Elastic Stack으로 Dashboard 구축하기

신효정

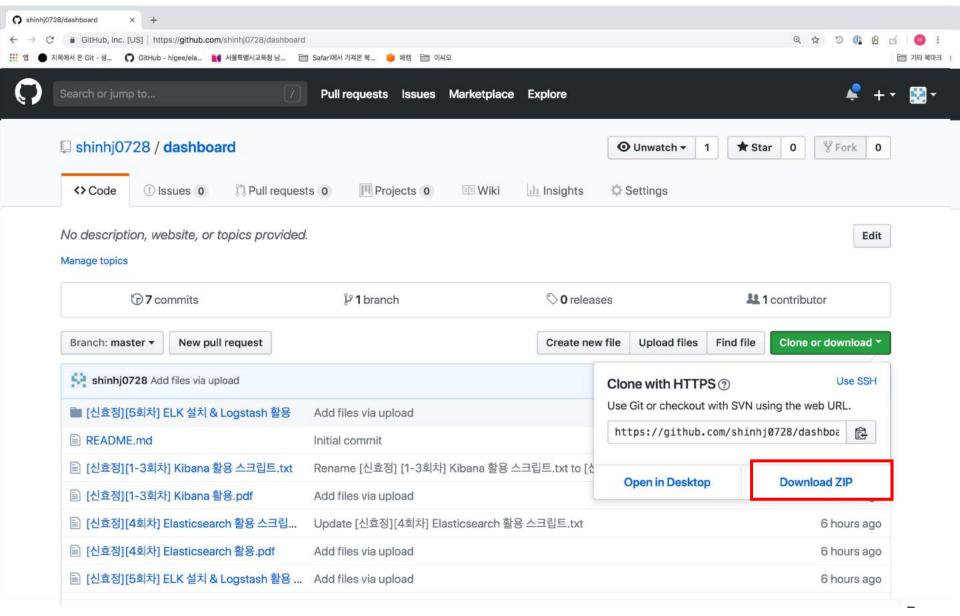




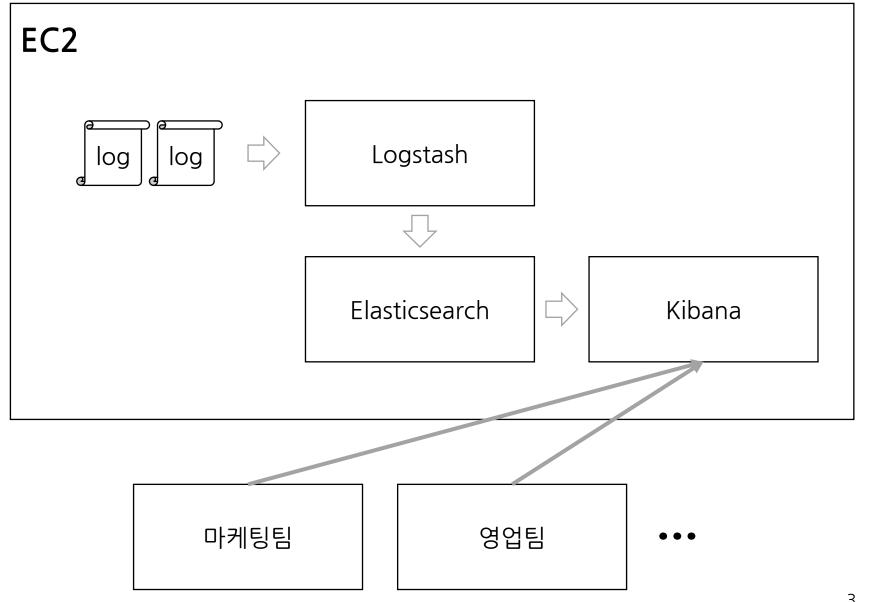


강의 자료

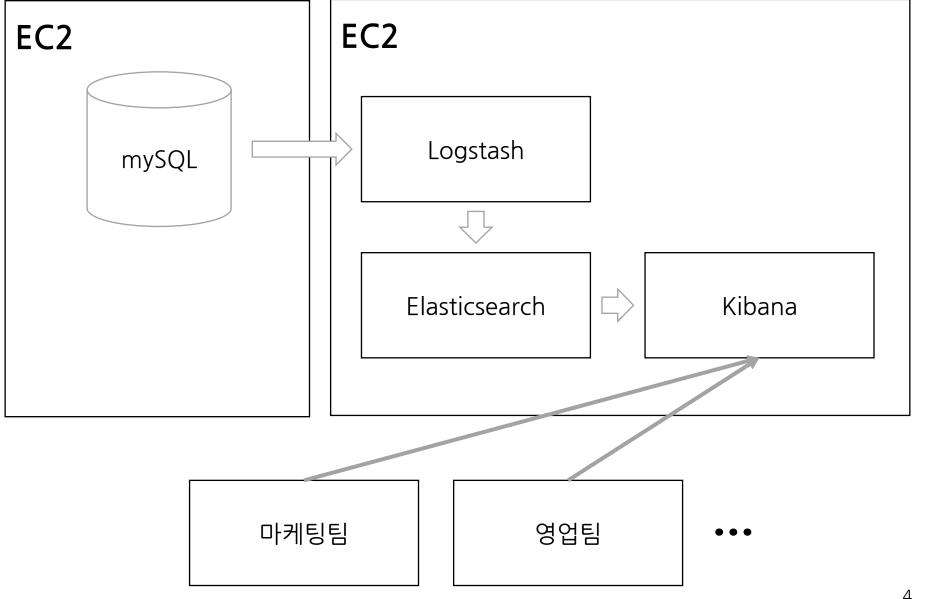
https://github.com/shinhj0728/dashboard



실습 1. file로 elasticsearch index 구축하고 Dashboard 만뚤뼛ㄸㅠpus



실습 2. mysql로 elasticsearch index 구축하고 Dashboard 만들어마



Dashboard 만들기 workflow

```
EC2 환경설정
```

-----> 작업 시작

ELK 설치

EK 설정 및 실행

-----> ELK 설치 완료

E index 구조 잡기

L 설정 및 실행

-----> index 생성(데이터 수집)

Dashboard 만들기

- 1) index 확인(discover)
- 2) dashboard의 목적 정의
- 3) metric 정의
- 4) 각 metric을 위한 agg./Visualize type 선정
- 5) metric별 object 생성 후 dashboard 구축
- -----> dashboard 구축 완료

Feed back

- 1) 수집 데이터 수정
- 2) 'Dashboard 만들기' 수정

File to ES

Dashboard 만들기 workflow

- EC2 환경설정 [신효정][5회차]&LK6설치 & Logstash (활용 스크립트.txt -----> 작업 시작
- ELK 설치 [신효정][5회차]ŒLK6설치 & 6Logstash 全용 스크립트.txt
- EK 설정 및 실행 [신효정][5회차]&LK6설치 & Logstash (활용 스크립트.txt
- -----> ELK 설치 완료
- E index 구조 잡기 [신효정][6회차] Elastic Stack으로 Dashboard 구축하기 스크립트 1.txt
- L 설정 및 실행 [신효정][6회차] Elastic Stack으로 Dashboard 구축하기 스크립트 1.txt
- -----> index 생성(데이터 수집)

Dashboard 만들기

- 1) index 확인(discover)
- 2) dashboard의 목적 정의
- 3) metric 정의
- 4) 각 metric을 위한 agg./Visualize type 선정
- 5) metric별 object 생성 후 dashboard 구축
- -----> dashboard 구축 완료

Feed back

- 1) 수집 데이터 수정
- 2) 'Dashboard 만들기' 수정

Data 준비



개인 EC2 접속 - ssh / putty

```
데이터 준비
$ cd ~/data
$ git pull
$ ls
```

```
| Equation | Telegration | Te
```

정보확인



ES로 입력할 파일의 위치 및 정보(logstash가 실행 될 서버의)

- /home/ec2-user/data/

Elasticsearch가 실행되고 있는 서버의 IP주소

- ec2-54-180-12-248.ap-northeast-2.compute.amazonaws.com

Kibana - mapping

```
PUT1myfile
 "mappings": {
  "myfile":1{
    "properties":1{
     "Country":1
      "type":1"keyword"
     "Item Type":1{
      "type":1"keyword"
    },
    "Unit Price": {
      "type":1"float"
     "Units Sold":1
      "type":1"integer"
```

알아두면 유용한 API

GET1_cat/indices

DELETE 1myfile

GET1myfile/_search

GET1myfile/_mapping

Logstash - input

```
input {
  file {
    path => "/home/ec2-user/data/titanic.csv"
    start_position => "beginning"
    sincedb_path => "/dev/null" #only for develop
  }
}
```

```
input {
  file {
    path => "/home/ec2-user/data/titanic.csv"
    start_position => "beginning"
  }
}
```

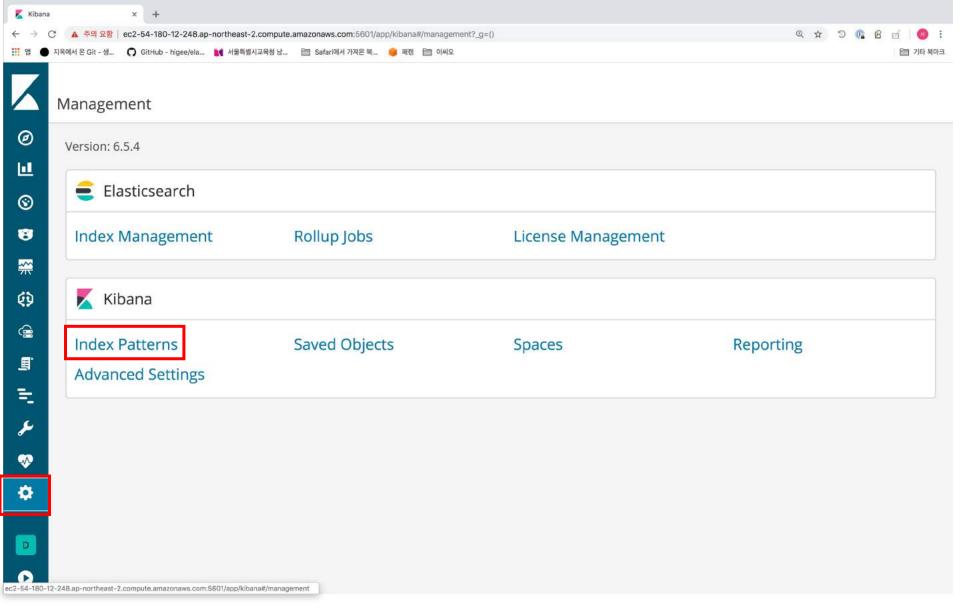
Logstash - filter

```
filter {
 csv {
    separator => ","
   autodetect_column_names => true
 date {
   match => ["Ship Date", "MM/dd/YYYY", "IS08601"]
   timezone => "Asia/Seoul"
    target => "Ship Date"
 mutate {
    remove_field => ["@version", "message", "path"]
```

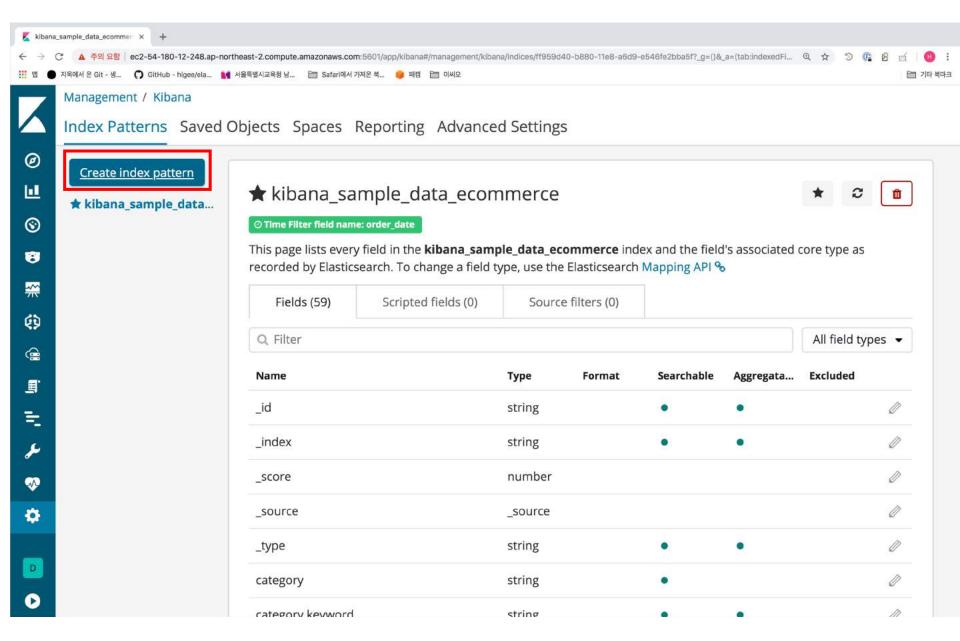
파일원본	Logsatsh timezone	elasticsearch	kibana
2019.01.01 05:11:23	UTC	2019.01.01 05:11:23	2019.01.01 14:11:23
2019.01.01 05:11:23	Asiz/Seoul	2019.12.31 20:11:23	2019.01.01 05:11:23

Logstash - output

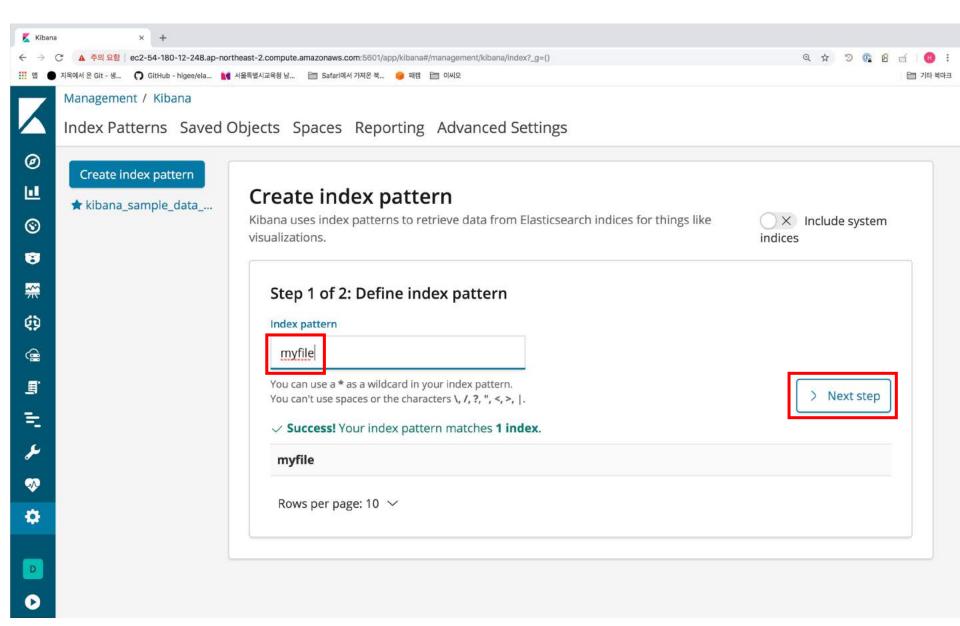
```
output {
  elasticsearch {
    hosts => ["ec2....compute.amazonaws.com:9200"]
    index => "myfile"
    document_type => "myfile"
  }
  stdout { } #only for develop
}
```



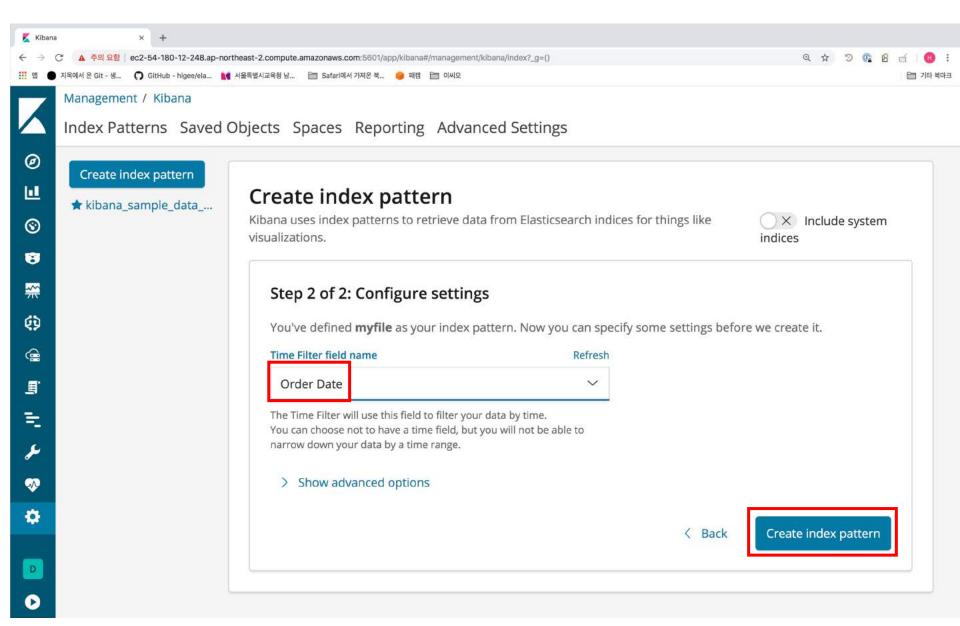












Dashboard 만들기 workflow

- EC2 환경설정 [신효정][5회차] ELK 설치 & Logstash 활용 스크립트.txt -----> 작업 시작
- ELK 설치 [신효정][5회차] ELK 설치 & Logstash 활용 스크립트.txt
- EK 설정 및 실행 [신효정][5회차] ELK 설치 & Logstash 활용 스크립트.txt
- -----> ELK 설치 완료
- E index 구조 잡기 [신효정][6회차] Elastic Stack으로 Dashboard 구축하기 스크립트 1.txt
- L 설정 및 실행 [신효정][6회차] Elastic Stack으로 Dashboard 구축하기 스크립트 1.txt
- -----> index 생성(데이터 수집)

Dashboard 만들기

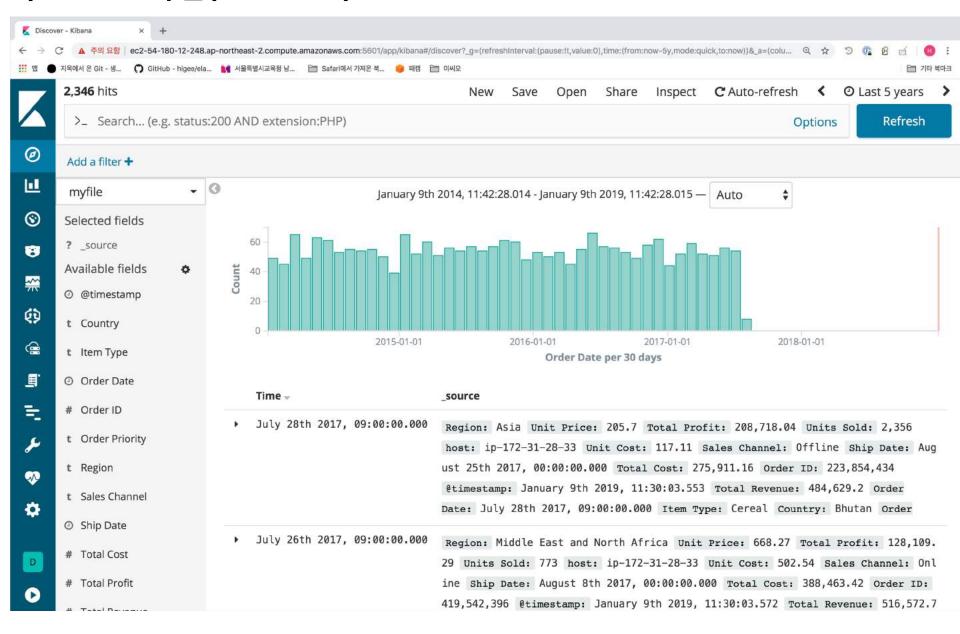
- 1) index 확인(discover)
- 2) dashboard의 목적 정의
- 3) metric 정의
- 4) 각 metric을 위한 agg./Visualize type 선정
- 5) metric별 object 생성 후 dashboard 구축
- -----> dashboard 구축 완료

Feed back

- 1) 수집 데이터 수정
- 2) 'Dashboard 만들기' 수정

1) index 확인(discover)





매출 데이터 분석을 통한 현황 파악

- 구매 특징에 따른 매출 비교 for 마케팅 자료
- Item Type에 따른 order date, ship date 차이 분석 **for 재고 관리**

3) metric 정의



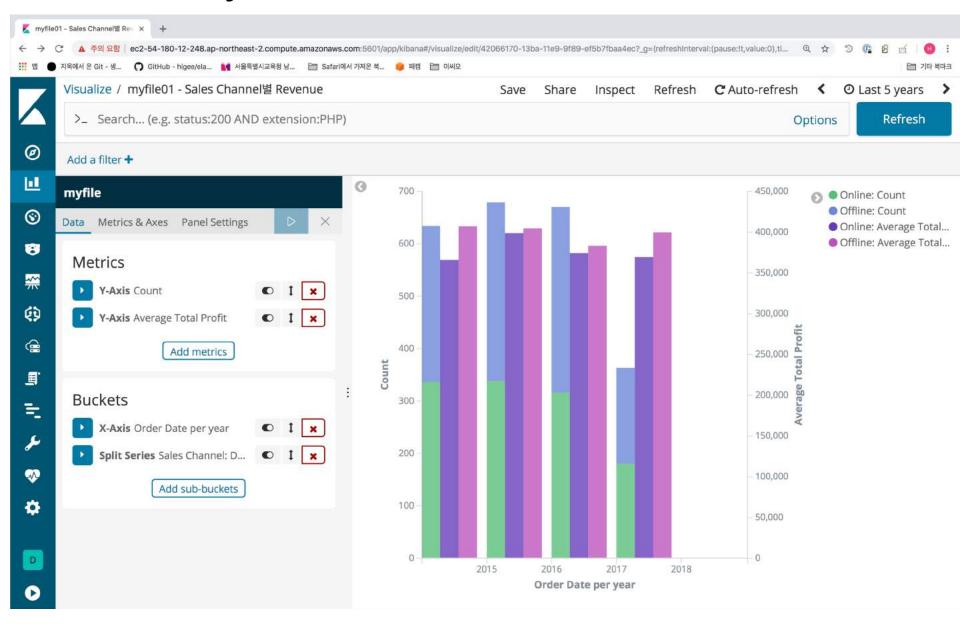
- 매출 데이터 분석을 통한 현황 파악
 - 구매 특징에 따른 매출 비교
 - 1) Sales Channel별 최대 매출/Revenue 등
 - 2) Region/Country별 최대 매출/Revenue 등
 - 3) Item Type별 최대 매출/Revenue 등
 - Item Type에 따른 order date, ship date 차이 분석
 - 1) Sales Channel별 시차
 - 2) Region/Country별 시차
 - 3) Item Type별 시차



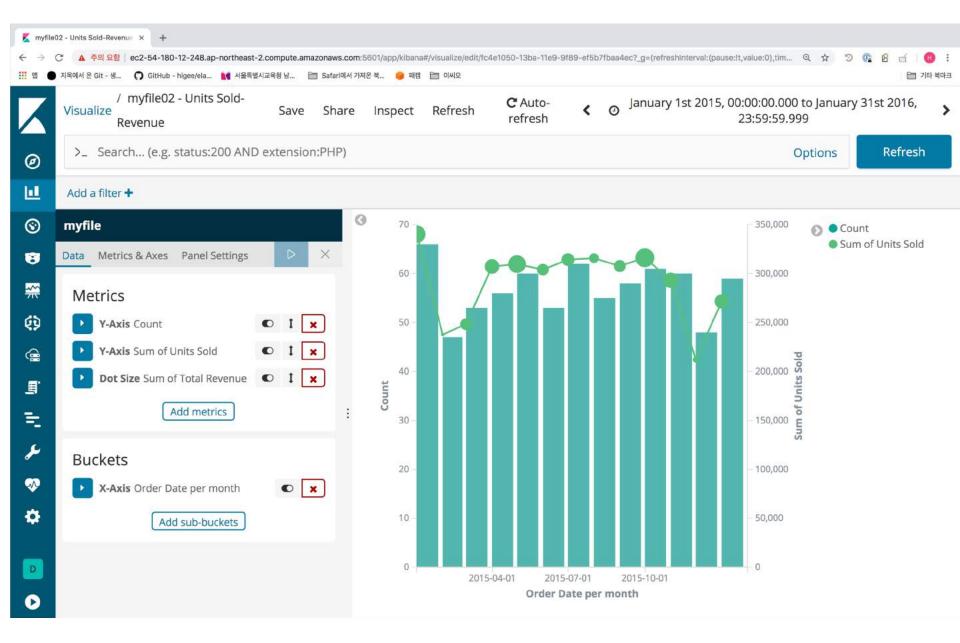
매출 데이터 분석을 통한 현황 파악

- 구매 특징에 따른 매출 비교
 - 1) Sales Channel별 매출/Revenue 등
 - 2) Region/Country별 매출/Revenue 등
 - 3) Item Type별 매출/Revenue 등
 - → term agg, sum/min/max/avg agg
- Item Type에 따른 order date, ship date 차이 분석
 - 1) Sales Channel별 시차
 - 2) Region/Country별 시차
 - 3) Item Type별 시차
 - → term agg, std agg, data table

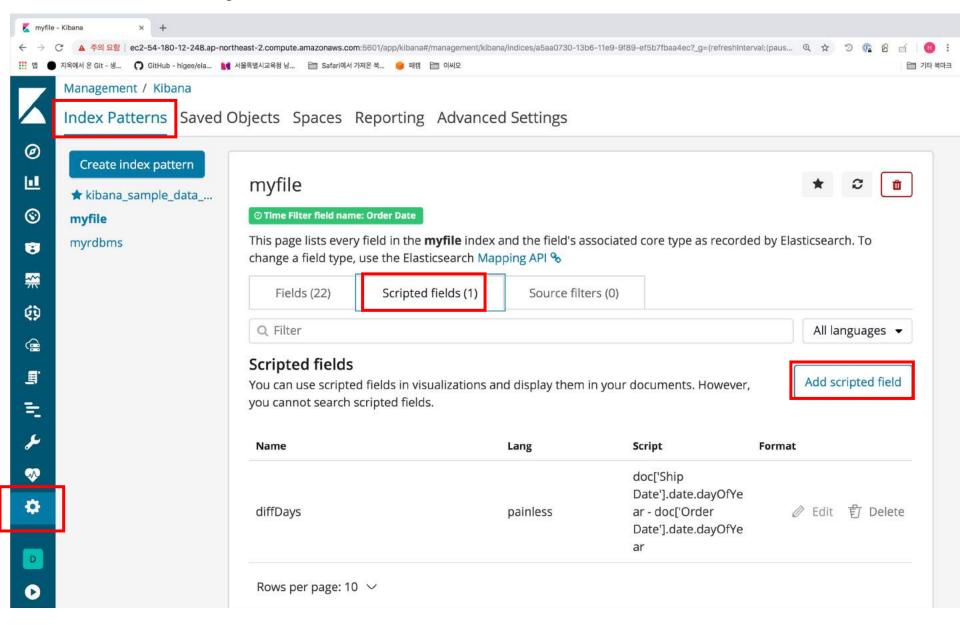




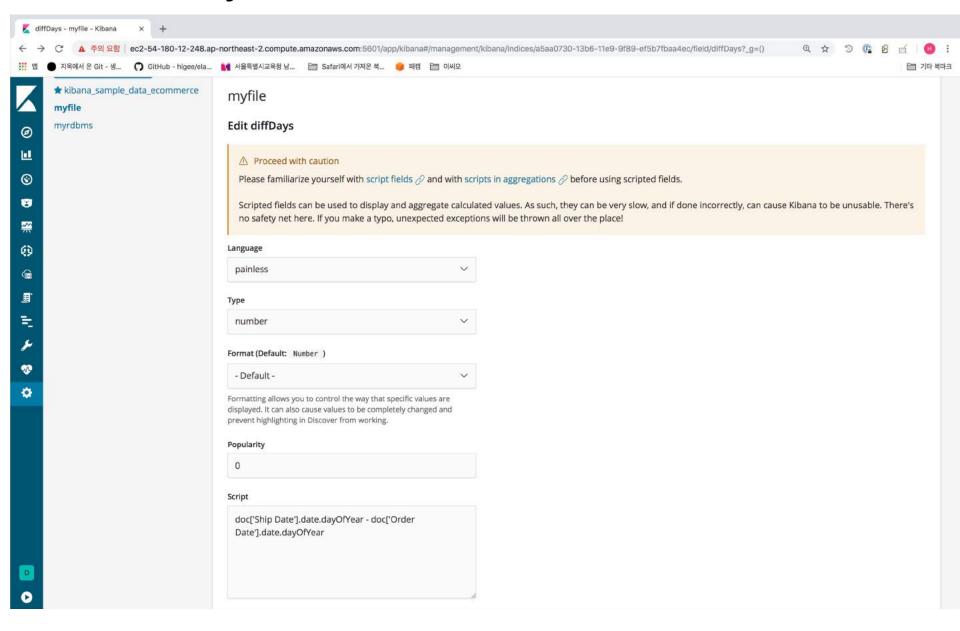




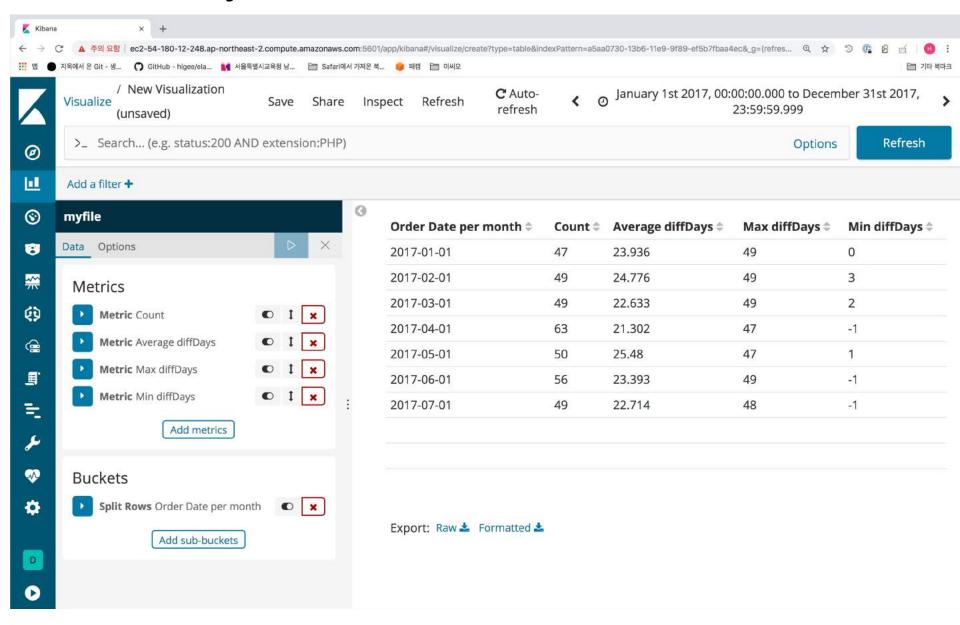




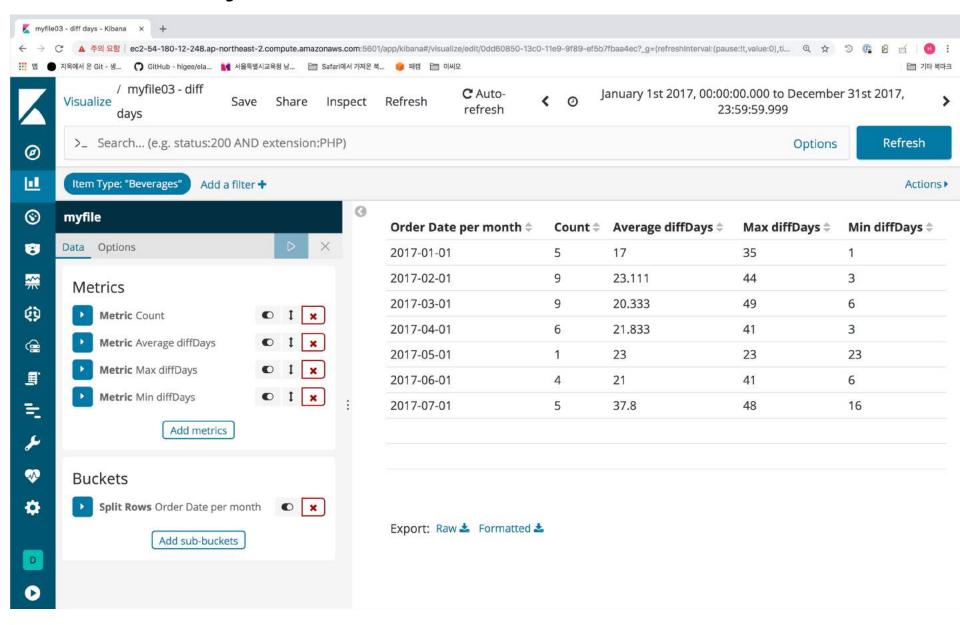












mySQL to ES

Dashboard 만들기 workflow

- EC2 환경설정 [신효정][5회차] ELK 설치 & Logstash 활용 스크립트.txt -----> 작업 시작
- ELK 설치 [신효정][5회차] ELK 설치 & Logstash 활용 스크립트.txt
- EK 설정 및 실행 [신효정][5회차] ELK 설치 & Logstash 활용 스크립트.txt
- -----> ELK 설치 완료
- E index 구조 잡기 [신효정][6회차] Elastic Stack으로 Dashboard 구축하기 스크립트 2.txt
- L 설정 및 실행 [신효정][6회차] Elastic Stack으로 Dashboard 구축하기 스크립트 2.txt
- -----> index 생성(데이터 수집)

Dashboard 만들기

- 1) index 확인(discover)
- 2) dashboard의 목적 정의
- 3) metric 정의
- 4) 각 metric을 위한 agg./Visualize type 선정
- 5) metric별 object 생성 후 dashboard 구축
- -----> dashboard 구축 완료

Feed back

- 1) 수집 데이터 수정
- 2) 'Dashboard 만들기' 수정

정보확인



ES로 입력할 mysql의 정보(logstash가 실행 될 서버의)

- server ip: ec2-52-78-134-20.ap-northeast-2.compute.amazonaws.com

- user : root

- password : password

- db name : fast

- table name : titanic

Elasticsearch가 실행되고 있는 서버의 IP주소

- ec2-54-180-12-248.ap-northeast-2.compute.amazonaws.com

- 1.survival Survival (0 = No; 1 = Yes)
- 2.class Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd)
- 3.name Name
- 4.sex Sex
- 5.age Age
- 6.sibsp Number of Siblings/Spouses Aboard
- 7.parch Number of Parents/Children Aboard
- 8.ticket Ticket Number
- 9.fare Passenger Fare
- 10.cabin Cabin
- 11.embarked Port of Embarkation (C = Cherbourg; Q =

Queenstown; S = Southampton)

Kibana - mapping

```
PUT myrdbms
 "mappings": {
  "myfile": {
    "properties": {
     "age": {
      "type": "integer"
     },
     "sibsp": {
      "type": "integer"
     "survived": {
      "type": "keyword"
     "ticket": {
      "type": "keyword"
```

알아두면 유용한 API

GET _cat/indices

DELETE myrdbms

GET myrdbms/_search

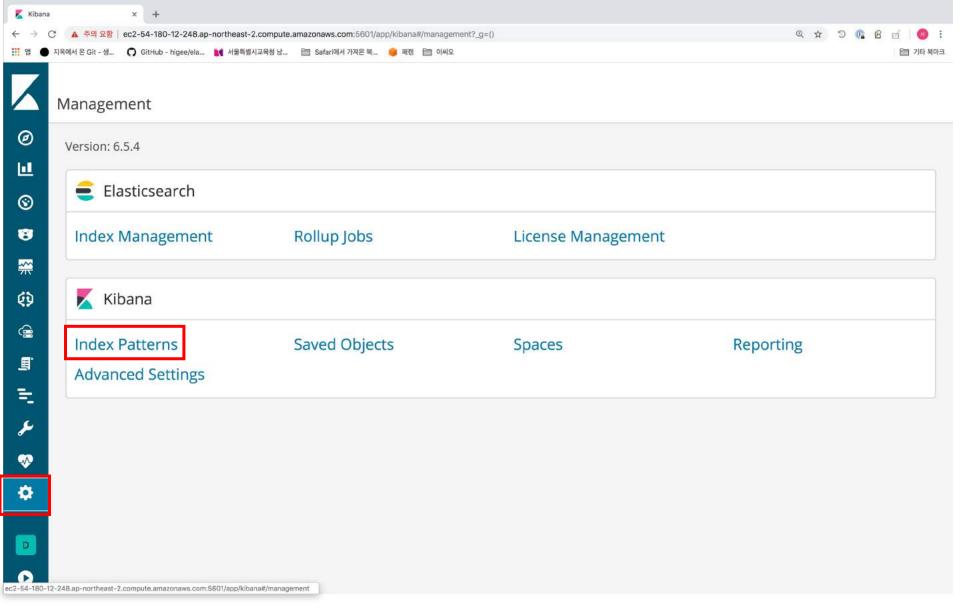
GET myrdbms/_mapping

Logstash - input

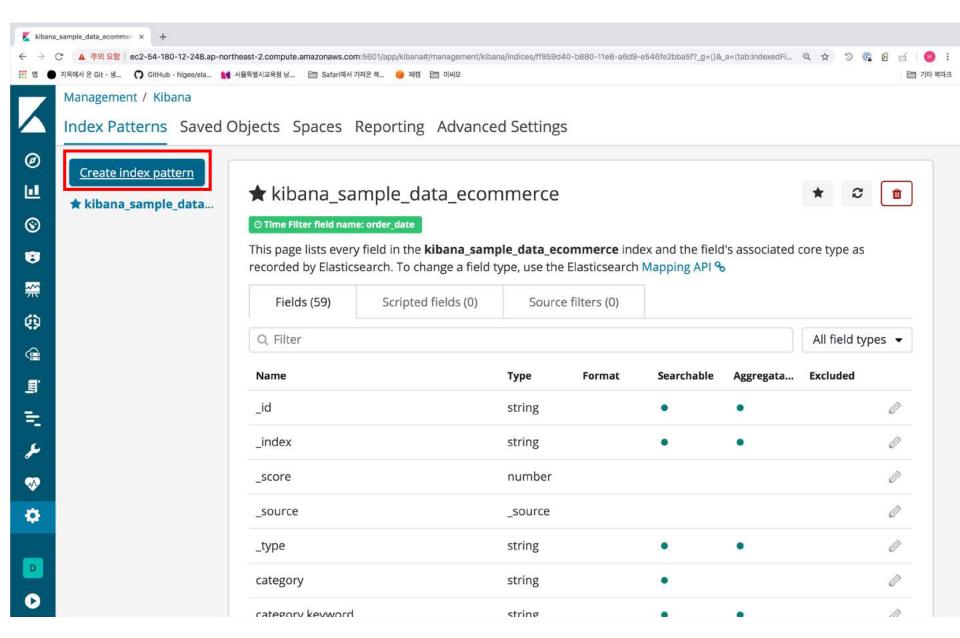
```
테스트 시간 단축 statement => "SELECT * FROM titanic limit 10"
```

Logstash - output

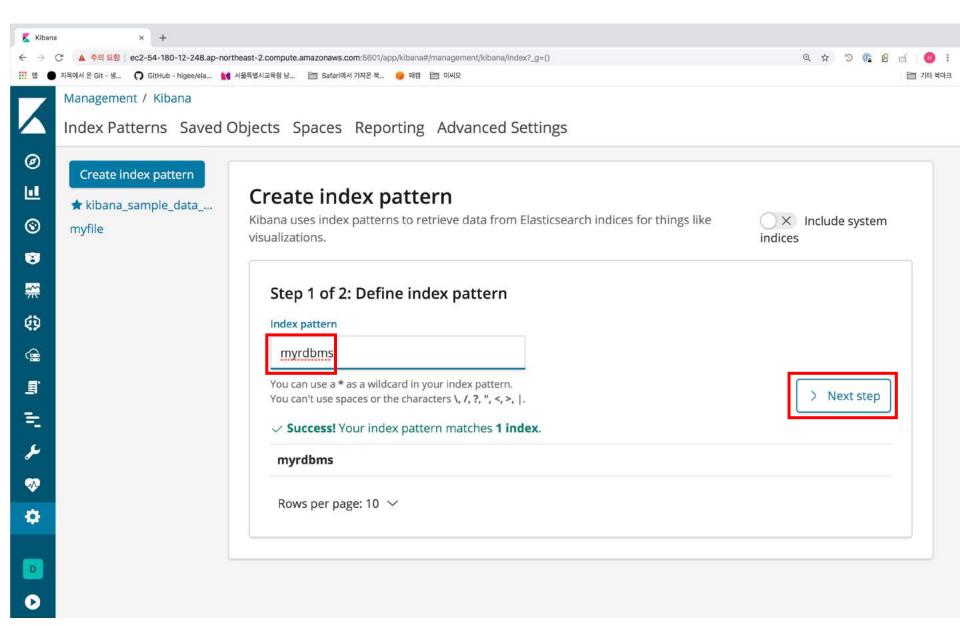
```
output {
   elasticsearch {
     hosts => ["ec2....compute.amazonaws.com:9200"]
     index => "myrdbms"
     document_type => "myrdbms"
   }
   stdout { } #only for develop
}
```



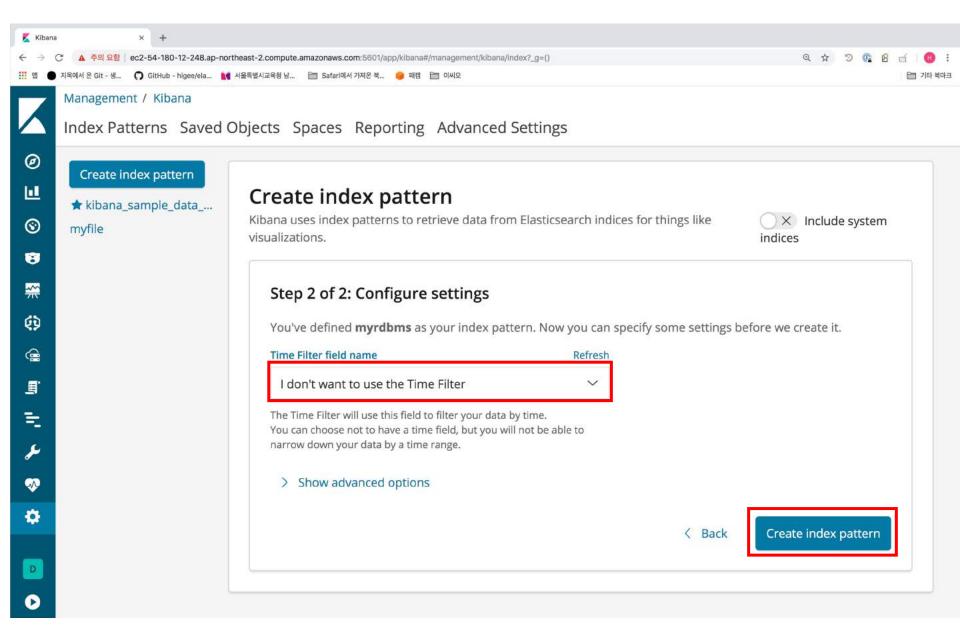












Dashboard 만들기 workflow

- EC2 환경설정 [신효정][5회차] ELK 설치 & Logstash 활용 스크립트.txt -----> 작업 시작
- ELK 설치 [신효정][5회차] ELK 설치 & Logstash 활용 스크립트.txt
- EK 설정 및 실행 [신효정][5회차] ELK 설치 & Logstash 활용 스크립트.txt
- -----> ELK 설치 완료
- E index 구조 잡기 [신효정][6회차] Elastic Stack으로 Dashboard 구축하기 스크립트 2.txt
- L 설정 및 실행 [신효정][6회차] Elastic Stack으로 Dashboard 구축하기 스크립트 2.txt
- -----> index 생성(데이터 수집)

Dashboard 만들기

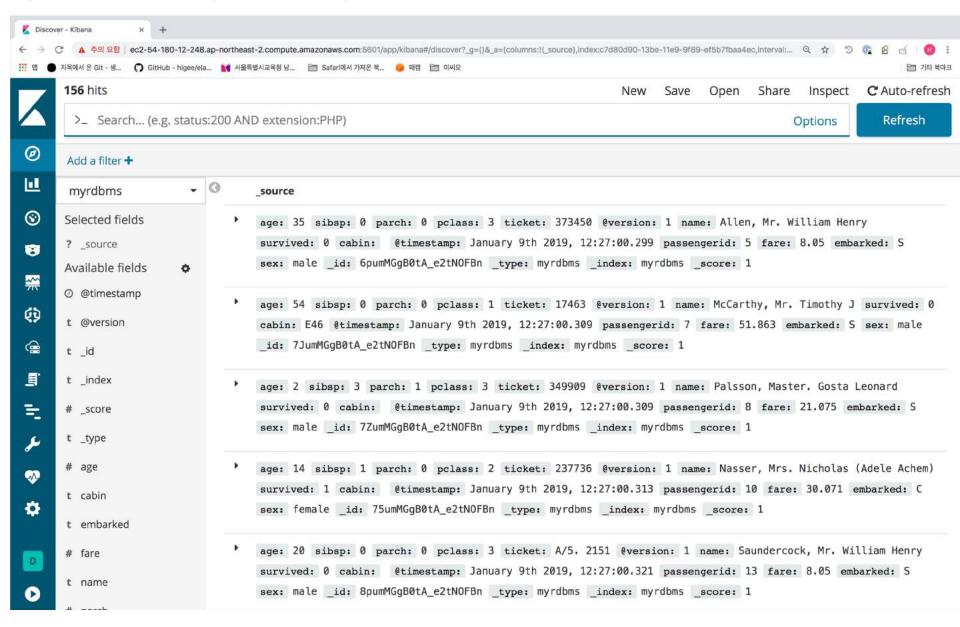
- 1) index 확인(discover)
- 2) dashboard의 목적 정의
- 3) metric 정의
- 4) 각 metric을 위한 agg./Visualize type 선정
- 5) metric별 object 생성 후 dashboard 구축
- -----> dashboard 구축 완료

Feed back

- 1) 수집 데이터 수정
- 2) 'Dashboard 만들기' 수정

1) index 확인(discover)





2) dashboard의 목적 정의

Fast campus

타이타닉 승객 특성에 따른 생존율 분석 - 승객 특징에 따른 생존율 비교

3) metric 정의

Fast campus

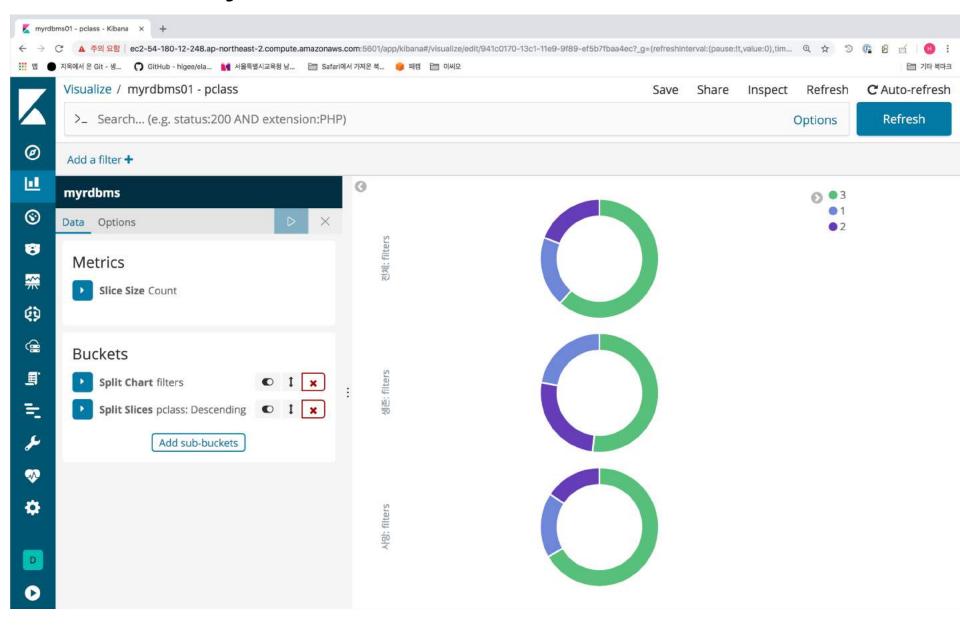
타이타닉 승객 특성에 따른 생존율 분석

- 승객 특징에 따른 생존율 비교
 - 1) pclass에 따른 생존율
 - 2) age에 따른 생존율
 - 3) sex에 따른 생존율
 - 4) fare에 따른 생존율

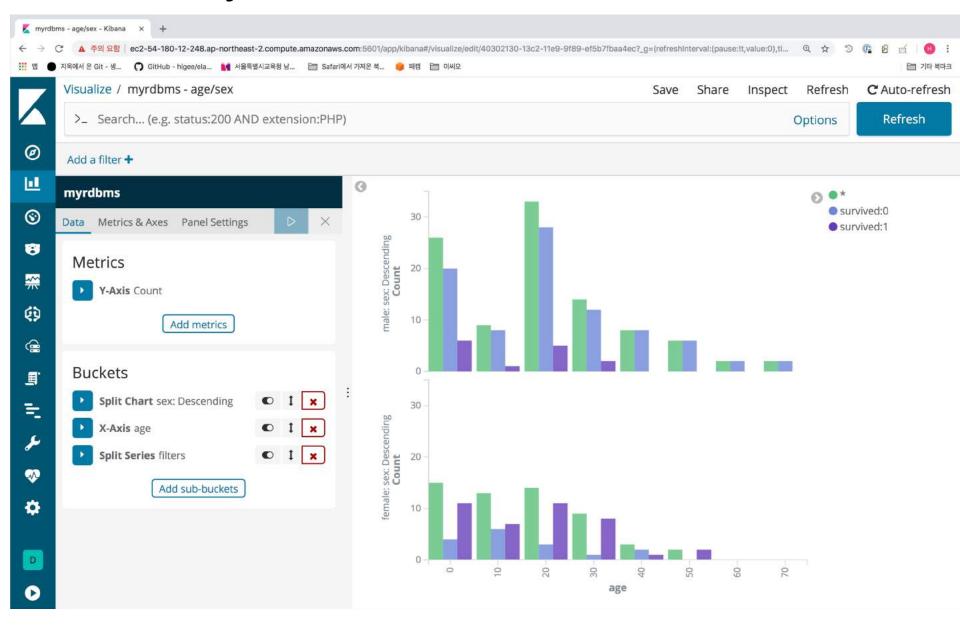
4) 각 metric을 위한 agg./Visualize type 선정

- 타이타닉 승객 특성에 따른 생존율 분석
 - 승객 특징에 따른 생존율 비교
 - 1) pclass에 따른 생존율
 - 2) age에 따른 생존율
 - 3) sex에 따른 생존율
 - 4) fare에 따른 생존율
 - → term agg, range agg, cnt agg.









감사합니다.