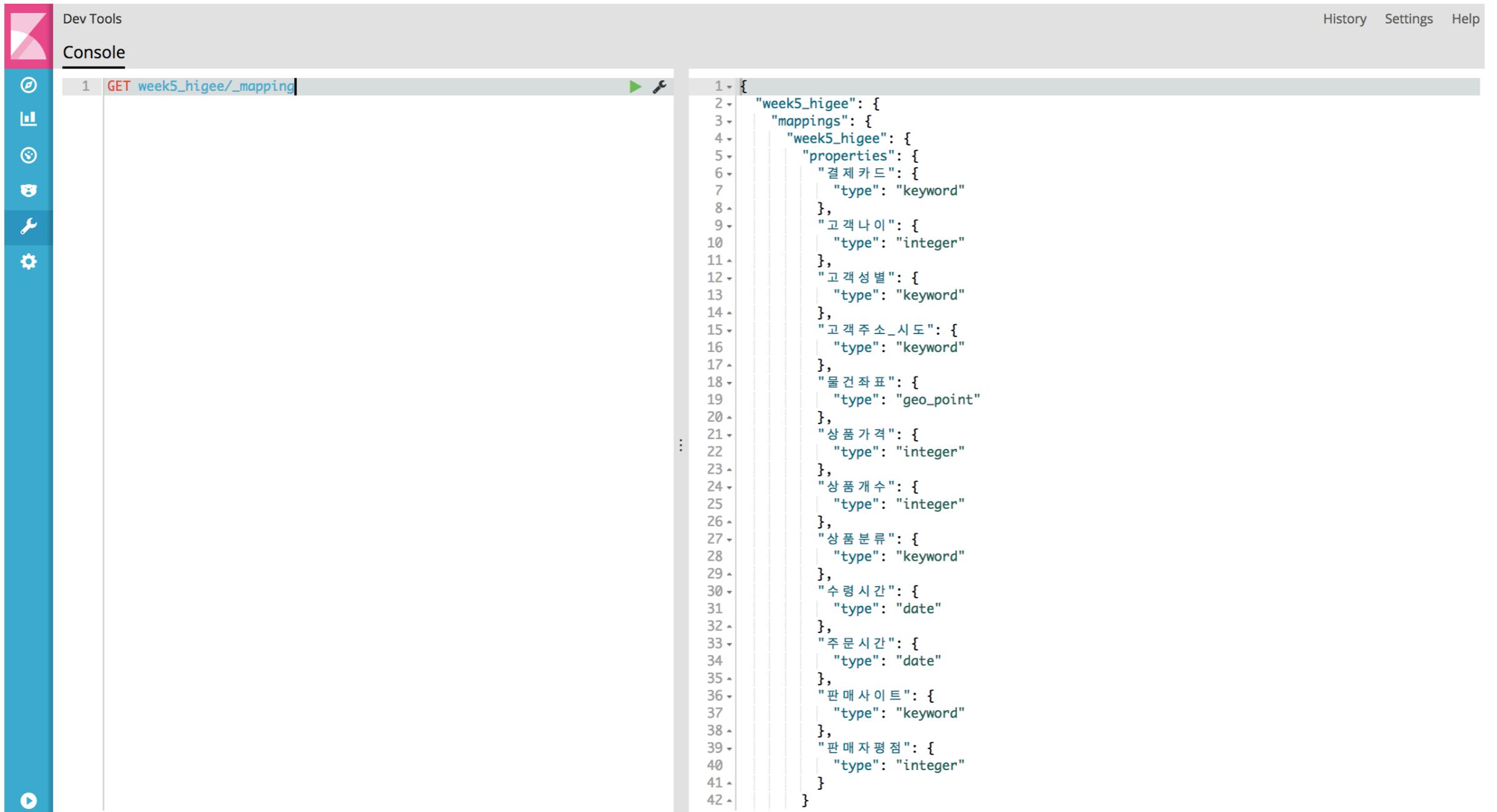
week5_{id} 라는 이름으로 Index를 생성하자



The screenshot shows the Elasticsearch Dev Tools interface with the 'Console' tab selected. On the left is a vertical toolbar with icons for Dev Tools, Search, Metrics, Health, Scripts, and Settings. The main area is the 'Console' panel, which contains a command-line input field and a results panel. The results panel displays the following JSON response:

```
1 {  
2   "acknowledged": true,  
3   "shards_acknowledged": true,  
4   "index": "week5_higee"  
5 }
```

아래와 같이 mapping을 설정하자



The screenshot shows the Elasticsearch Dev Tools interface with the 'Console' tab selected. A search bar at the top contains the query 'GET week5_higee/_mapping'. The main area displays the mapping configuration for the 'week5_higee' index. The mapping is defined as follows:

```
1 {  
2   "week5_higee": {  
3     "mappings": {  
4       "week5_higee": {  
5         "properties": {  
6           "결제 카드": {  
7             "type": "keyword"  
8           },  
9           "고객 나이": {  
10            "type": "integer"  
11           },  
12           "고객 성별": {  
13            "type": "keyword"  
14           },  
15           "고객 주소_시도": {  
16            "type": "keyword"  
17           },  
18           "물건 좌표": {  
19            "type": "geo_point"  
20           },  
21           "상품 가격": {  
22            "type": "integer"  
23           },  
24           "상품 개수": {  
25            "type": "integer"  
26           },  
27           "상품 분류": {  
28            "type": "keyword"  
29           },  
30           "수령 시간": {  
31            "type": "date"  
32           },  
33           "주문 시간": {  
34            "type": "date"  
35           },  
36           "판매 사이트": {  
37            "type": "keyword"  
38           },  
39           "판매자 평점": {  
40            "type": "integer"  
41           }  
42         }  
      }  
    }  
}
```

Logstash

다음과 같은 조건을 만족하는 elasticsearch의 document만 logstash로 옮겨보자

Input

- host : 13.125.21.52:9200
- index : shopping
- query
 - Field : 결제카드, 고객나이, 고객성별, 고객주소_시도, 물건좌표, 상품가격, 상품개수, 상품분류, 수령시간, 주문시간, 판매사이트, 판매자평점
 - 조건
 - 20 <= 고객나이 <= 30
 - 구매사이트 : 옥션, 쿠팡, 11번가 중 하나

Output

- host : 13.125.21.52:9200
- index : week5_{id}

```
1 input {
2   elasticsearch {
3     type => "log"
4     hosts => ["13.125.21.52:9200"]
5     query {
6       _source {
7         excludes [ "판매자평점" ]
8         query {
9           bool {
10             must {
11               query_string {
12                 query => "판매자평점:5"
13                 boost => 1.0
14               }
15             }
16           }
17         }
18       }
19     }
20     query {
21       _source {
22         excludes [ "판매자평점" ]
23         query {
24           bool {
25             must {
26               query_string {
27                 query => "판매자평점:5"
28                 boost => 1.0
29               }
30             }
31           }
32         }
33       }
34     }
35   }
36 }
37 output {
38   elasticsearch {
39     hosts => ["13.125.21.52:9200"]
40     type => "log"
41     index => "week5_higee"
42   }
43 }
```

~
~
~

"final1.conf" 42L, 965C

42,1 All

Logstash

mysql의 데이터를 logstash로 elasticsearch에 옮겨보자

- host : 13.125.21.52:3306
- user/password : fc/fc
- database : fc
- table : fc

elasticsearch

- host : 13.125.21.52:9200
- index : week5_{id}_jdbc
- type : week5_{id}_jdbc

id age salary name location employment_status
1 33 4000 Josh US employed
2 33 45000 Tom US employed
3 23 3000 Kirk US employed
4 27 3500 Ken US employed
5 38 4500 Jessie US employed
6 31 5200 Jennifer US unemployed
7 33 22200 Bob US unemployed
8 25 2200 John US unemployed
9 45 4200 Aaron UK unemployed
10 45 6200 Hanks UK unemployed
11 43 8200 Gordon UK employed
12 47 11200 Searl UK employed
13 37 9900 Waterhouse UK employed
14 25 3900 Lisa UK employed
15 28 4500 Kelly UK employed
16 38 47000 Tim UK employed
17 38 3500 Tanaka JP employed
18 28 2500 Kogawa JP unemployed
19 31 3500 Oota JP employed
20 30 3700 Yamada JP employed
21 44 4500 Saeki JP employed
22 44 4500 Abe JP employed
23 42 4200 Murata JP unemployed
24 46 5600 Yamauchi JP unemployed
25 43 8600 Siranui JP unemployed
26 48 8200 Kawakami JP unemployed
27 48 8200 Kim KR unemployed
28 48 3200 Lee KR unemployed
29 48 3500 Kwon KR unemployed
30 28 3300 Kang KR employed
31 23 3500 Yoon KR unemployed
32 23 3600 Yang KR unemployed
33 26 3800 Park KR employed

Logstash

다음 csv 파일을 적당히 처리해서 elasticsearch로 전송하자

column 이름 (순서) : "Index", "Name", "Survival", "Pclass", "Sex", "Age", "SibSp", "Parch", "Ticket", "Fare", "Embarked"]

data type 변환

- "Pclass" => "integer"
- "Index" => "integer"
- "Survival" => "integer"
- "Age" => "integer"
- "Fare" => "float"

데이터

<https://gist.githubusercontent.com/higee/1e3c3137195cf14eb23dd827a55e9b1d/raw/388c7df111beded4d136465474fcd9bf7826b545/titanic.csv>

elasticsearch

- host : 13.125.21.52:9200
- index : week5_{id}_titanic
- type : week5_{id}_titanic

Kibana

Kibana에서 Index Patterns (**week5_higee**) 를 등록하자

The screenshot shows the Kibana Management interface with the title 'Management / Kibana'. On the left, there is a sidebar with various icons and a list of index patterns. The main area is titled 'Configure an index pattern' and contains the following fields:

- Index pattern:** week5_higee
- Time Filter field name:** @version
- Use event times to create index names [DEPRECATED]

At the bottom right of the configuration area is a blue 'Create' button.

Discover에서 다음과 같은 질문에 답해보자

1. 지난 2년 간 기준으로 볼 경우, 총 Documents의 수는?
2. 지난 2년 간 기준으로 볼 경우, Count가 가장 많았던 월은?
3. 수령시간 기준으로 최신순으로 정렬할 경우, 결제카드의 비율은?
4. 상품가격을 큰 순으로 정렬할 경우, 상품분류의 비율은?

수령시간이라는 Field의 표기를 “XX월 XX일” 형식으로 변환

fields (17)		scripted fields (5)	source filters (0)			
		Filter	All field types ▾			
name	type	format	searchable	aggregatable	excluded	controls
_id	string		✓			
_index	string		✓	✓		
_score	number					
_source	_source					
_type	string		✓	✓		
결제카드	string		✓	✓		
고객나이	number		✓	✓		
고객성별	string		✓	✓		
고객주소_시도	string		✓	✓		
물건좌표	geo_point		✓	✓		
상품가격	number		✓	✓		
상품개수	number		✓	✓		
상품분류	string		✓	✓		
수령시간	date	Date	✓	✓		
주문시간	date		✓	✓		
판매사이트	string		✓	✓		
판매자평점	number		✓	✓		

[Scroll to top](#)Page Size

Scripted Field를 이용해서 다음과 같은 2개의 Field를 생성해보자

- 시간대 : 주문시간에서 시간대만 추출 (예: 13시 55분 -> 13)
- 요일 : 주문시간에서 요일만 추출 (예: 2017년 12월 12일 -> 화)

Management / Kibana

Index Patterns Saved Objects Advanced Settings

★ week5_higee

Time Filter field name: 수령시간

This page lists every field in the **week5_higee** index and the field's associated core type as recorded by Elasticsearch. While this list allows you to view the core type of each field, changing field types must be done using Elasticsearch's [Mapping API](#).

fields (17) scripted fields (3) source filters (0)

Filter All languages ▾

Scripted fields

These scripted fields are computed on the fly from your data. They can be used in visualizations and displayed in your documents, however they can not be searched. You can manage them here and add new ones as you see fit, but be careful, scripts can be tricky!

+ Add Scripted Field

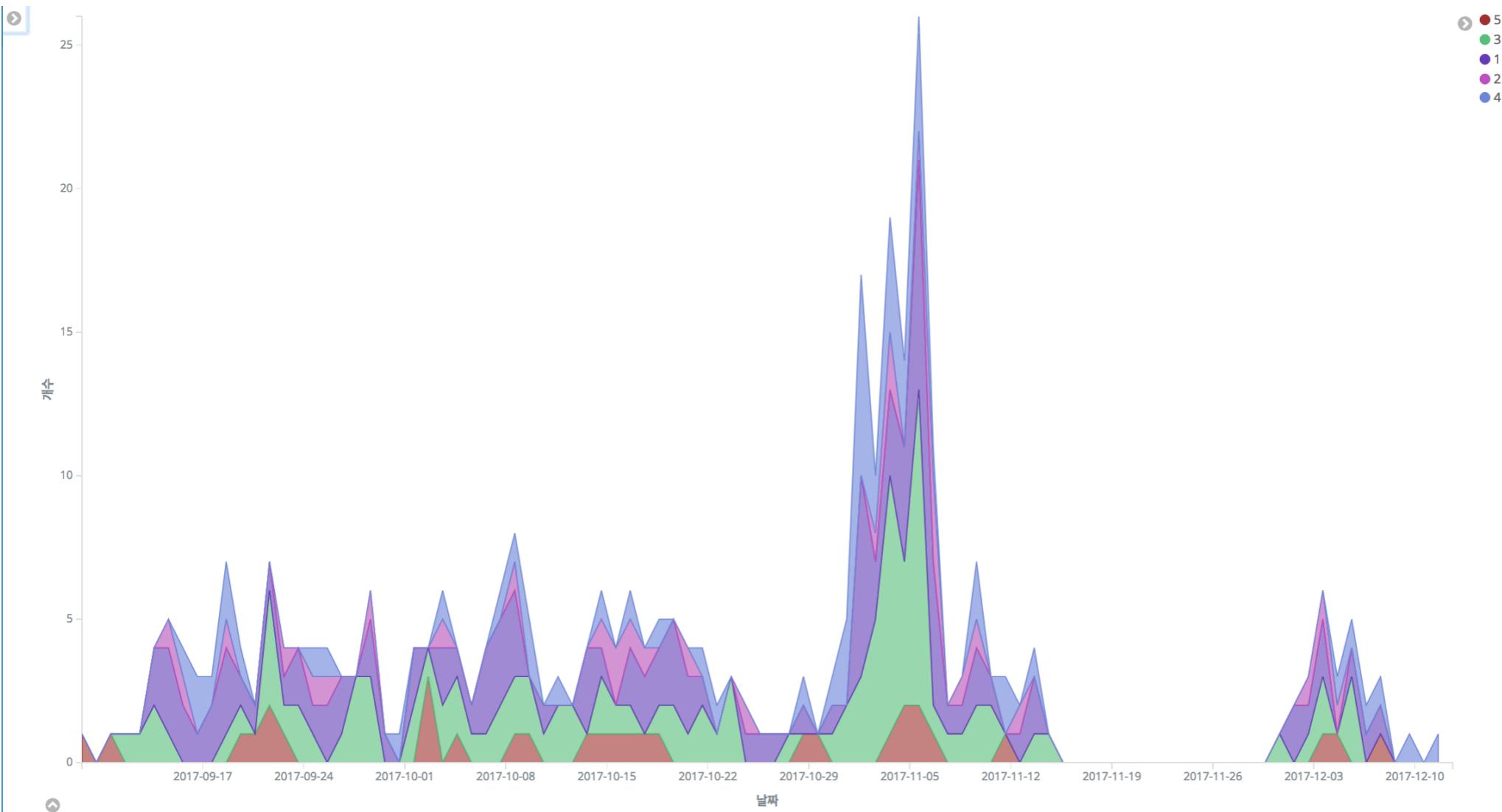
name	lang	script	format	controls
시간대	painless	doc['수령시간'].date.hourOfDay		
요일	painless	doc['수령시간'].date.dayOfWeek		
_id				
요일	painless	{ doc['수령시간'].date.dayOfWeek == 1 ? '월' : (doc['수령시간'].date.dayOfWeek == 2 ? '화' : (doc['수령시간'].date.dayOfWeek == 3 ? '수' : (doc['수령시간'].date.dayOfWeek == 4 ? '목' : (doc['수령시간'].date.dayOfWeek == 5 ? '금' : (doc['수령시간'].date.dayOfWeek == 6 ? '토' : '일'))))})		

Scroll to top Page Size 25 ▾

Kibana

다음과 같은 Visualization을 만들어보자

- x축 : 주문시간 기준으로 일별로 정렬
- y축 : 주문 개수 (판매자 평점 별)

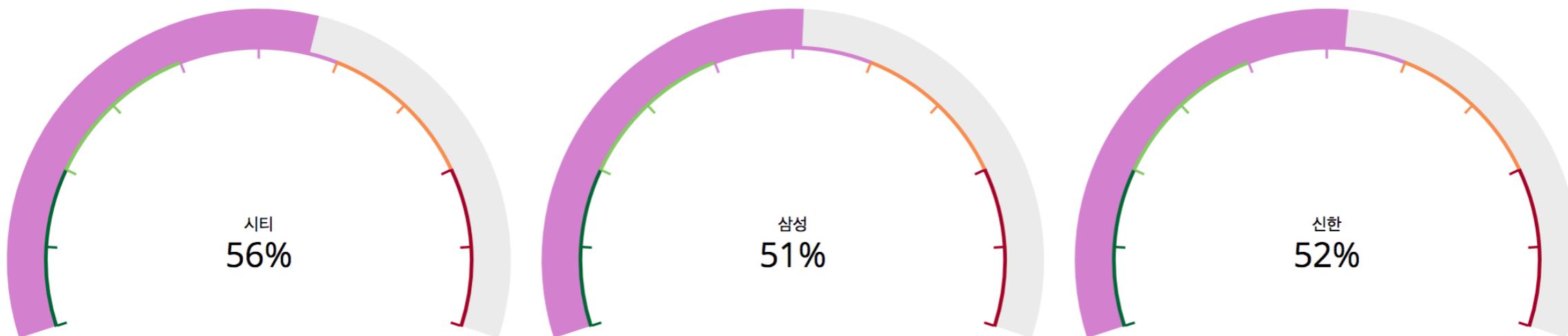


날짜 Last 90 days

Kibana

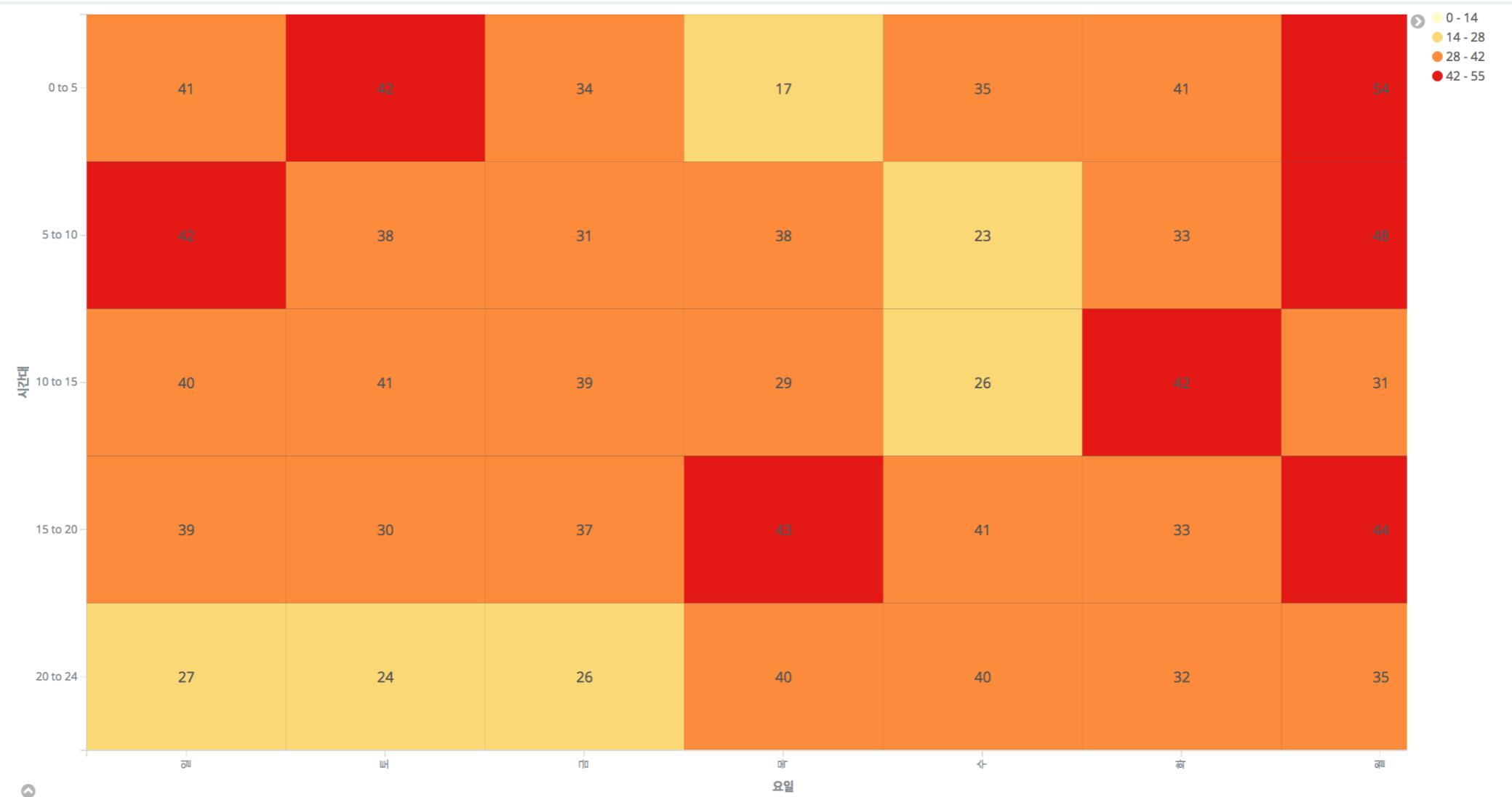
다음과 같은 Visualization을 만들어보자

- 평균 상품가격이 높은 결제카드 3개 선별
- 각 카드사의 평균 판매자 평점을 0~1, 1~2, 2~3, 3~4, 4~5 구간으로 목표 선정



날짜는 Last 1 year

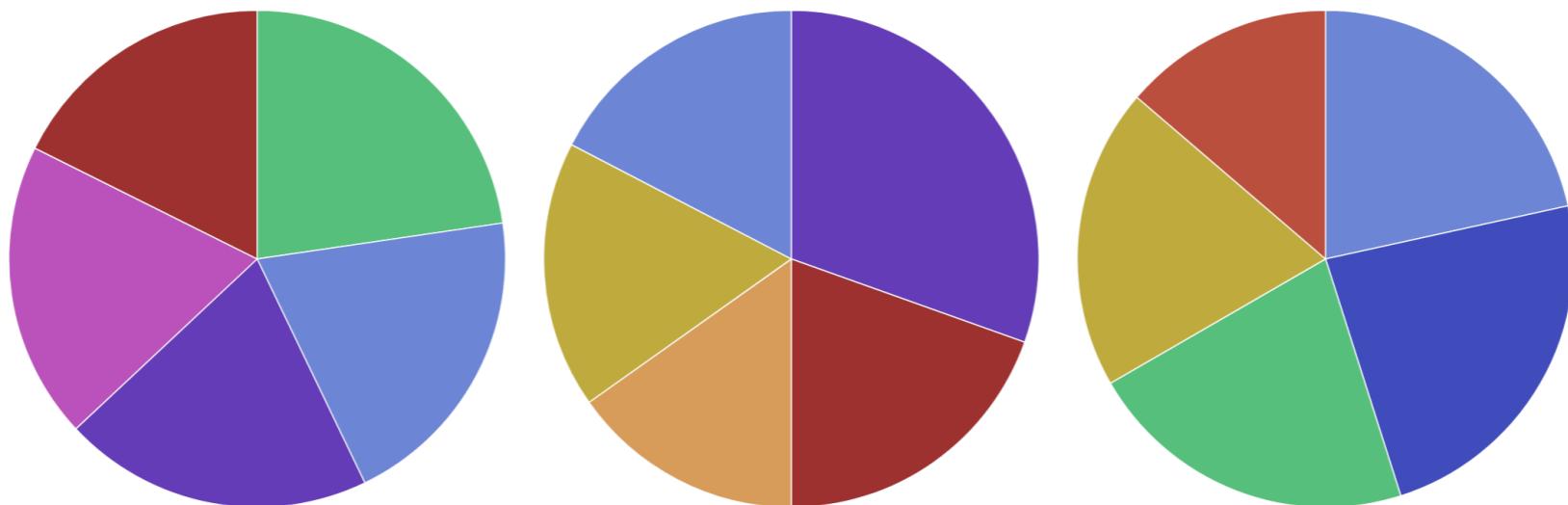
요일별/시간대별 Count를 나타내는 Heatmap을 만들어보자



날짜는 Last 1 year

다음과 같은 Visualization을 만들어보자

- 상품가격의 합이 높은 고객주소_시도 3곳 선정
- 각각의 Pie에서는 상품가격의 합이 큰 상품분류 5곳의 Count Value 시각화



서울특별시: 고객주소_시도

충청남도: 고객주소_시도

경상남도: 고객주소_시도

날짜는 Last 1 year

Kibana

다음과 같은 Visualization을 만들어보자

- 날짜는 수령시간 기준으로 Daily
- 첫 번째 column에는 일별 상품가격의 합 (매출)
- 두 번째 column에는 일별 매출 증감
- 세 번째 column에는 매출 누적합
- 네 번째 column에는 매출 3일 이동 평균

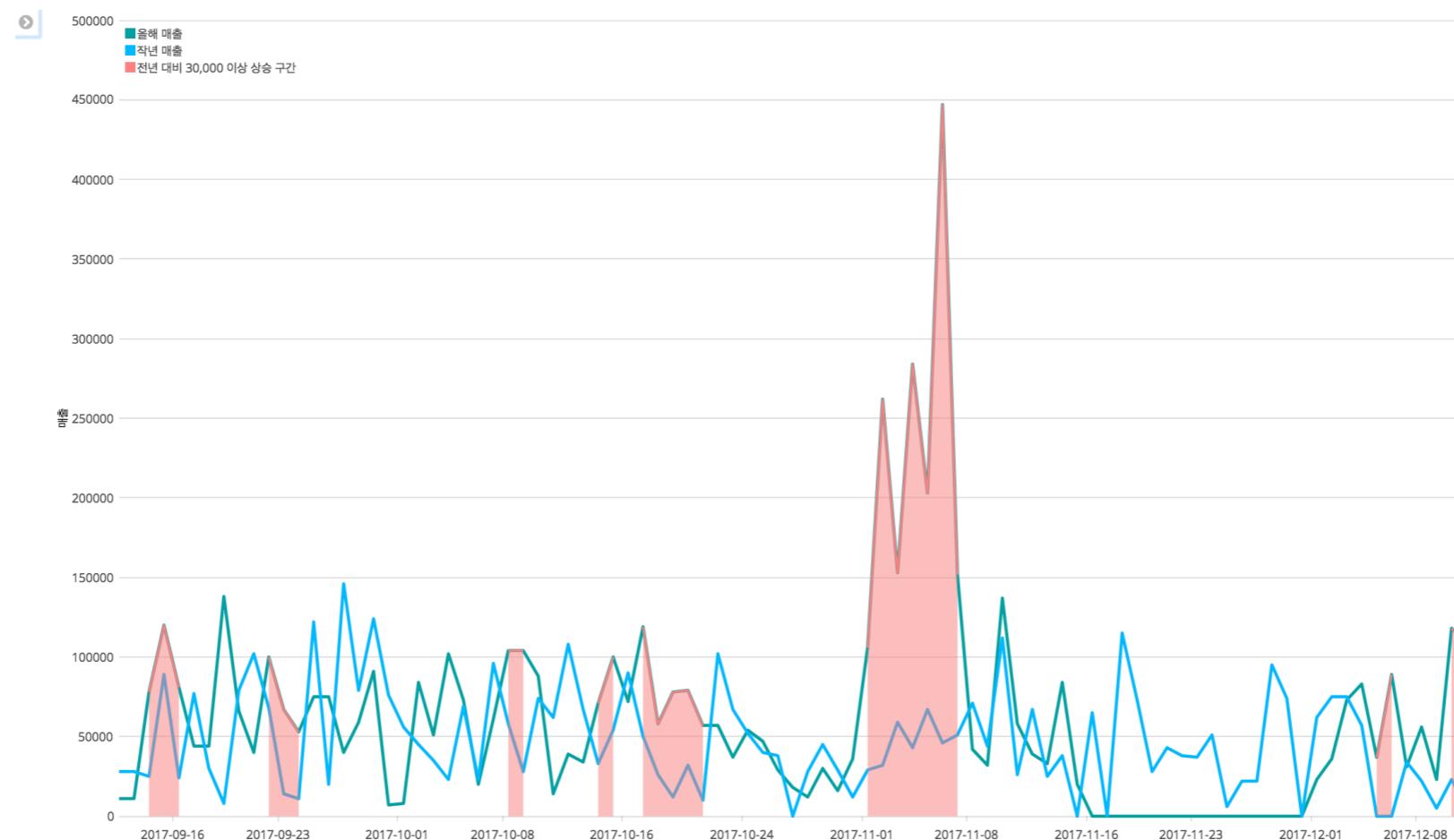
수령 시간 ◆	상품가격 합 (매출) ◆	일별 매출 증감 ◆	매출 누적 ◆	매출 이동 평균 (3일) ◆
12월12일	17,000	-	17,000	-
12월13일	18,000	1,000	35,000	17,000
12월14일	21,000	3,000	56,000	17,500
12월15일	60,000	39,000	116,000	18,666.667
12월16일	47,000	-13,000	163,000	33,000
12월17일	75,000	28,000	238,000	42,666.667
12월18일	117,000	42,000	355,000	60,666.667
12월19일	86,000	-31,000	441,000	79,666.667
12월20일	7,000	-79,000	448,000	92,666.667
12월21일	21,000	14,000	469,000	70,000

날짜는 Last 1 year

Kibana

다음과 같은 Visualization을 만들어보자

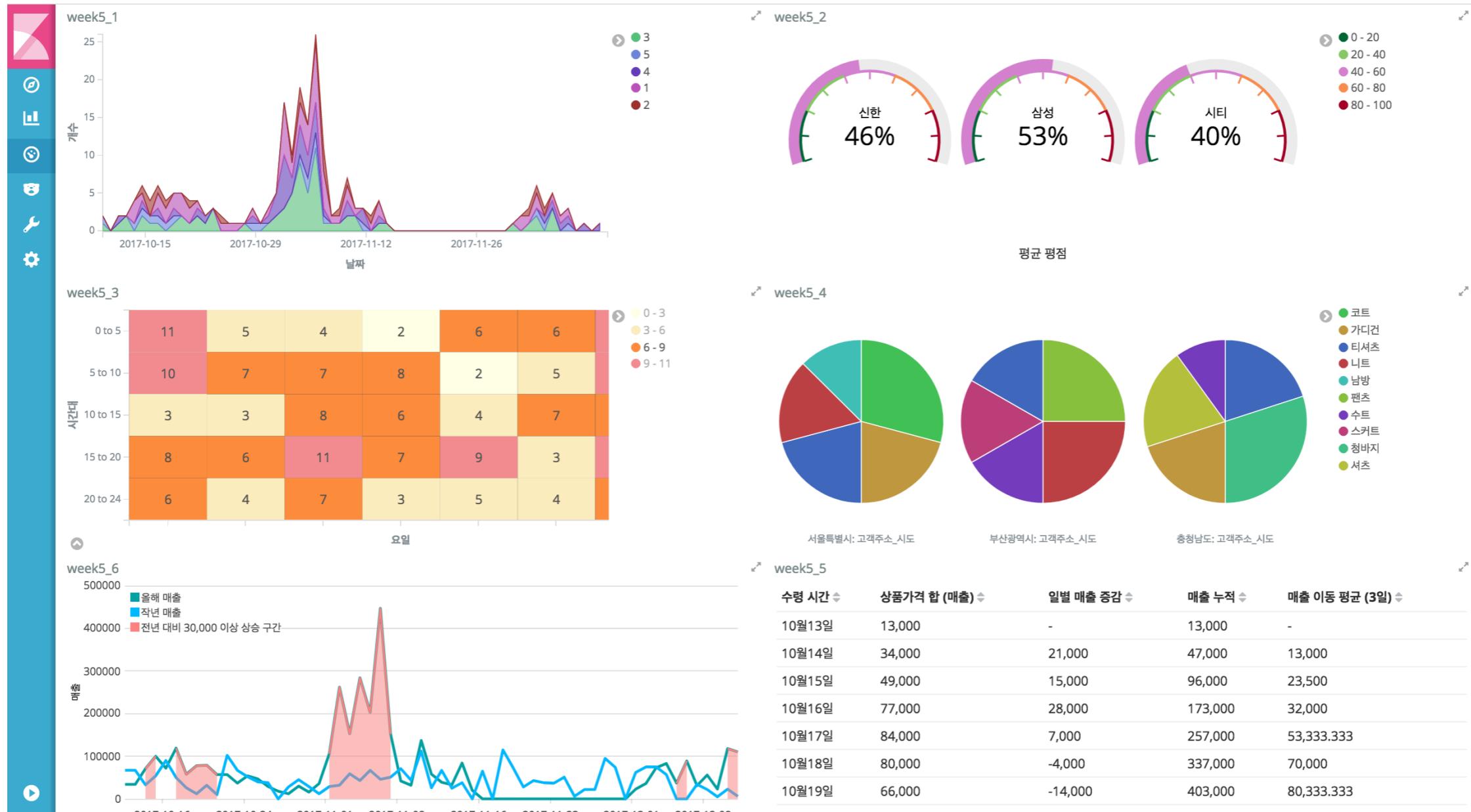
- 올해 매출 (=상품가격의 합)을 표시하고
- 작년 매출을 표시하고
- 전년 대비 30,000원 이상 오른 구간을 색칠해보자



날짜는 Last 90 days

Kibana

위에서 만든 Visualization을 하나의 Dashboard로 합쳐보자



Kibana

아래 visualization 데이터를 csv로 export 해보자

수령 시간 ◆	상품가격 합 (매출) ◆	일별 매출 증감 ◆	매출 누적 ◆	매출 이동 평균 (3일) ◆
12월12일	17,000	-	17,000	-
12월13일	18,000	1,000	35,000	17,000
12월14일	21,000	3,000	56,000	17,500
12월15일	60,000	39,000	116,000	18,666.667
12월16일	47,000	-13,000	163,000	33,000
12월17일	75,000	28,000	238,000	42,666.667
12월18일	117,000	42,000	355,000	60,666.667
12월19일	86,000	-31,000	441,000	79,666.667
12월20일	7,000	-79,000	448,000	92,666.667
12월21일	21,000	14,000	469,000	70,000

Filter와 Lucene Query를 이용해서 Discover Page를 보며 다음 질문에 답해보자

1. 주말 거래건수(총 Count)는?
2. 주말 동안 신한 카드 또는 국민 카드 결제 건수는?
3. 판매자 평점이 2~4 사이면서 상품분류가 “자”로 시작하는 건수는?
4. 20~25세 남성 고객의 10~17시를 제외한 시간대 이용건수는?
5. 고객주소_시도가 광역시이면서 (니트, 스웨터, 청바지) 중에 적어도 하나를 이용한 건수는?

(단, 기간은 2017-01-01에서 2017-12-11까지)

Kibana

다음과 같은 Document를 week5_{id} index에 넣어보자

```
"고객주소_시도": "경상남도",
"상품가격": 9000,
"고객성별": "남성",
"고객나이": 35,
"판매자평점": 1,
"수령시간": "2017-09-18T14:15:33",
"상품분류": "자켓",
"상품개수": 3,
"물건좌표": "36.95300223329071, 126.06934136285218",
"주문시간": "2017-11-24T10:31:33",
"결제카드": "국민"
},
```

Kibana

다음과 같이 Document를 수정해보자

- 1) 고객나이가 20-25세 사이면서
- 2) 고객주소_시도가 서울특별시이고
- 3) 고객성별이 남성인

Documents의 상품분류를 모두 “자켓”으로 바꿔보자

Kibana

다음과 같이 Document를 삭제해보자

- 1) 고객나이가 20-25세 사이면서
- 2) 고객주소_시도가 서울특별시이고
- 3) 고객성별이 남성

Search API(+aggregation)를 이용해서 다음을 구해보자

- 1) 고객나이가 20-25세 사이인 사람들의 평균 상품금액
- 2) 고객주소_시도가 서울특별시거나 경상남도인 사람들의 평균 판매자평점
- 3) Daily 수령시간 별 거래건수가 많은 카드사 3개 별 상품가격이 가장 컸던 상품분류