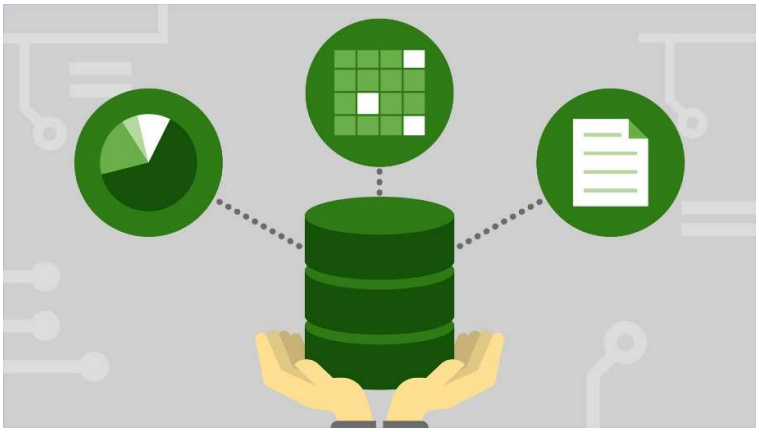
به نام خدا

دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



آزمایشگاه پایگاه‌داده

دستورکار شماره ۷

مهلت تحویل :

۱۴۰۰/۱۰/۰۱

مجتبی بنائی

دستور کار شماره ۷ - کار با الستیک سرچ

یکی از دیتابیس های معروف در حوزه جستجوی متن با سرعت بالا در ذخیره انواع داده‌های متنی و پاسخگویی به انواع کوئری های کاربر بر روی آنها، الستیک سرچ است که در اکوسیستم استارتاپی ایران هم بسیار پرطرفدار است.

در این دستور کار هم مشابه با دستورکارهای اخیر، هدف اصلی ، آشنایی اولیه با این دیتابیس و نحوه کار با آن است که حداکثر با دوساعت صرف زمان، میتوانید به راحتی آنرا انجام دهید.

گام اول :کافی است به آدرس زیر مراجعه کرده و تمامی مراحل آنرا انجام دهید :

http://yun.ir/fi9loe

تمام دستورات آنرا از ابتدا تا انتها در محیط کیبانا که محیط گرافیکی کار با الستیک‌سرچ است انجام داده، با گرفتن اسکرین شات از خروجی آنها، گزارش خود را آپلود کنید. داده‌ها را طوری وارد کنید که هر کوئری حداقل دوجواب در خروجی برگرداند.

گام دوم : با صدا زدن API زیر در یک برنامه پایتون، حداقل ۵۰۰ توئیت را در الستیک سرچ ذخیره کرده (هشتگ‌ها که همان نمادهای بورسی هستند را جداگانه در یک لیست بریزید و سپس در الاستیک سرچ ذخیره کنید) و یک داشبورد با حداقل دو ویژوالیزیشن ایجاد کنید (مثلا ابرهشتگ‌ها یا تعداد نمادهای پرتکرار)

https://www.sahamyab.com/guest/twiter/list?v=0.1

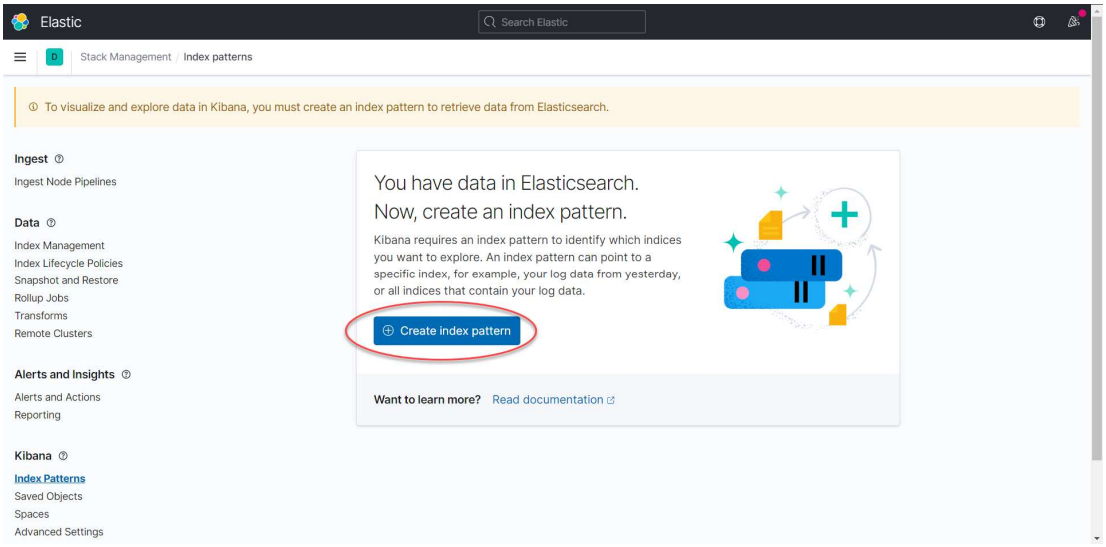
نمونه کد مورد نیاز و نحوه ایجاد یک داشبورد بر اساس داده‌های متنی می تواند به صورت زیر باشد(کتابخانه elasticsearch در پایتون را باید نصب کنید) :

```
import time
from elasticsearch import Elasticsearch
import requests
import logging
import sys
import re
url = 'https://sahamyab.com/guest/twiter/list?v=0.1'
total = 1000
api_sleep = 60

def connect_elasticsearch():
    _es = None
    _es = Elasticsearch([{'host': 'localhost', 'port': 9200}])
    if _es.ping():
```

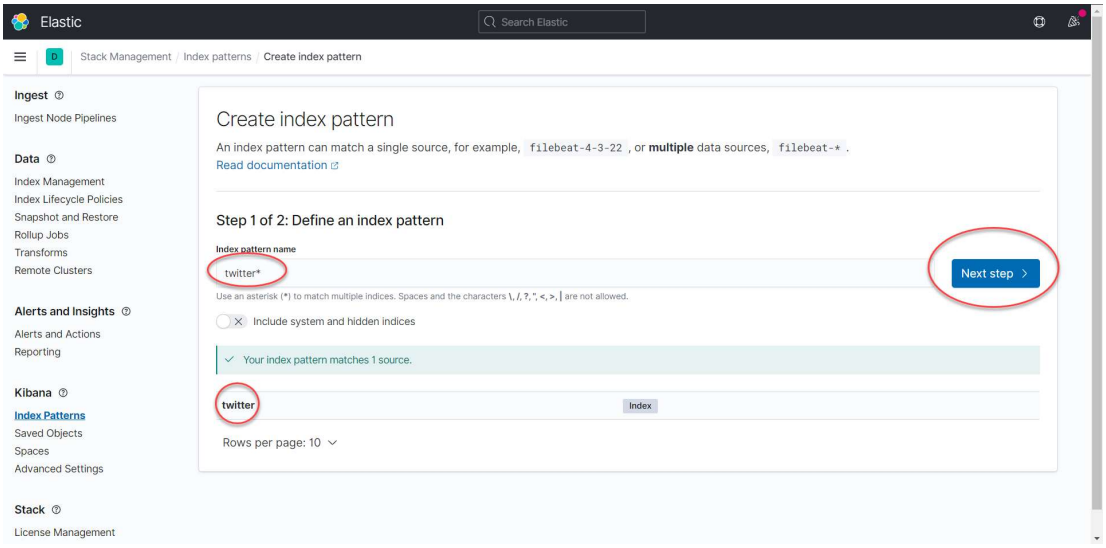
```
        print('Connected')
    else:
        print('Failed!')
    return _es
es = connect_elasticsearch()

def find_hashtags(text):
    return re.findall(r"#(\w+)", text)

def store_record(elastic_object, index_name, record):
    try:
        outcome = elastic_object.index(index=index_name, doc_type='twitter', bod
y=record)
    except Exception as ex:
        print('Error in indexing data')
        print(str(ex))

res = 0
while res < total:
    response = requests.request('GET', url, headers={'User-Agent': 'Chrome/61'})
    if response.status_code == requests.codes.ok:
        data = response.json()['items']
        for d in data:
            d['hashtags'] = find_hashtags(d['content'])
            store_record(elastic_object=es, index_name='twitter', record=d)
        res = es.search(index='twitter', body={'query':{'match_all':{}}})['hits'
]['total']['value']
    else:
        print("Response code error: " + str(response.status_code))
    print('Fetched and inserted {} tweets so far'.format(res))
    print('Waiting for {} seconds...'.format(api_sleep))
    time.sleep(api_sleep)
```

نحوه ایجاد داشبورد هم به صورت زیر است :

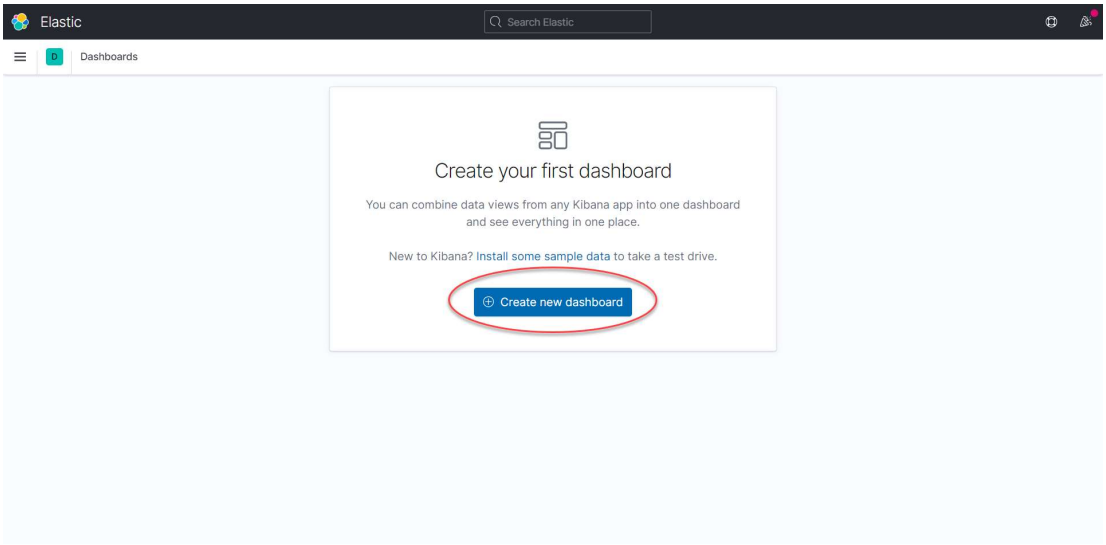
ابتدا یک index pattern می سازیم.



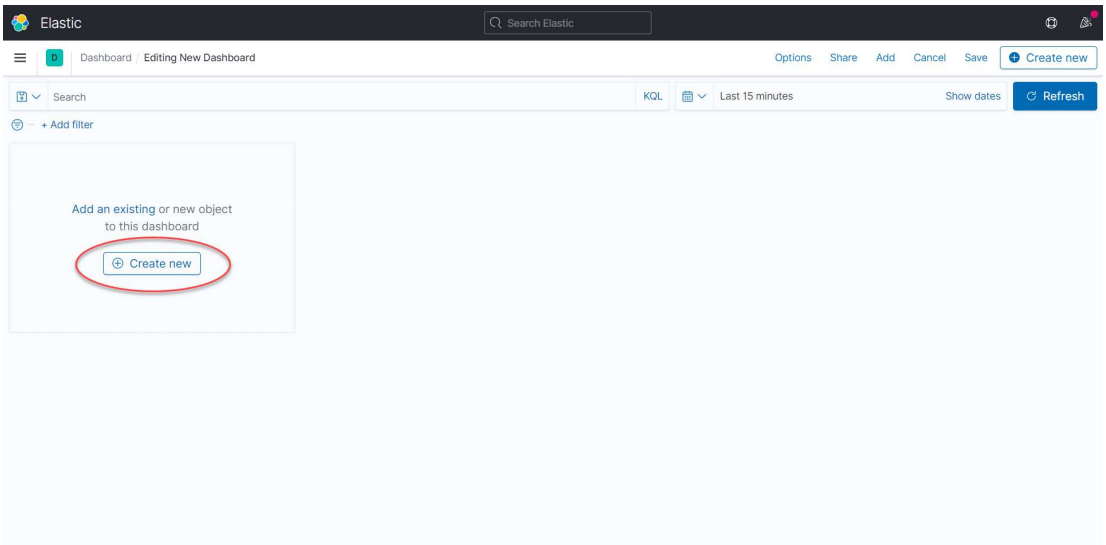
ایندکس با نام twitter را انتخاب کرده و next را می زنیم.



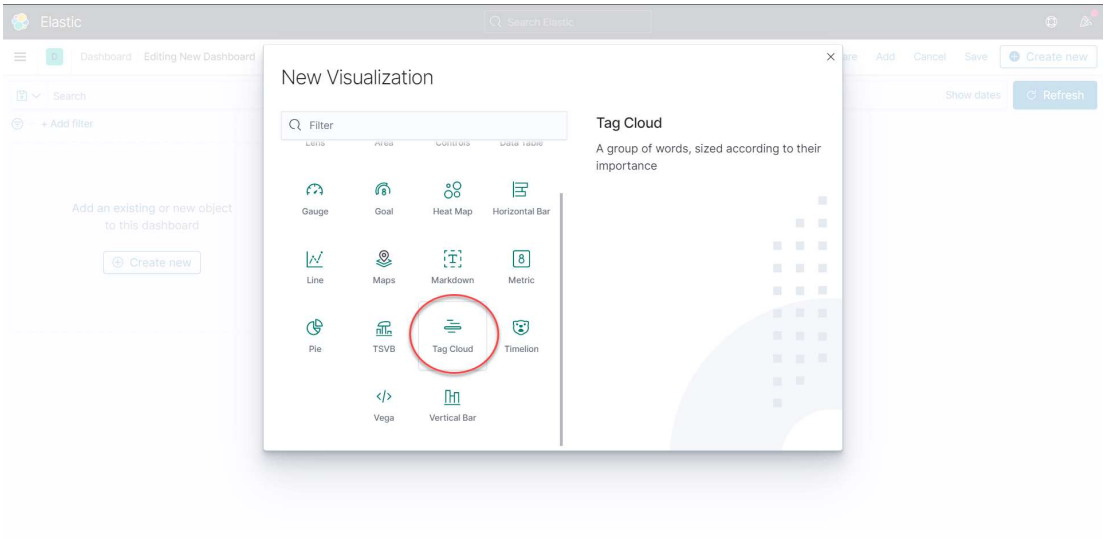
سپس یک داشبورد می سازیم.



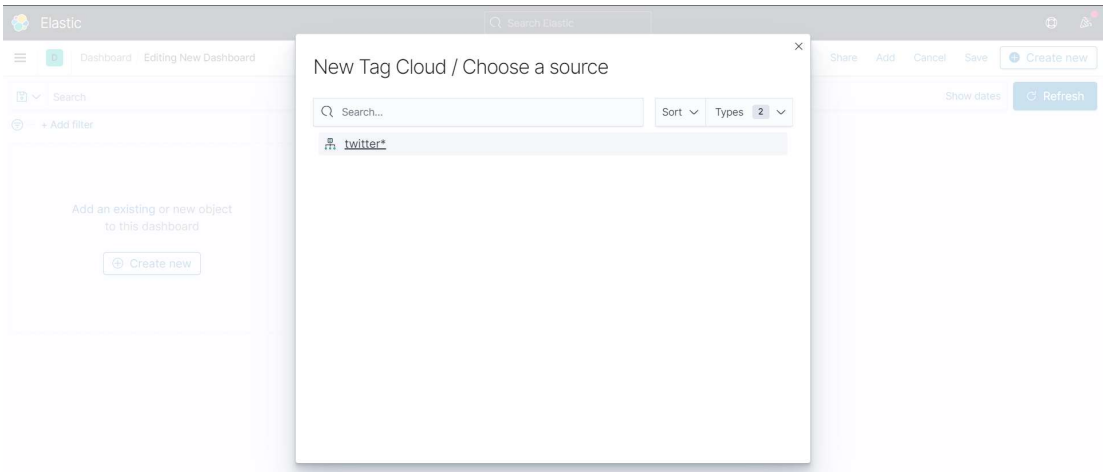
سپس create new را می زنیم تا twitter را اضافه کنیم.



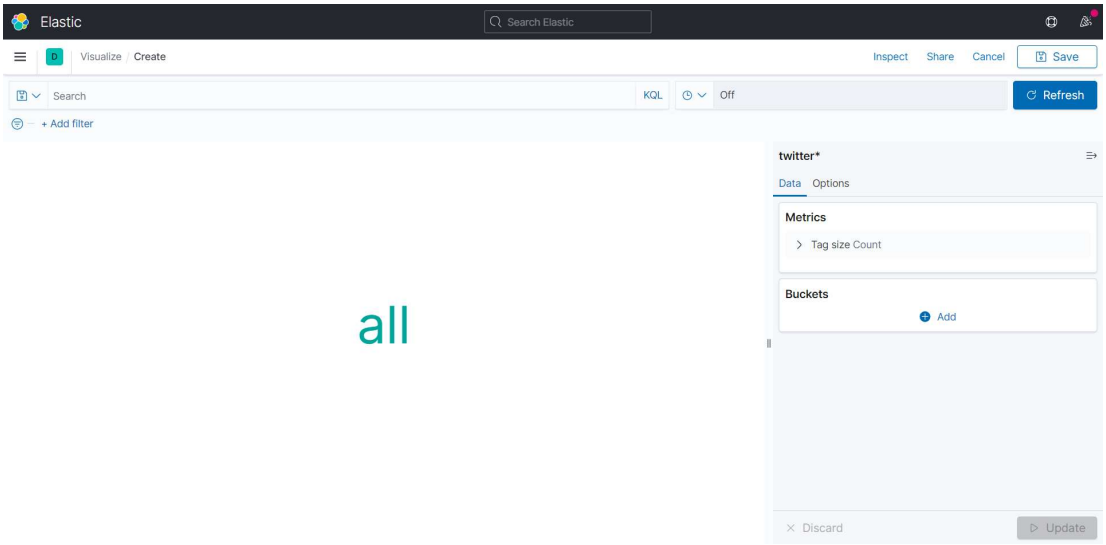
برای visualization اول tag cloud را انتخاب می کنیم که بسته به تعداد تکرار کلمه، سایز آن ست می کند.



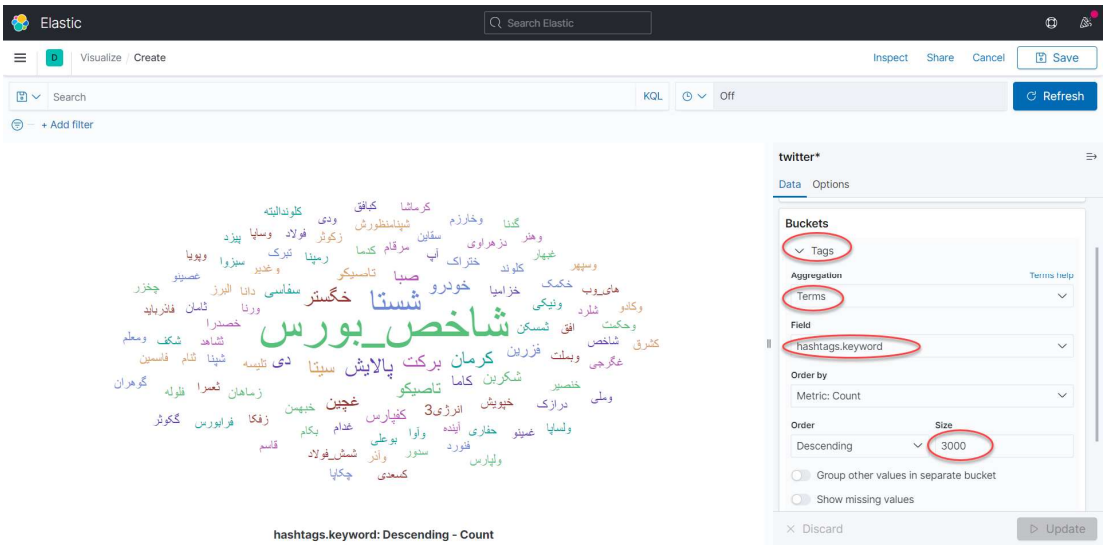
سپس twitter را انتخاب می کنیم.



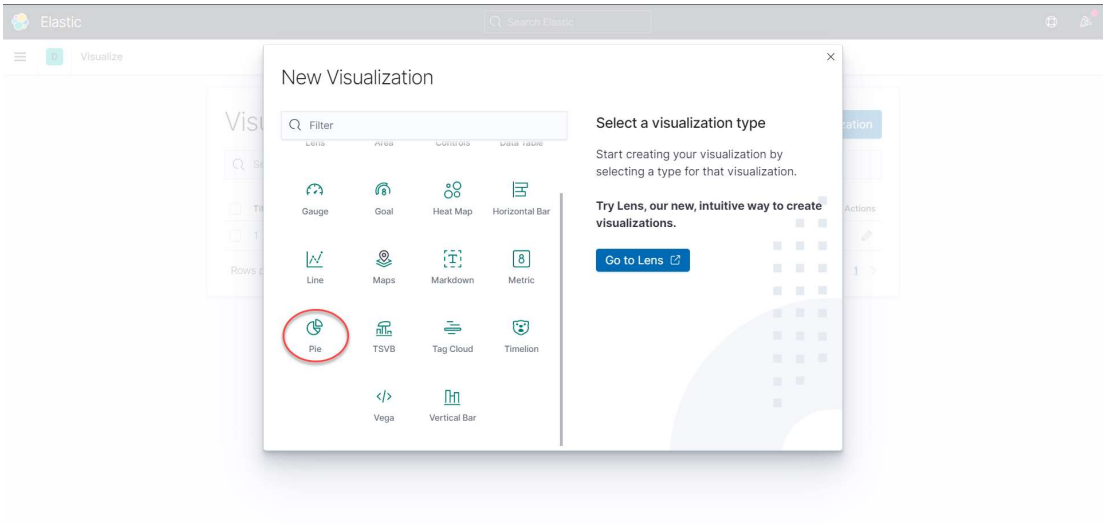
در ابتدا تنها all نشان داده می شود.



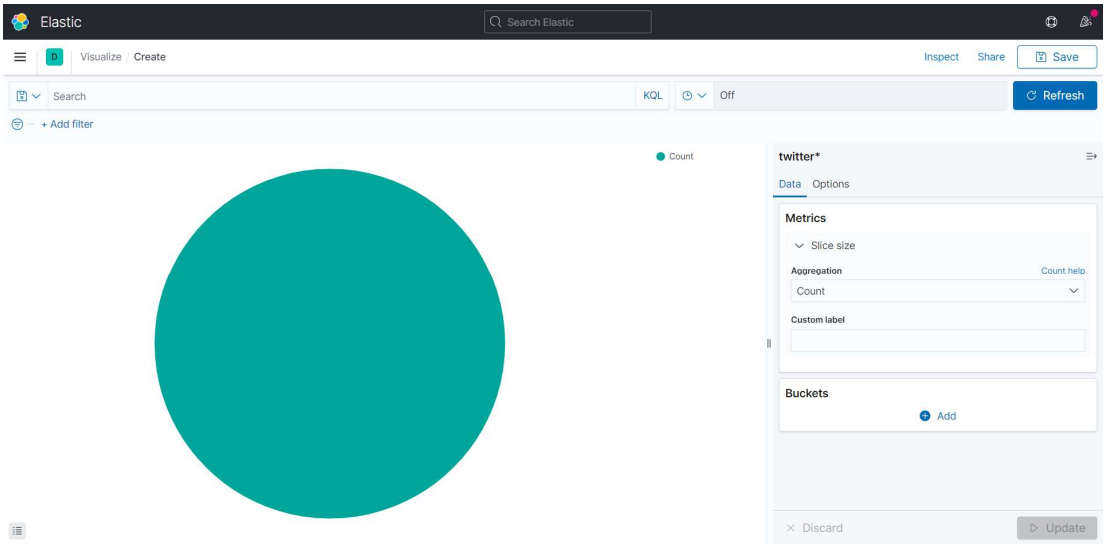
تنظیمات رو در سمت راست تغییر می دهیم تا برای hashtag ها این visualization انجام شود که خروجی به صورت زیر می شود.



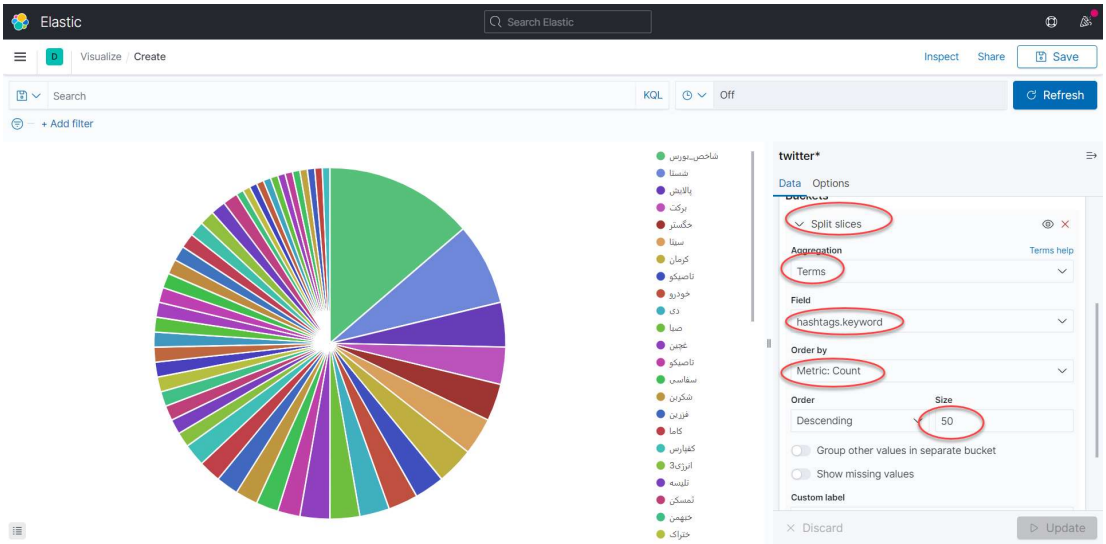
برای visualization دوم، Pie را انتخاب می کنیم که سایز slice دایره بر اساس تعداد تکرار کلمه می شود.



خروجی اولیه به این صورت می شود.



در تنظیمات bucket مانند قبل hashtags رو انتخاب می کنیم و خروجی نهایی به این صورت می شود.


Elastic
Search Elastic
Visualize / Create
Inspect
Share
Save
Search
KQL
Off
Refresh
+ Add filter
twitter*
Data
Options
Split slices
Aggregation
Terms help
Terms
Field
hashtags.keyword
Order by
Metric: Count
Order
Size
Descending
50
Group other values in separate bucket
Show missing values
Custom label
Discard
Update