II 키바나(kibana)

비즈니스 인텔리전스(Business Intelligence; BI) 도구는 많은 양의 비정형적인 데이터를 수집하고 처리해서 보고서나 대시보드 형태로 데이터 시각화를 지원하므로 의사결정과정에서 통찰력을 얻을 수 있도록 해 준다.

앞어 우리는 비츠와 로그스태시를 이용해 데이터를 수집, 가공하고 엘라스틱서치에 가공된 데이터를 저장하는 방법을 살펴보았다. 이제 간단하고 직관적인 방법으로 데이터 시각화를 가능하게 해 주는 키바니에 대하여 자세히 알아보고자 한다. 키바나는 엘라스틱서치의 쿼리나 집계를 활용해 데이터를 처리하므로 제대로 활용하려면 엘라스틱 서치의 쿼리 DSL, 버킷, 메트릭 집계와 같은 핵심 개념을 기본적으로 이해하고 있어야 한다.

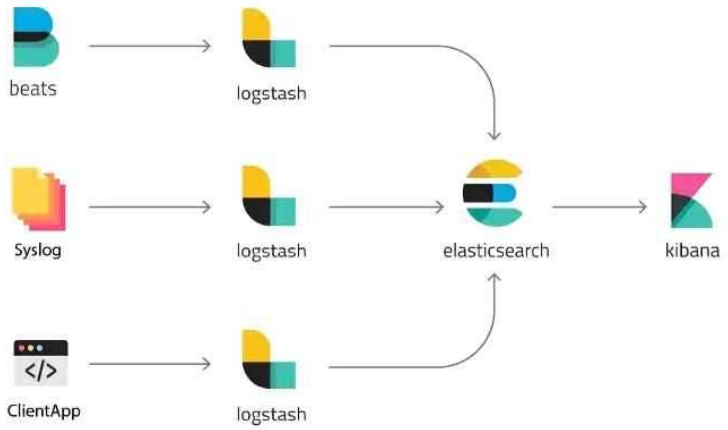
여러 인덱스를 통합해서 하나로 탐색이 가능하게 만드는 인덱스 패턴을 시작으로 데이터를 탐색하는 디스커버, 시각화를 위한 타입정리, 대시보드와 캔버스를 활용한 화면 표시, 서울 지역의 우편번호 데이터를 사용한 지도 표시를 살펴보자

① 키바나 소개

키바나의 대표적인 기능은 다음과 같다.

키바나 기능	설명
데이터 분석과 시각화 툴	오픈소스 기반의 데이터 탐색 및 시각화 도구 제공
엘라스틱 관리	보안, 스냅샷, 인덱스 관리, 개발자 도구 등을 제공
엘라스틱 중앙 허브	모니터링을 비롯해 엘라스틱 솔루션을 탐색하기 위한 포털

시각화 관점에서 키바나의 역할을 살펴보자

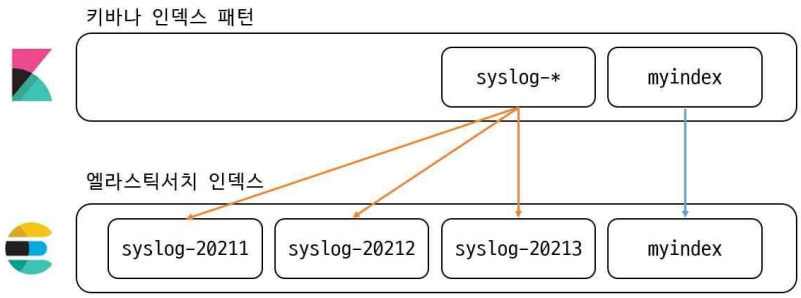


키바나는 시스템의 끝단에서 사용자에게 정보를 효율적으로 제공한다. 엘라스틱서치에서 제공되는 시계열 데이터나 위치 분석 같은 다양한 분석 결과를 키바나를 통해 시각화한다. 유료 서비스가 필요한 머신러닝이나 네트워크 그래프 기능도 있지만 대부분의 기능은 무료인 베이직 라이센스 수준에서 제공된다. 대표적인 시각화 기능은 다음과 같다.

시각화 기능	설명
디스커버(Discover)	데이터를 도큐먼트 단위로 탐색해 구조와 관계 등을 확인할 수 있다
시각화(Visualize)	다양한 그래프 타입으로 데이터 시각화를 할 수 있다
대시보드(Dashboard)	그래프, 지도 등을 한곳에서 확인하면서 다양한 인사이트를 얻을 수 있다
캔버스(Canvas)	그래프와 이미지 등을 프레젠테이션 슬라이드처럼 구성할 수 있다
맵스(Maps)	위치 기반 데이터를 지도 위에 표현할 수 있다

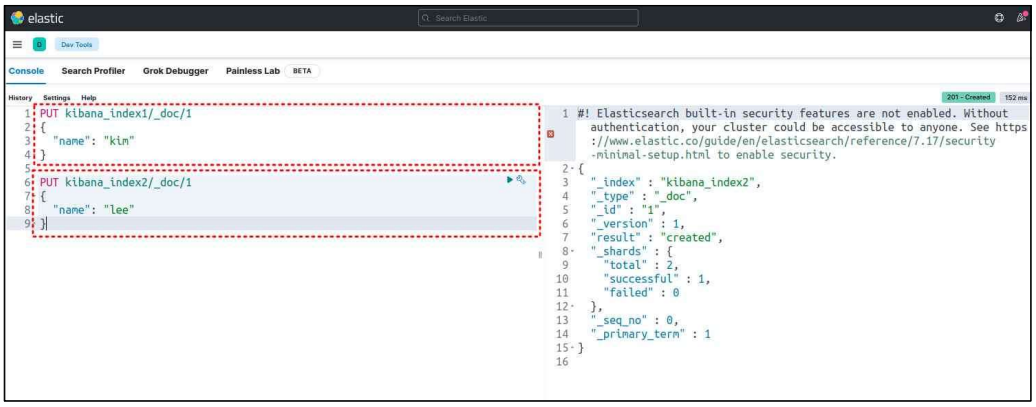
■ 인덱스 패턴

키바나에서 시각화를 하기 위해서는 반드시 엘라스틱서치 인덱스에 연결되어야 한다. 그래서 키바나를 범용적인 시각화 툴로 사용하기에는 무리가 있다. 하지만 오픈소스이면서 다양한 그래프와 지도를 지원하고 빅데이터 처리가 가능하다는 장점이 있다. 키바나는 데이터 소스를 엘라스틱서치 인덱스에서 가져오는데 이를 인덱스 패턴이라고 한다. 키바나 인덱스 패턴은 인덱스 매핑 정보 등을 키바나에 사용하기 적합하게 미리 캐싱해 둔것으로 여러 개의 인덱스에 대한 메타데이터를 병합해 저장해 두었다가 검색이나 시각화 생성 시 활용하게 된다.

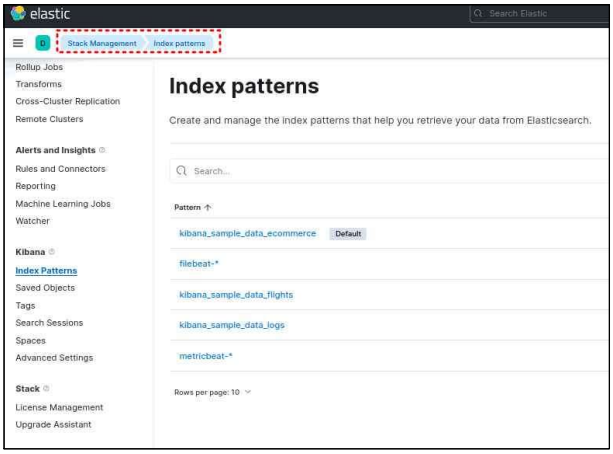


syslog-* 라는 키바나 인덱스 패턴을 보자. 이는 키바나에서 활요하기 쉽도록 필드 포맷 등을 캐싱해둔 메타데이터로, 엘라스틱 서치의 syslog-XXXXX 인덱스들과 연결되어 있다. 키바나에서 엘라스틱서치 인덱스에 직접 접근하지 않고 syslog-* 인덱스 패턴에 접근해 쿼리를 하고 시각화한다. 그렇다면 키바나는 엘라스틱서치 인덱스에 직접 접근하지 않고 인덱스 패턴 구조를 한 단계 더 거치는 것일까? 인덱스 패턴을 만드는 이유는 복수의 인덱스에 대한 매핑을 사전에 병합해 두어 쿼리 생성이나 시각화에 활용할 수 있기 때문이다.

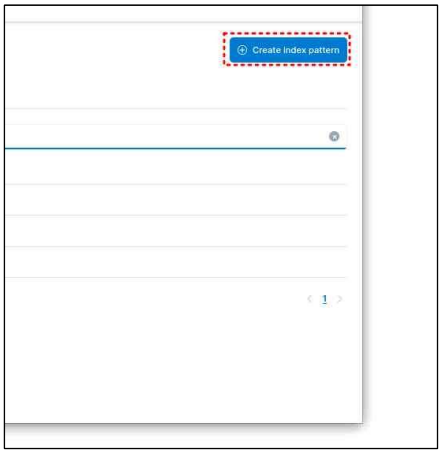
키바나에서 인덱스 패턴을 아래와 같이 생성한다.



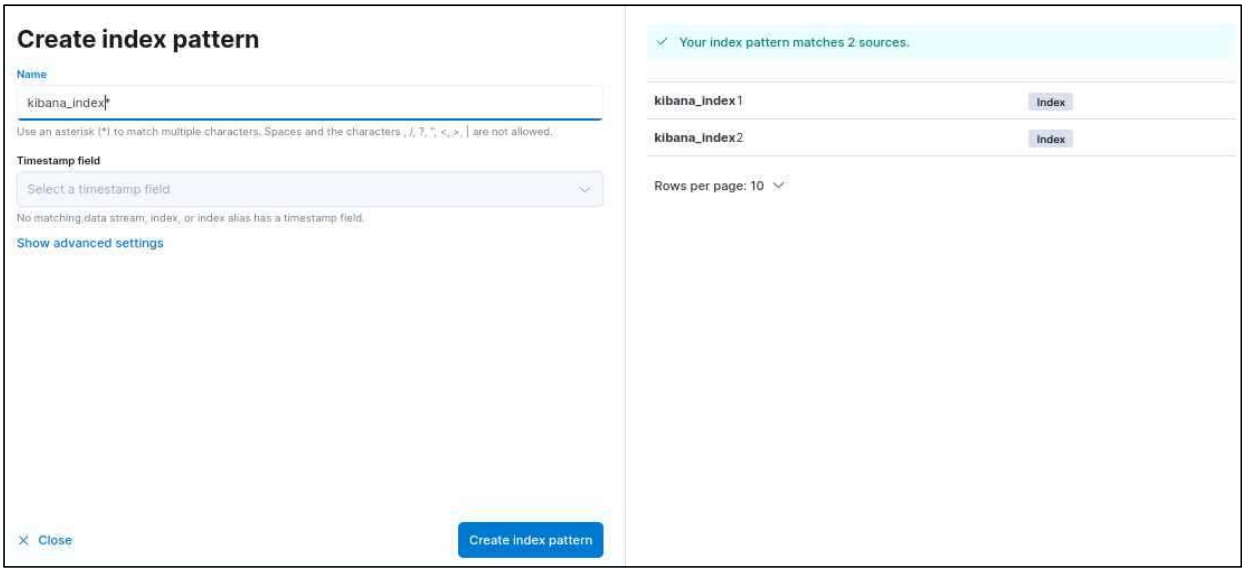
이제 키바나의 Discover, Visulize 메뉴에 들어가보자. 키바나에서는 인덱스 패턴을 통해서만 엘라스틱 서치 인덱스에 접근할 수 있다. 키바나는 엘라스틱스처 인덱스가 아니라 키바나 인덱스 패턴을 데이터 소스로 인식하기 때문이다. 이제 키바나 인덱스 패턴을 만들어보자.



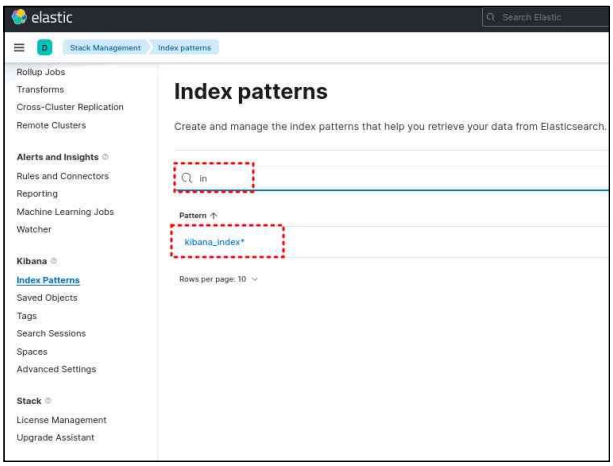
〈인덱스 패턴 생성 준비1〉



〈인덱스 패턴 생성 준비2〉



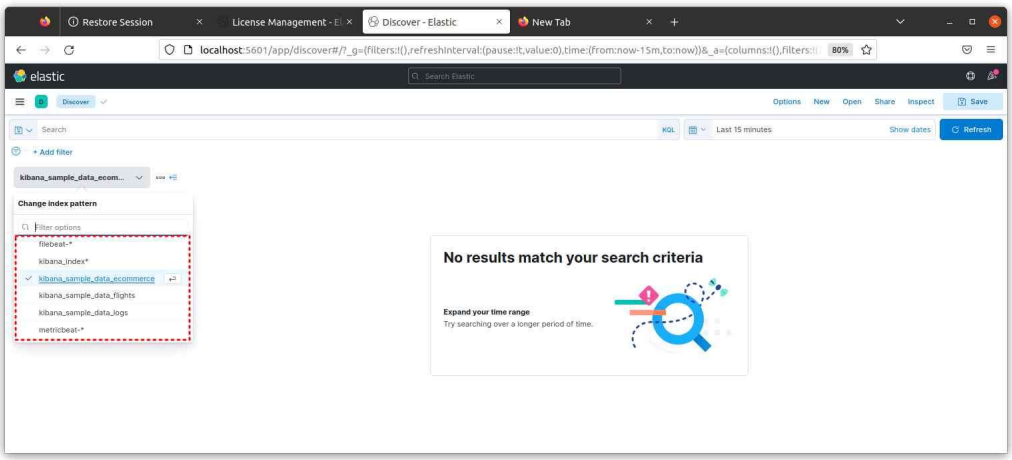
kibana_index* 로 작성하면 kibana_index 로 시작하는 모든 인덱스가 키바나 인덱스 패턴과 연결된다. "Create index pattern" 을 클릭하여 인덱스 패턴을 생성한다.



〈Discover 메뉴에서 인덱스 패턴 확인〉

② 디스커버

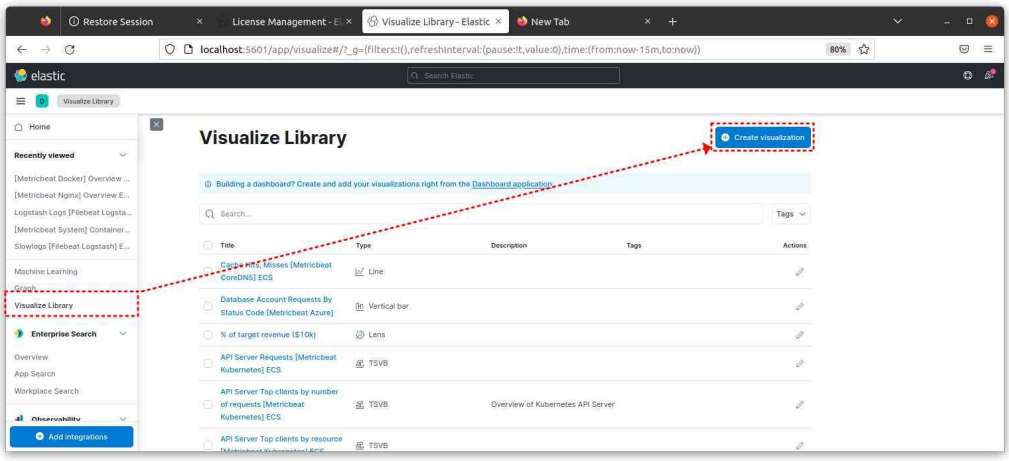
이제 키바나에서 시각화 메뉴들을 하나씩 살펴보자. 먼저 엘라스틱에서 제공하는 샘플 데이터 3개 (kibana_smaple_data_ecommerce, kibana_sample_data_flights, kibana_sample_data_logs) 가 로드되어 있는지 여부를 미리 확인해 두어야 한다. 샘플데이터를 로드하면 인덱스를 생성함과 동시에 인덱스 패턴도 만들어 준다. Discover 에서 아래와 같은 화면을 확인한다.



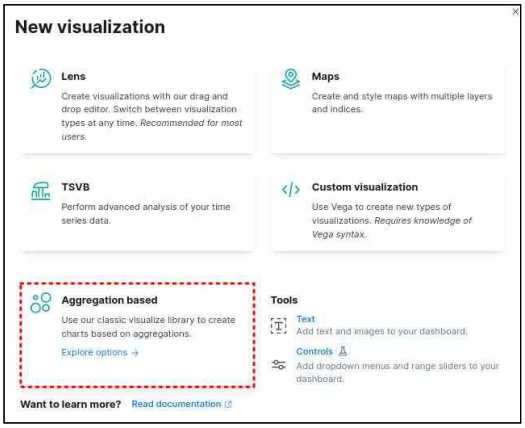
③ 시각화

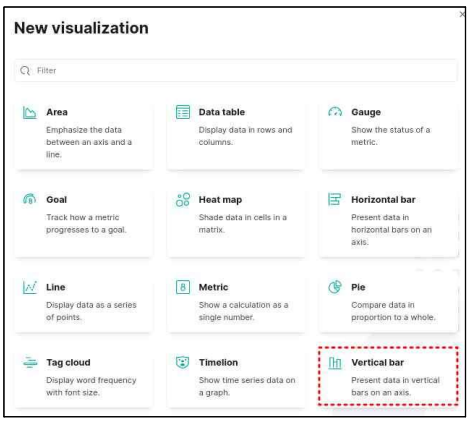
Visualize 메뉴는 엘라스틱서치에 저장된 데이터를 그래프나 표, 지도 등 다양한 타입으로 보여주는 역할을 한다. 라인, 바, 파이 차트부터 맵, 시계열 비주얼 빌더(Time Series Visual Builder, TSVB), 태그 클라우드 등 다양한 시각화 타입을 지원한다. 데이터를 가장 효과적으로 보여줄 수 있는 방법을 떠올리고 그에맞는 타입을 선택하면 된다.

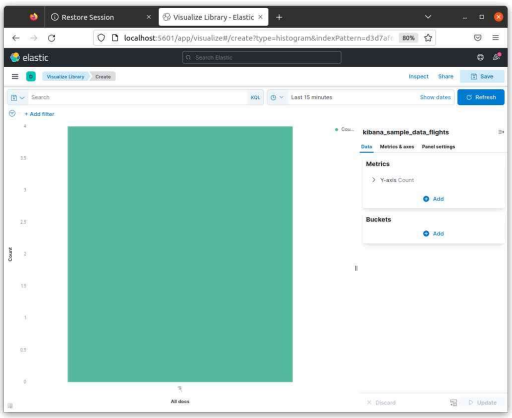
간단한 그래프를 하나 만들면서 키바나와 엘라스틱서치의 집계를 이용하여 시각화 해 보자
먼저 키바나 메뉴의 "Visualize Library" 에서 "Create visualization" 을 클릭한다.



아래와 같은 화면이 보이면 "Aggregation based" 를 클릭한 뒤, "Vertical Bar" 를 클릭한다.





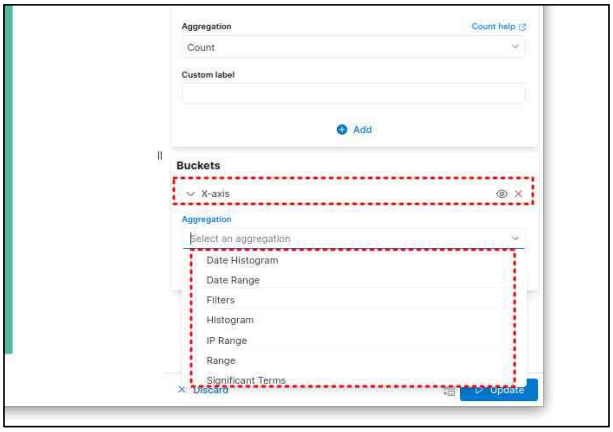


<kibana_sample_data_flights 선택>

이제 kibana_sample_data_flights 데이터를 가지고 시각화를 해볼 수 있다. 키바나는 다양한 종류의 시각화 타입을 지원하는데 대표적으로 막대그래프, 히트맵, TSVB 를 이용할 수 있다.

■ 막대 그래프

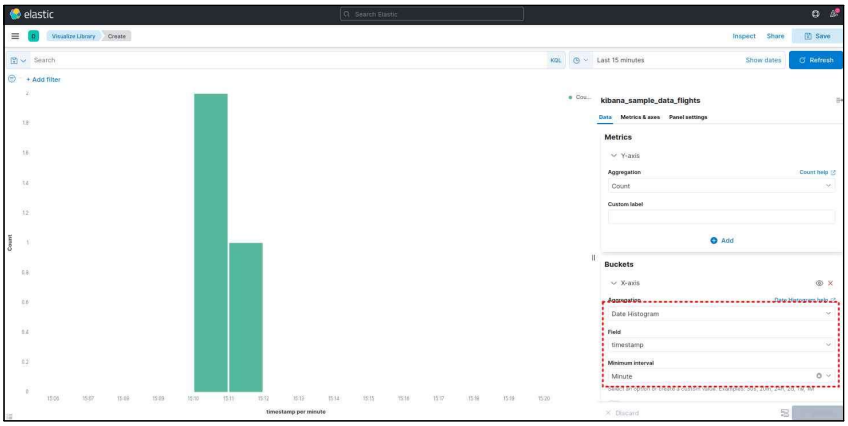
먼저 Metrics 와 Buckets 을 설정해야 한다.Metrics 는 평균값/최소값/최대값 같은 통계를 보여주고 그래프 상에서 Y 축에 속한다. Buckets 은 특정 기준으로데이터를 나누는 역할을 하며 그래프상에서 X 축에 속한다. 먼저 Buckets 설정을 해본다. Buckets 하단에 있는 +Add 버튼을 클릭하면 세가지 메뉴가 나오는데 X-axis 는 X축을 의미한다. split series 는 서브 버킷 용도로 사용되고, split chart 는 그래프를 버킷 기준으로 쪼개서 보여줄 때 쓰인다. X-axis 를 선택한다.



위의 그림에서 추가적으로 나오는 Aggregation 은 X 축을 어떤 기준으로 나눠야 하는지 정하는 화면이다. 버킷 집계 종류는 다음과 같다.

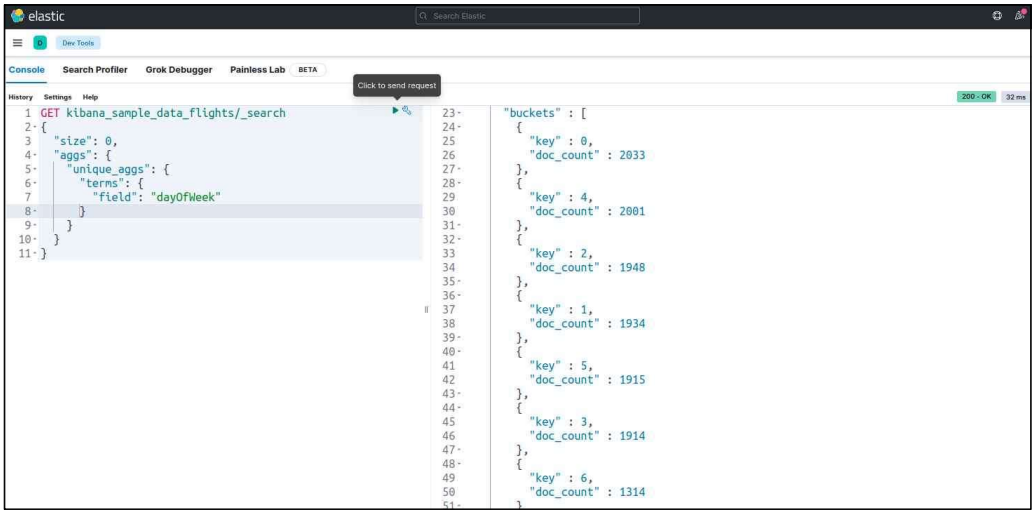
버킷 집계	설명
Data Histogram	날짜/시간 데이터 타입을 가진 필드만 사용 가능. 일정한 주기를 기준으로 버킷 구분
Date Range	날짜/시간 데이터 타입을 가진 필드만 사용 가능. 사용자가 임의범위를 지정해 버킷 구분
Filters	필터 적용 가능
Histogram	일정한 주기를 기준으로 버킷을 구분함
Ipv4 Range	ip 타입을 가진 필드만 사용가능. ip 범위를 임의 지정해 버킷을 구분함
Range	사용자가 임의의 범위를 지정해 버킷을 구분함
Significant Terms	필드의 유니크한 값 중 통계적으로 의미 있는 용어를 기준으로 구분함
Terms	필드의 유니크한 값을 기준으로 구분함

Date Histogram 을 선택해보자. 날짜 히스트그램 집계는 날짜/시간 데이터 타입을 가진 필드가 있을 때 일정한 날짜/시간 간격으로 버킷을 생성한다.

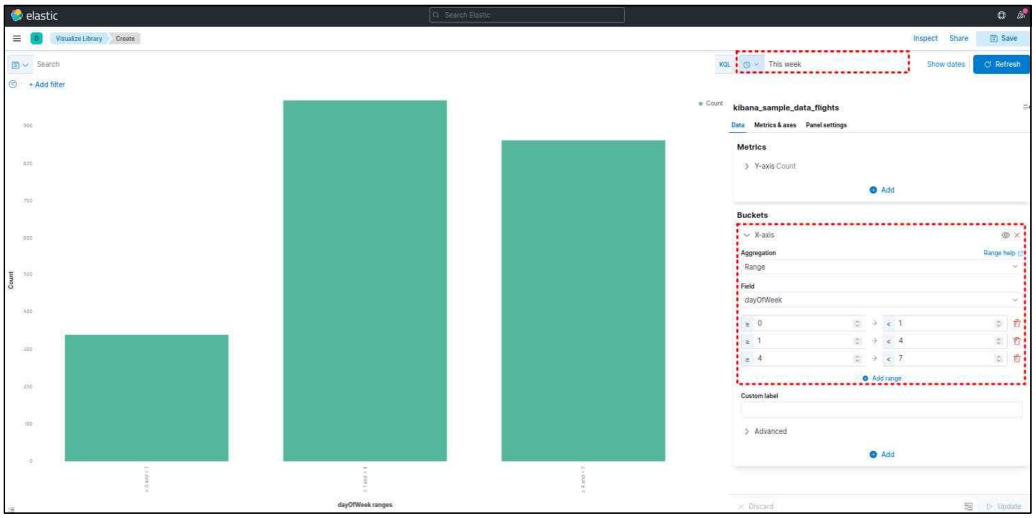


kibana_sample_data_flights 인덱스 패턴에는 timestamp 라는 날짜/시간 필드가 있다. Field 에 timestamp 를 선택하고 Minimum interval 을 Minute 로 설정하고 하단의 Update 버튼을 누른다.

다음은 Aggregation 에서 Range 를 사용한다. Range 는 사용작 임의로 간격을 지정해 버킷을 생성하는 것으로 사용자가 데이터값의 범위를 대략적으로 알고 있어야 한다. 범위 집계에 사용할 필드의 데이터 범위를 확인하기 위해 키바나 콘솔을 이용하자.



이제 dayOfWeek 필드에 범위를 직접 지정해 주고 막대그래프를 그려보자 구간은 3구간으로 나눈다.

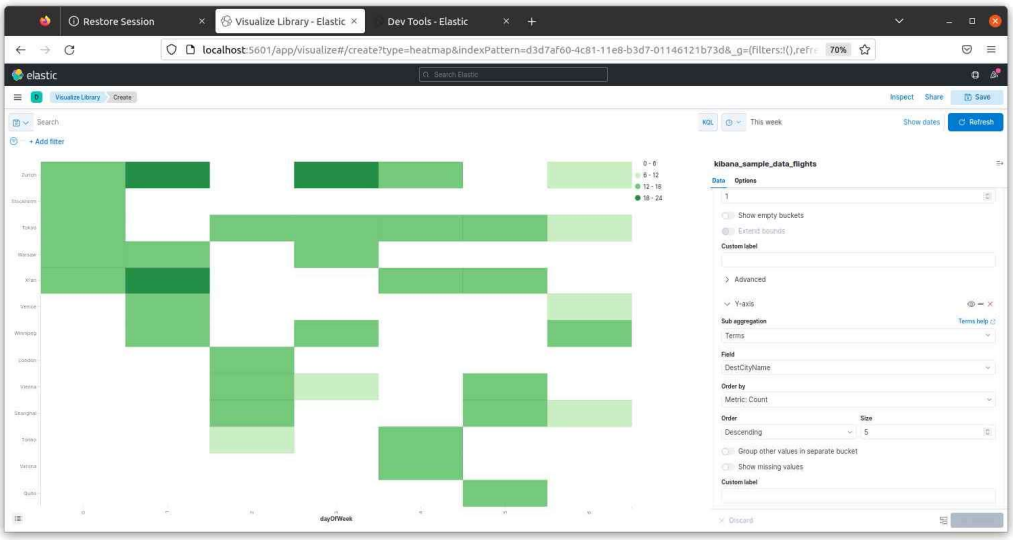


■ 히트맵

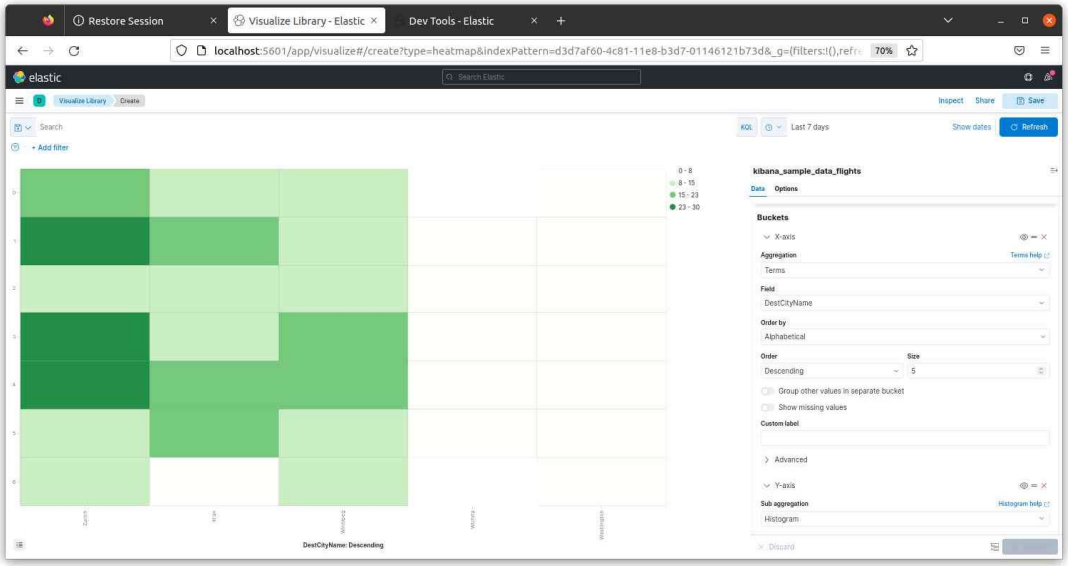
히트맵은 열분포 형태의 시각화 표현방식이다. X,Y 축을 2개의 버킷으로 생성하고 메트릭값을 색의 진하기로 표현함으로써 3차원 데이터 처리가 가능하다. 색의 종류, 그라데이션 깊이 등은 설정을 통해 수정할 수 있다. 요일별 비행기 목적지로 가장 많이 가는 도시가 어디인지 확인하기 위해 시각화를 생성해보자.

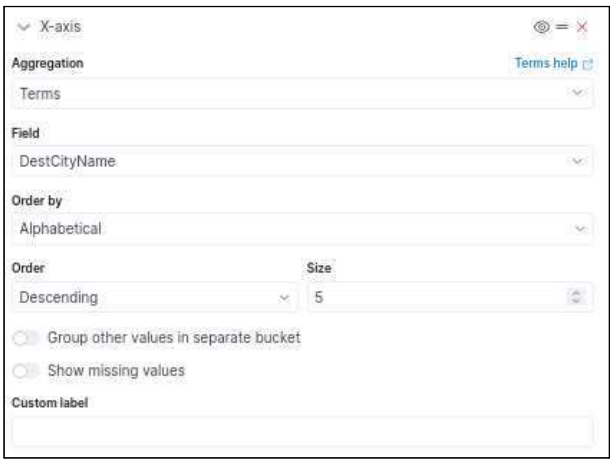
먼저 새로운 시각화 타입을 만드는데 Heat map 을 선택하고 인덱스 패턴은 "kibana_sample_data_flights" 를 선택한다. X축은 요일, Y축은 도시를 표현하자. Buckets 에서 +Add 버튼을 누르고 X-axis 를 선택한 뒤, Aggregation 은 Histogram, Field 는 dayOfWeek, Interval 은 1로 설정한다.
Y 축은 도시명이다. Bucket 에서 +Add 를 눌러 Y-axis 를 선택한 뒤, Sub aggregation 은 Terms, Field 는 DestCityName, Order by 는 Alphabetical 로 지정하고 Order 는 Descending(내림차순), Size 는 5 를 지정해 상위 5개만 보이도록 한다.

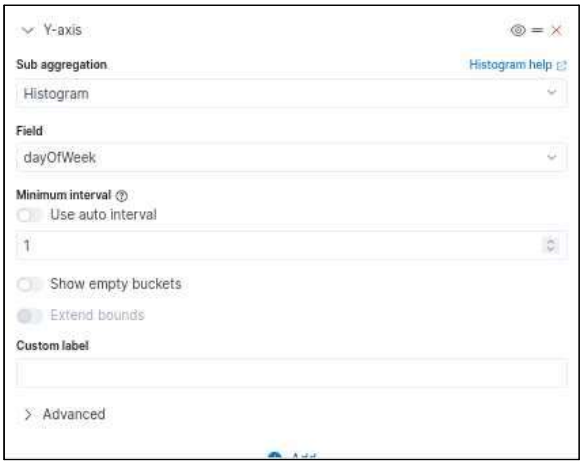
Z 축은 색상이다. 단순 도시 수이기 때문에 Count 메트릭을 사용하면 되므로 특별히 수정할 필요는 없다. 설정이 끝나면 Update 를 눌러 결과를 확인해 본다.(일부 옵션선택을 잘못한 화면입니다 ㅜㅜ)



하지만 중간에 비어있는 일정으로 인해 Y 축이 너무 길어지는 문제가 발생할 수 있다. 히트맵은 X,Y 축이 고정되지 않으면 자동으로 늘어지는 문제가 있다. 설정을 변경해 보자. X 축과 Y 축을 변경한 결과이다.





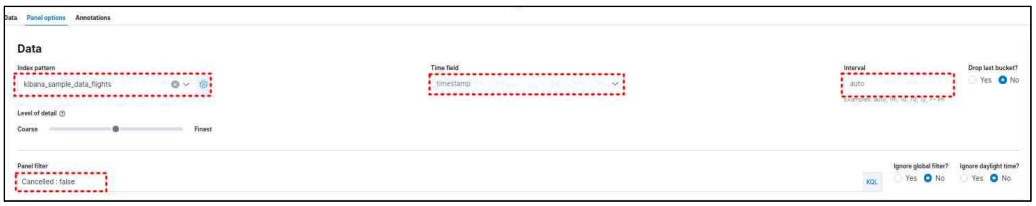


■ TSVB(Time Series Visual Builder)

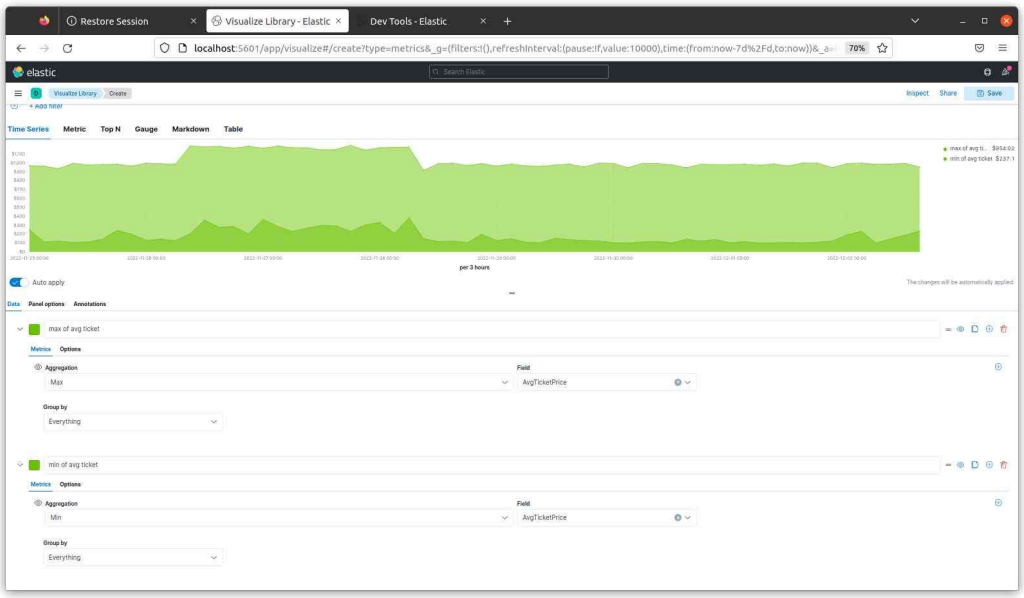
시계열 데이터를 처리하기 위한 메뉴로, 로그 모니터링이나 시간 범위 내의 특정 동작을 시각화하는 데 유용하다. TSVB 를 통해 시계열 데이터, 통계 정보, n 번째 상위 값, 게이지 등을 편하게 확인할 수 있다. TSVB 가 어떤 분석에 인사이트를 줄 수 있는지 직접사용해 보자.

새로운 시각화를 하나 만들고 타입을 TSVB 를 선택한다.

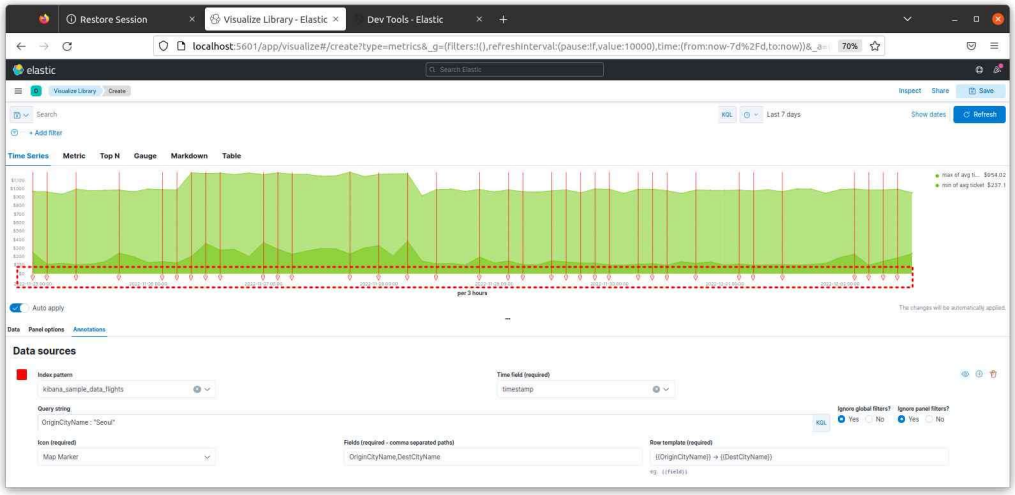
상단 인터페이스에 6개의 서비스가 보인다. Time Series 는 시계열 데이터를 히스토그램 형태로 확인할 수 있고 나머지 5개의 서비스는 특별한 타입으로 데이터 정보를 보여준다. 텍스트 형태로 표현하는 Metric, 수평 바 형태로 표현하는 Top N, 게이지 형태로 표현하는 Gauge, 마크다운 형태로 표현하는 Markdown, 테이블 형태로 표현하는 Table 이있다. Auto apply 를 활성화하면 데이터 소스가 변할 경우 바로 시각화에 반영된다. 우선 인덱스 패턴을 지정해보자. Panel options 에 인덱스 패턴을 선택할 수 있는 항목이 있다.



Panel filter 는 필터를 설정하는 것으로 쿼리바와 사용법이 같다. Cancelled : false 와 같이 작성하여 취소하지 않은 항공권에 대해서만 출력시킨다. 이제 Data 메뉴로 이동한다.



위와 같이 AvgTicketPrice 필드의 최대 최소값을 작성한다. TSVB 는 시계열 데이터를 분석하는데 특화되어 있는데, 그 중 하나인 어노테이션(Annotations)을 알아보자



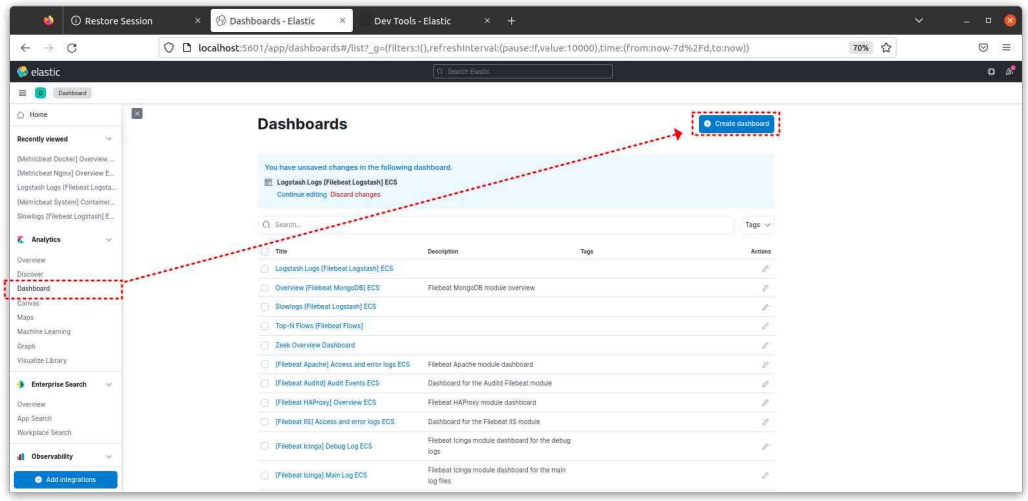
Fields 와 Row template 는 어노테이션에 마우스를 올렸을 때 보여주는 문자들이다. 필드명을 적어주고 여러개일 경우 쉼표를 사용하면된다. 이 경우 Row template 이 Fields 에 적은 필드들을 변수로 인식하고 어노테이션에 마우스를 올리면 텍스트로 보여준다.

④ 대시보드

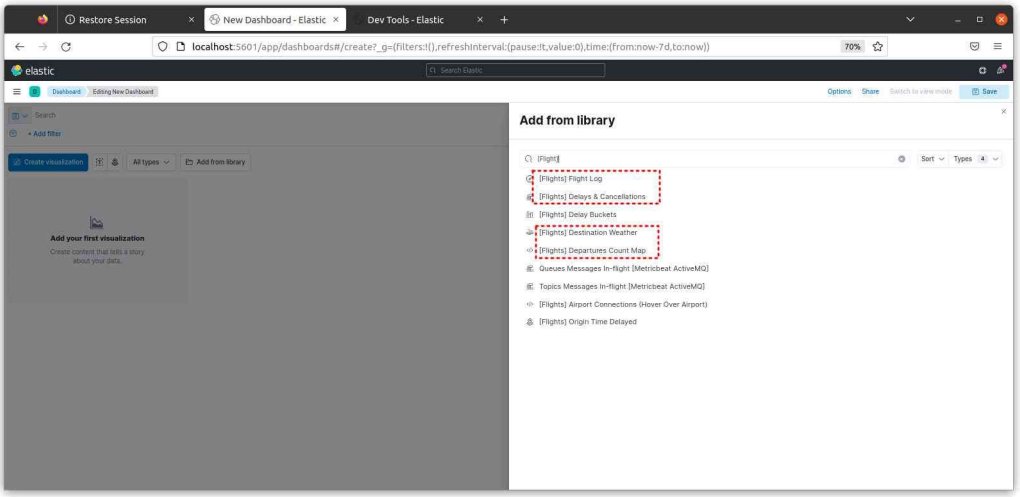
대시보드는 앞서 살펴본 시각화 타입들을 한 페이지에 모아 볼 수 있는 기능으로 한화면에서 다양한 관점으로 데이터를 보면서 분석할 수 있다.

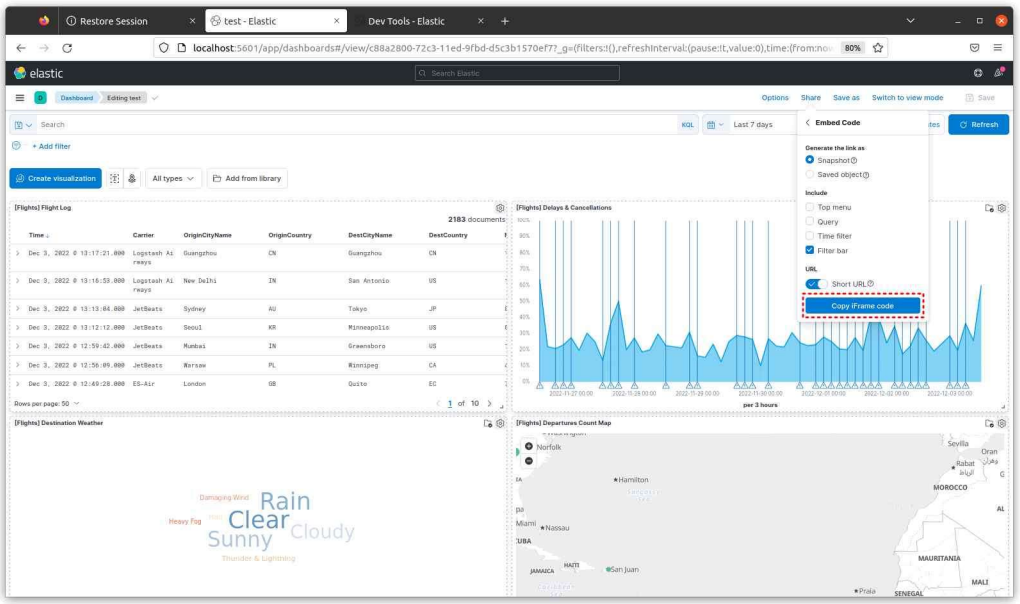
■ 대시보드 만들기

Dashboard 메뉴를 선택하면 샘플로 만들어진 대시보드를 볼 수 있다. 새로운 대시보드를 하나 만들어보자.



대시보드에서는 시각화 패널을 우리가 만들었거나 샘플에서 제공하는 타입들을 불러올 수 있다. 메뉴에서 “Add from library"를 클릭하고 ”[Flight]"를 검색하여 “Flight Log", "Delays & Cancellations", "Destination Weather", "Departures Count Map"을 선택한다.





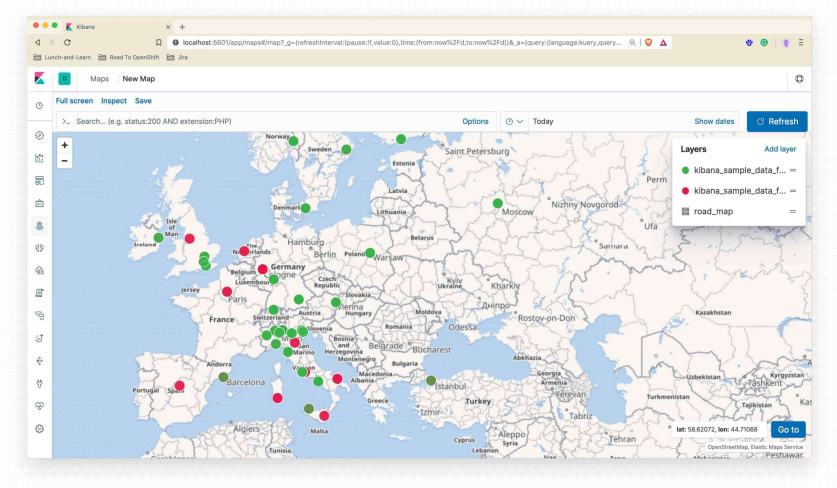
만들어진 대시보드는 저장하여 재활용하거나 편집이 가능하다. 또한 iframe 형태로 다른 웹 페이지에 임베딩 하거나 링크를 통해 웹에서 대시보드를 공유할 수도 있다

⑤ 캔버스

캔버스는 인포그래픽 형태로 데이터를 프레젠테이션 할 수 있게 해주는 툴이다. MS 사의 파워포인트처럼 사용하는 것처럼 화면을 편집하고 프레젠테이션 할 수 있다고 생각하면 된다. 키바나의 대시보드가 강력한 기능을 제공하지만 시각화 표현이 정형화되어 있다는 느낌을 받을 수 있는데, 좀 더 자신만의 방식으로 보고서나 인포그래픽 형태의 대시보드를 원하면 캔버스가 고민을 해결해 줄 것이다.

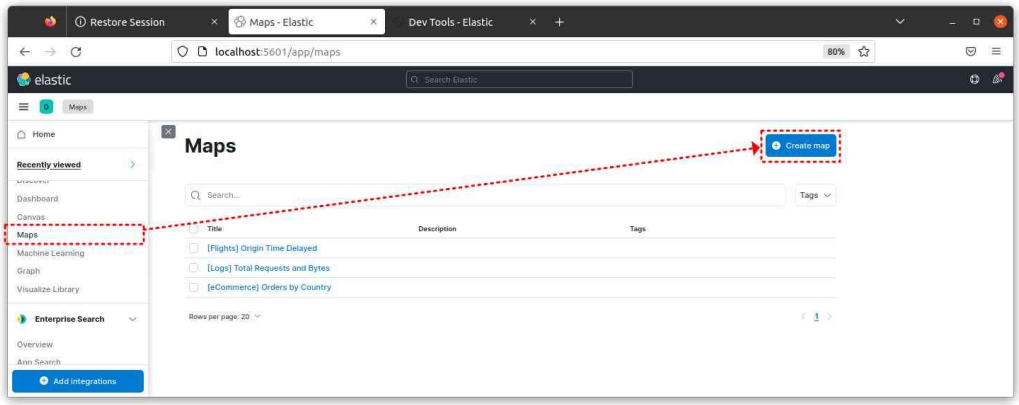
⑥ 맵스

키바나 맵스를 통해 사용자는 위치 정보가 포함된 데이터를 지도에 올려 시각화해 볼 수 있고, 멀티 레이어 기능을 통해 다양한 형태의 지도를 레이어한 화면에서 볼 수도 있다. 또 로컬이나 다른 서버에 있는 벡터 형태의 폴리곤(polygon)을 시각화할 수도 있다. 엘라스틱 스택은 위치/지역 데이터를 표현하고 처리할 수 있는 환경을 제공하는데 위치는 위경도가 포함된 특정 좌표이고, 지역은 위치가 모여서 만드는 특정 공간, 경계선이라고 생각하자.



멀티 레이어를 지원하고 선택한 항목에 대한 세부 정보도 확인할 수 있다. 또한 대시보드에 추가해 다른 시각화 객체와 함께 이용할 수 있다.

키바나의 Maps 메뉴에서 위치/지역 데이터를 시각화하는 방법을 알아본다. 맵스는 기본적으로 위치/지역 데이터가 있는 도큐먼트만 시각화가 가능하다. 키바나 왼쪽 상단의 토글 메뉴를 클릭하면 메뉴들을 확인할 수 있는데 여기에 Maps 라는 메뉴가 있다.

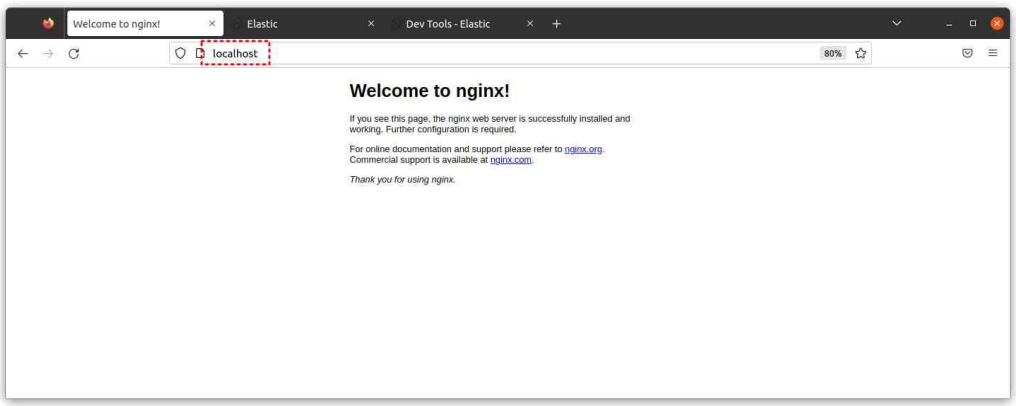


"Create map"을 클릭하고 "Add layer" 를 클릭하면 레이어를 추가할 수 있다. 레이어는 크게 벡터 레이어와 타일 서비스 두가지로 구분할 수 있다.

- 벡터레이어(점, 선, 폴리곤 등을 표현)
Upload GeoJSON, Documents, Choropleth, Clusters and grids, Heat map, Point to point, EMS Boundaries, Configured GeoJSON

- 타일 서비스
EMBS Basemap, Configured Tile Map Service, Tile Map Service, Web Map Service, Vector tiles

Configured GeoJSON 과 Configured Tile Map Service 는 키바나 설정 파일인 kibana,yml을 수정해야 보이는 메뉴이다. 자주 사용되는 레이어로 맵스를 익혀보자. 레이어를 추가하기에 앞서 사용자가 만든 GeoJSON 파일을 가져오려면 웹 서버가 필요하다. 미리 nginx 와 같은 웹서버 데몬을 설치해 두어야 한다.



■ 사용자 지정 타일맵과 GeoJSON 적용
앞서 키바나는 위치/지역 정보를 처리할 수 있다고 했는데, 위치는 단순 좌표이고 지역은 위치가 모여서 만든 폴리곤으로 키바나에서는 벡터 레이어라고 한다. 엘라스틱서치가 제공하는 벡터 레이어는 https://maps.elastic.co에서 확인할 수 있다.

전 세계 행정 구역 정보를 Vector Layers 라는 폴리곤 단위로 제공하는데, 한국에 대해서는 시도와 시군구 단위의 폴리곤으로 제공한다. 하지만 작업을 하다보면 행정동 단위나 우편번호 단위 혹은 사용자가 만든 형태 등 다양한 형태의 벡터 레이어가 필요하다. 우리는 서울시 우편번호 폴리곤이 담겨 있는 GeoJSON 파일을 벡터 레이어로 사용하는 방법을 알아보자.

```
root@elastic:/var/www/html# pwd
/var/www/html
root@elastic:/var/www/html# wget \
 https://raw.githubusercontent.com/beomtaek78/elasticstack/main/TL_KODIS_BAS_11.geojson
root@elastic:/var/www/html# ls
index.nginx-debian.html  TL_KODIS_BAS_11.geojson
root@elastic:/var/www/html#
```

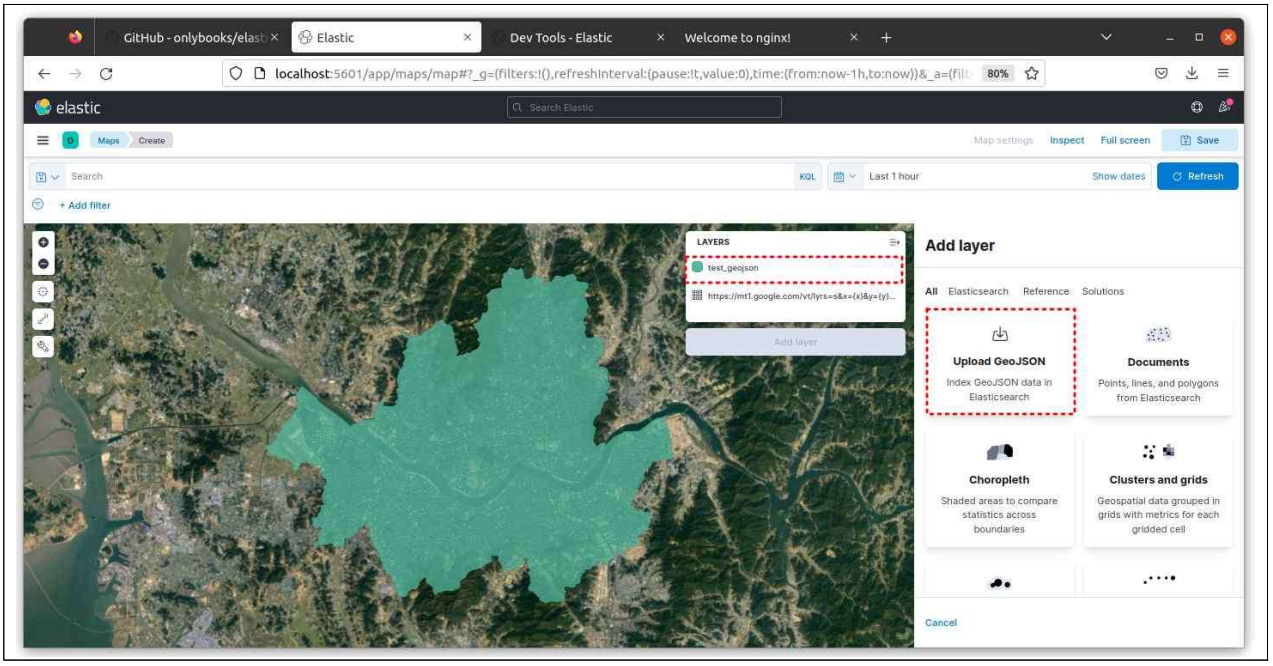
GeoJSON 파일이 엔진엑스의 루트 디렉토리에 복사되었다면 이제 키바나 설정파일을 수정해야 한다. 키바나 설정파일(kibana.yml) 에 아래내용을 가장 뒤에 추가한다.

```
root@elastic:~/kibana-7.17.7-linux-x86_64/config# pwd
/root/kibana-7.17.7-linux-x86_64/config
root@elastic:~/kibana-7.17.7-linux-x86_64/config# vi kibana.yml
```

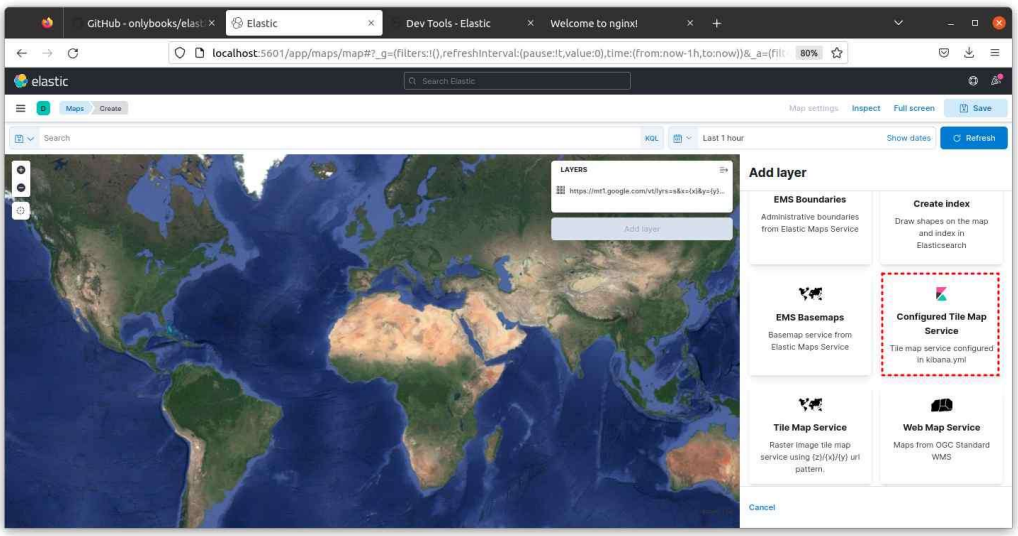
```
map.regionmap:
  layers:
    - name: "SEOUL ZIP CODE"
      url: "http://localhost/TL_KODIS_BAS_11.geojson"
      attribution: "INRAP"
      fields:
        - name: "BAS_MGT_SN"
          description: "zipcode number"
```

map.regionmap 은 사용자 벡터 레이어를 설정한다. layers 아래의 name 은 레이어이름을 정하고 URL을 작성하는데 키바나 서버와 도메인의 CORS 가 가능해야 한다. attribution 은 geoJSON 파일을 참고하는 방법을 정의하고 fields 는 geoJSON 속성중 노출할 필드를 정의한다. 이 필드는 나중에 다른 인덱스와 조인에 사용된다. BAS_MGT_SN 필드가 우편번호를 나타내는 고유한 값을 갖는다. 설정파일이 변경되었으므로 혹시 키바나가 실행중이라면 재실행, 중지 중이라면 실행시킨다.

버전 문제로 인하여 Congigured GeoJSON 이 "Add layer"에서 확인되지 않는다면 "Upload GeoJSON" 을 이용하여 파일을 직접 Upload 하는 방법도 가능하다



빠른 랜더링을 위해 사용된 타일맵은 지도를 타일 형태로 제공하는 것으로 kibana.yml 파일에 아래 내용을 추가하고 kibana를 재실행한다.

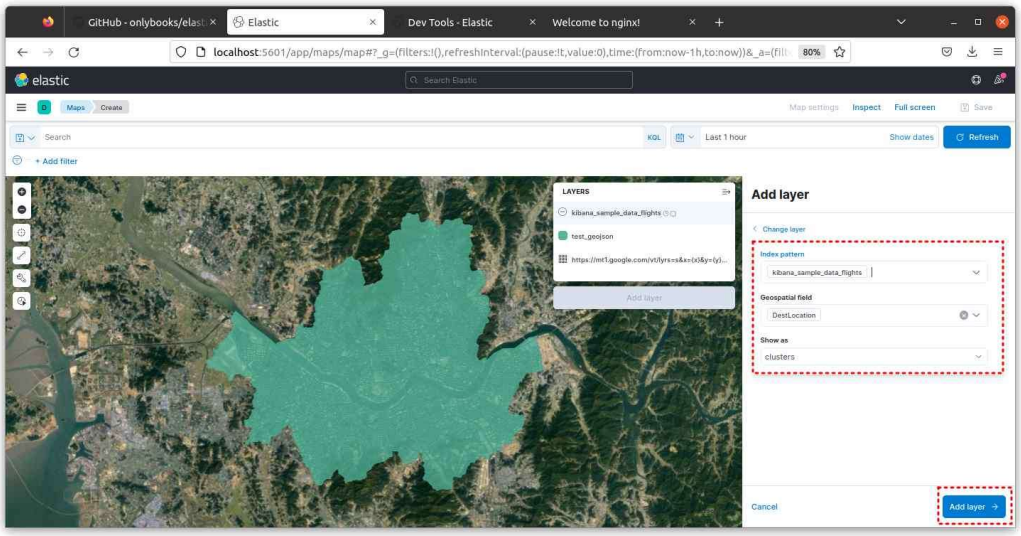



```
map.tilemap.url:
  "https://mt1.google.com/vt/lyrs=s&x={x}&y={y}&z={z}"
```

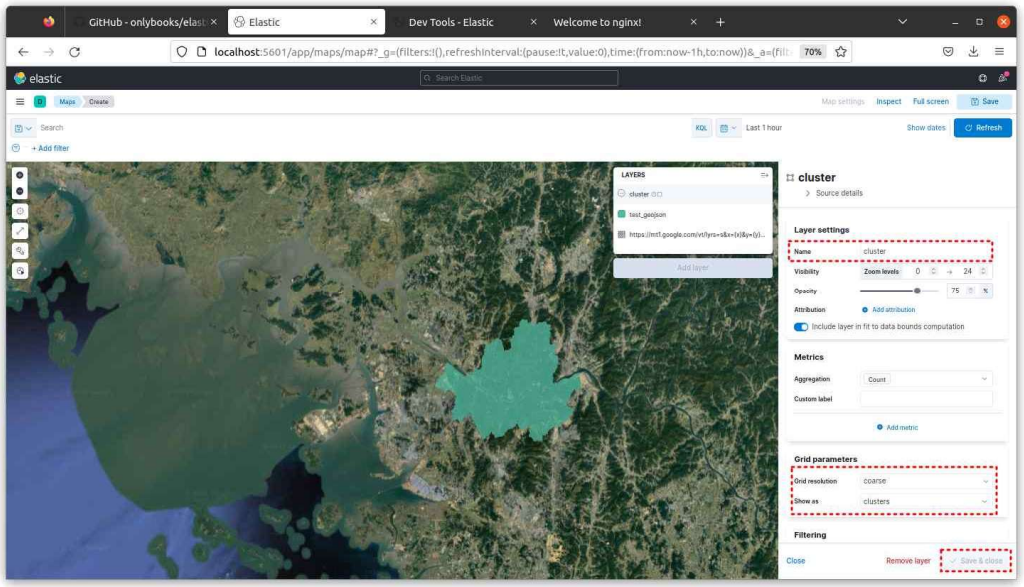
■ 클러스터와 그리드

2개의 레이어를 만들어본다. 하나는 클러스터 형태로 다른 하나는 그리드 형태로 만들 것이다. 맵스에서 Add layer를 누르고 Clusters and grids를 선택하면 아래와 같은 화면이 보인다. Index pattern 은 "kibana_sample_data_flights" , Geospatial field 는 위치 정보가 있는 필드다.

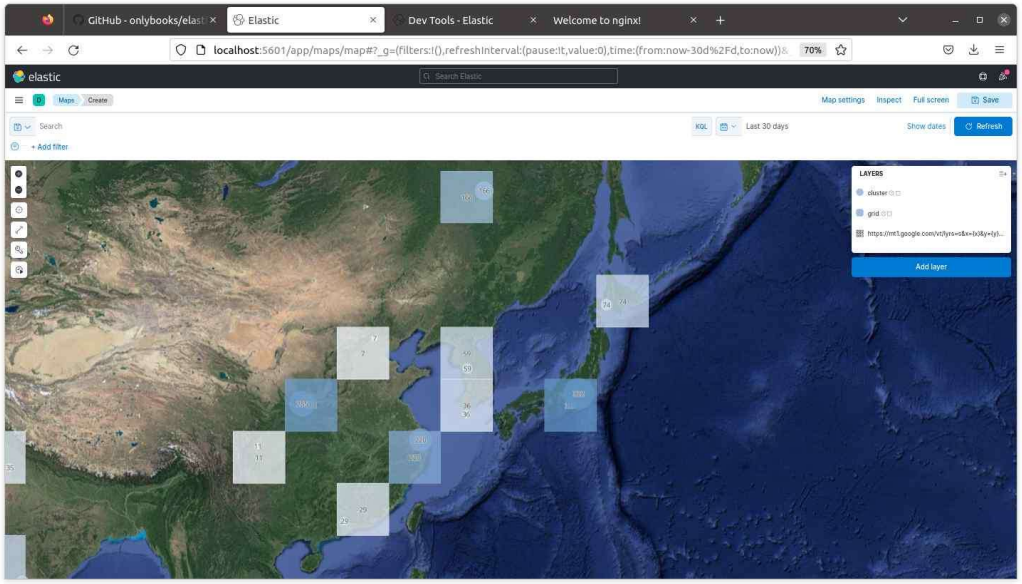
kibana_sample_data_flights 는 DestLocation 과 OrignLocation 2개의 필드를 선택할 수 있는데 DestLocation을 선택한다. Show as 는 보여주는 방식인데 clusters를 선택하고 Add layer를 클릭한다.



이후 세부설정을 할 수 있는데 레이어 Name 은 cluster 로 지정한다. Metrics에서 카운트/최소/최대/평균 같은 메트릭 집계를 구할 수 있는데 Count를 선택한다. Grid parameters 에서는 정확도를 설정할 수 있다. coarse에서 super file을 바꿔 보면서 클러스터 위치 정확도가 더 정교해 지는 모습을 확인해 보자. Save 버튼을 눌러 저장하면 cluster 레이어가 생긴다.



이제 grid 레이어를 만들어본다. 앞선 화면과 같이 Add layer에서 Cluster and grids를 선택하고 Show as 를 grids 로 지정한다. 레이어 이름은 grid 로 변경한다.



위와 같이 레이어 2개가 보인다. grid 레이어는 사각형 격자 형태로 보이는데 지오타일 그리드 집계방식으로 지오포인트를 타일 형식으로 그룹핑하여 보여주고 cluster 레이어는 원형 형태로 보인다. 지도를 확대해 보면 세부적으로 나뉘는 것을 볼 수 있다. 지도의 해상도에 따라 위치 정보가 더 정교하게 보이게 된다.