Project 4: Elasticsearch

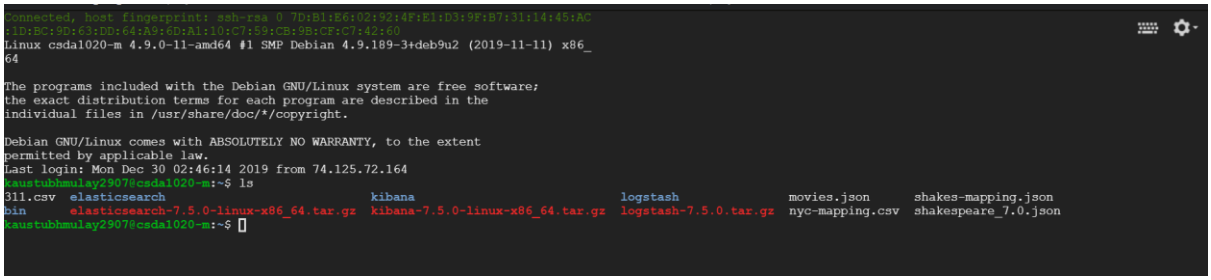
Participants:
1. Deenu Yadav
2. Fanny Guevara
3. Kaustubh Mulay

Dataset used: https://nycopendata.socrata.com/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9

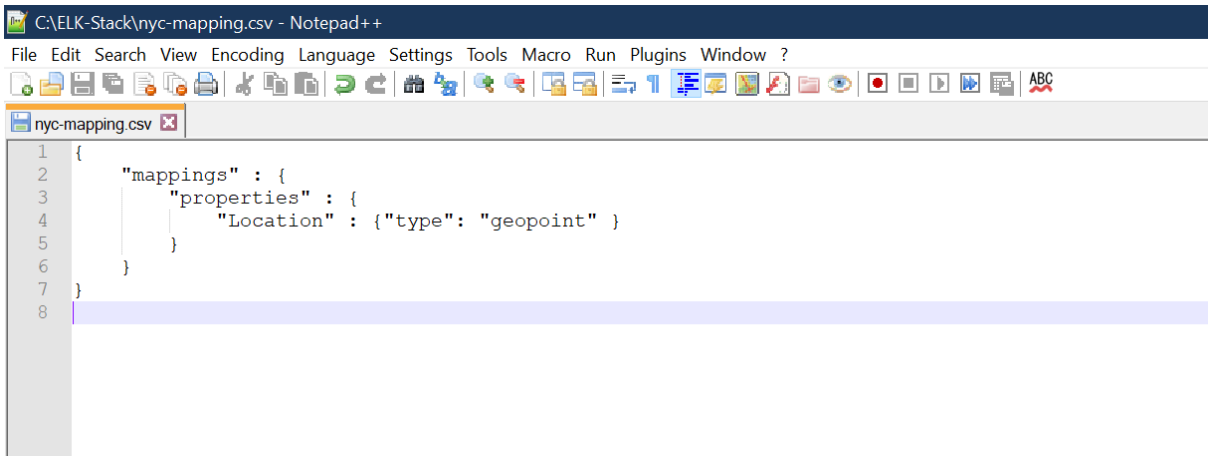
Number of rows: 22.1 million

Initial steps for setup:

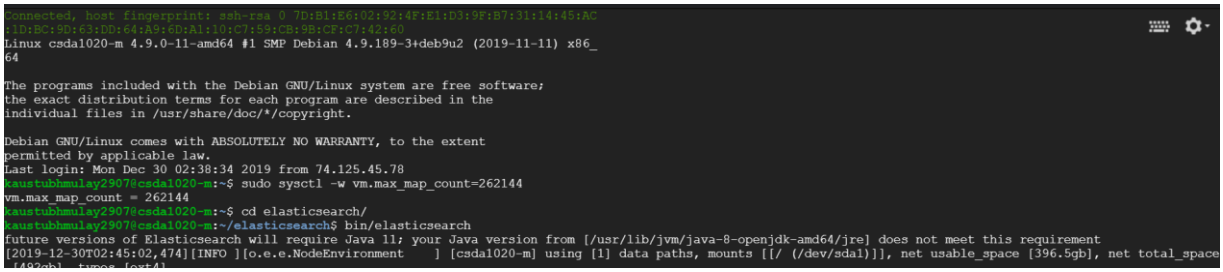
1. Upload dataset to VM instance (assuming Elasticsearch, Kibana and Logstash are all installed, the configuration files correctly modified as in Project 3). Checking contents of upload location:



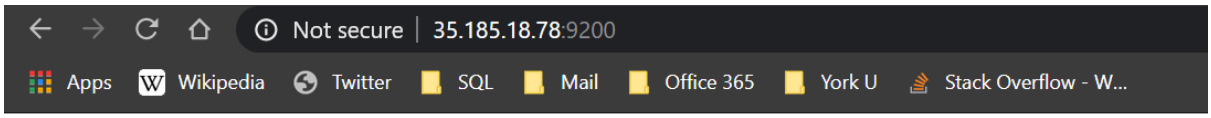
2. Create mapping for coordinate map question.



3. Run commands for Elasticsearch after setting a higher virtual memory for the purpose:

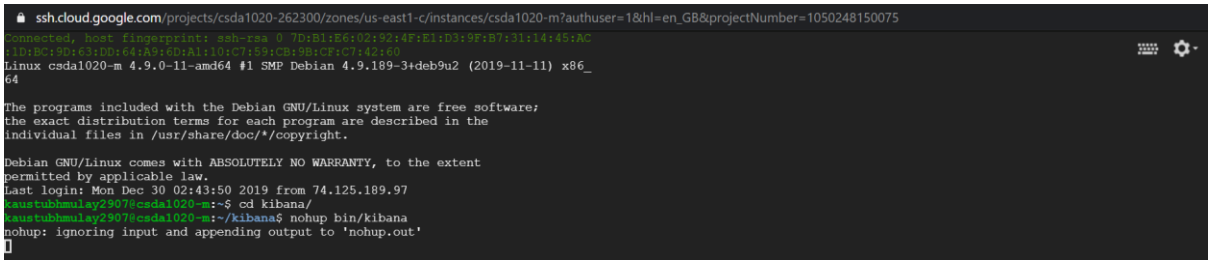


4. Check if Elasticsearch is running or not:

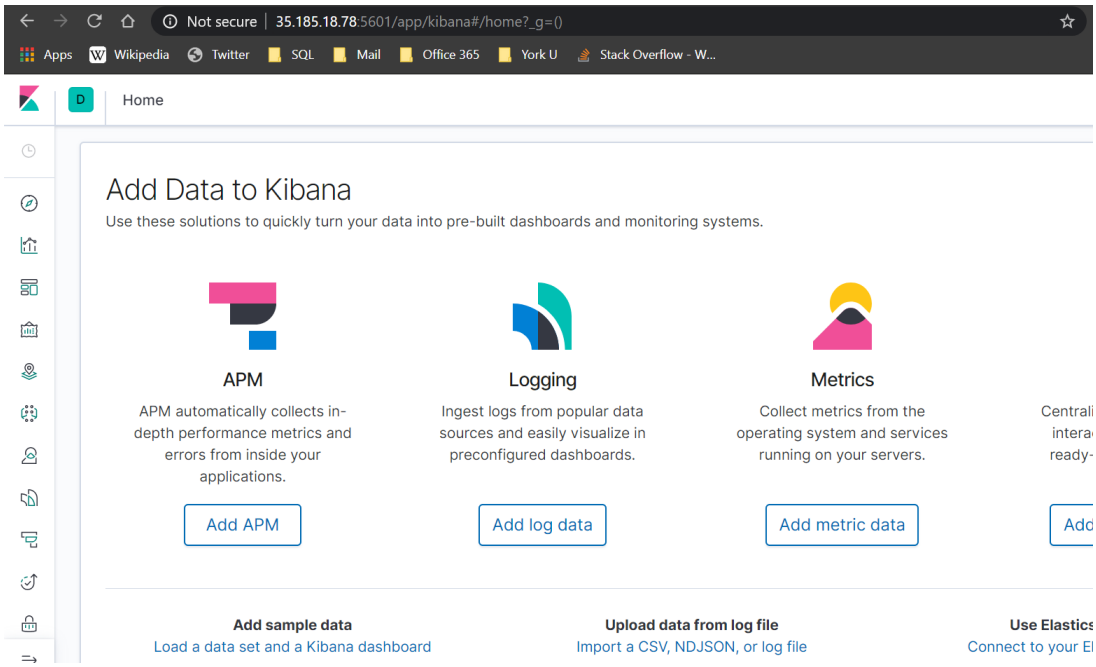



```
{
  "name" : "csda1020-m",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "ibe1yO4CRVuqj0G6L0xCCA",
  "version" : {
    "number" : "7.5.0",
    "build_flavor" : "default",
    "build_type" : "tar",
    "build_hash" : "e9ccaed468e2fac2275a3761849cbee64b39519f",
    "build_date" : "2019-11-26T01:06:52.518245Z",
    "build_snapshot" : false,
    "lucene_version" : "8.3.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

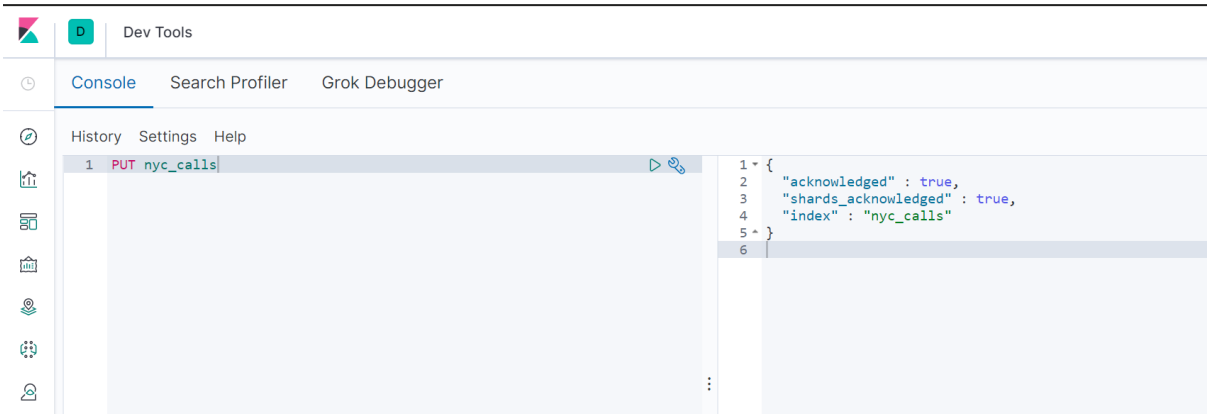

5. After opening another VM instance, running command to run Kibana in the background:



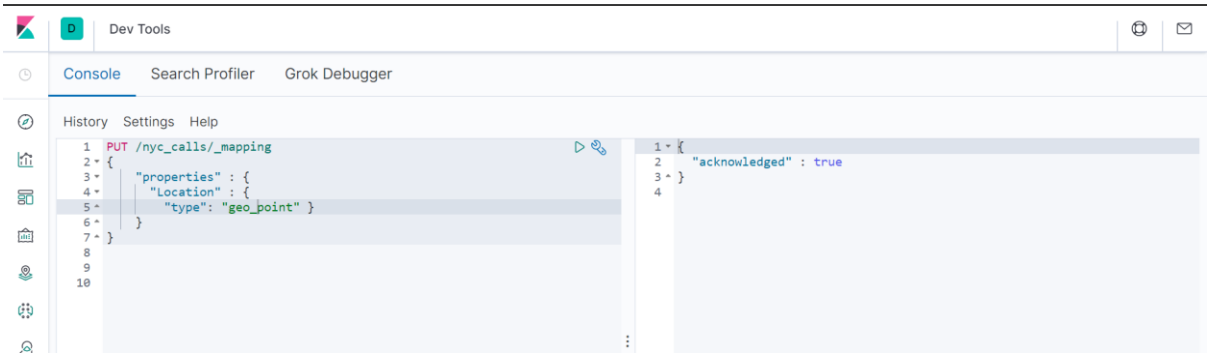
6. Checking if Kibana is successfully running or not:



7. Creating index in Kibana:



8. Changing the mapping of the index in Kibana for the location columns:



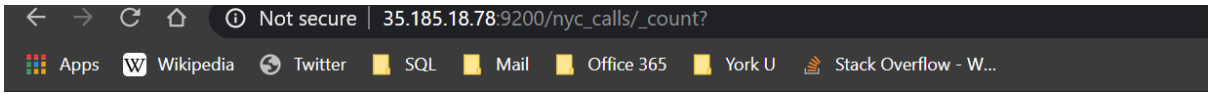
9. Checking contents of logstash configuration file to ensure that the index name is correctly reflected:

```
ssh.cloud.google.com/projects/csda1020-262300/zones/us-east1-c/instances/csda1020-m?authuser=1&hl=en_GB&projectNumber=1050248150075
input {
   file {
      path => "/home/kaustubhmulay2907/311.csv"
      start_position => "beginning"
     sincedb_path => "/dev/null"
     }
  }

filter {
        csv {
               separator =>","
               columns => ["Unique Key","Created Date","Closed Date","Agency","Agency Name","Complaint Type","Descriptor","Location Type","Incident Zip","Incident
Address","Street Name","Cross Street 1","Cross Street 2","Intersection Street 1","Intersection Street 2","Address Type","City","Landmark","Facility Type","Status","
Due Date","Resolution Description","Resolution Action Updated Date","Community Board","BBL","Borough","X Coordinate (State Plane)","Y Coordinate (State Plane)","Ope
n Data Channel Type","Park Facility Name","Park Borough","Vehicle Type","Taxi Company Borough","Taxi Pick Up Location","Bridge Highway Name","Bridge Highway Directi
on","Road Ramp","Bridge Highway Segment","Latitude","Longitude","Location"]
               }
     mutate {convert => {"X Coordinate (State Plane)","integer"}}
     mutate {convert => {"Y Coordinate (State Plane)","integer"}}
     mutate {convert => {"Latitude","float"}}
     mutate {convert => {"Longitude","float"}}
     mutate {replace => { "Location" => "%{Latitude},%{Longitude}" }}
     }

output {
 elasticsearch {
 hosts => "10.142.0.2:9200"
 index => "nyc_calls"
  }
stdout {}
}
~
~
~
~
~
~
~
~
~
~
"logstash_NYC311.config" [dos] 27L, 1374C                                                    18,64          All
```

10. Checking the contents of the Logstash file in VM instance and then running command to run Logstash in background:

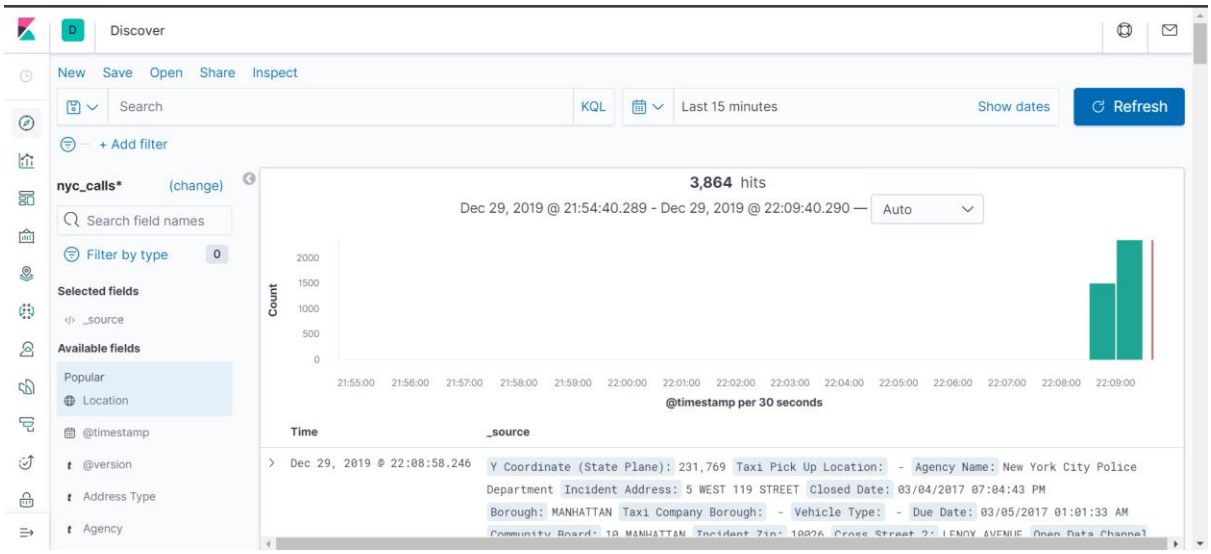
```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 30 02:46:14 2019 from 74.125.72.164
kaustubhmulay2907@csda1020-m:~$ ls
311.csv  elasticsearch                                  kibana                                  logstash                movies.json      shakes-mapping.json
bin      elasticsearch-7.5.0-linux-x86_64.tar.gz  kibana-7.5.0-linux-x86_64.tar.gz  logstash-7.5.0.tar.gz   nyc-mapping.csv  shakespeare_7.0.json
kaustubhmulay2907@csda1020-m:~$ cd logstash/
kaustubhmulay2907@csda1020-m:~/logstash$ nohup bin/logstash -f /home/kaustubhmulay2907/logstash/logstash_NYC311.config
nohup: ignoring input and appending output to 'nohup.out'
^Ckaustubhmulay2907@csda1020-m:~/logstash$ ls
bin     CONTRIBUTORS  Gemfile       kibana.yml  LICENSE.txt  logstash-core            logstash_NYC311.config  nohup.out   tools   x-pack
config  data          Gemfile.lock  lib         logs         logstash-core-plugin-api  modules                 NOTICE.TXT  vendor
kaustubhmulay2907@csda1020-m:~/logstash$ vi  logstash_NYC311.config
kaustubhmulay2907@csda1020-m:~/logstash$ nohup bin/logstash -f /home/kaustubhmulay2907/logstash/logstash_NYC311.config
nohup: ignoring input and appending output to 'nohup.out'
```

11. Checking if the data is successfully getting uploaded or not:



{"count":3191,"_shards":{"total":1,"successful":1,"skipped":0,"failed":0}}

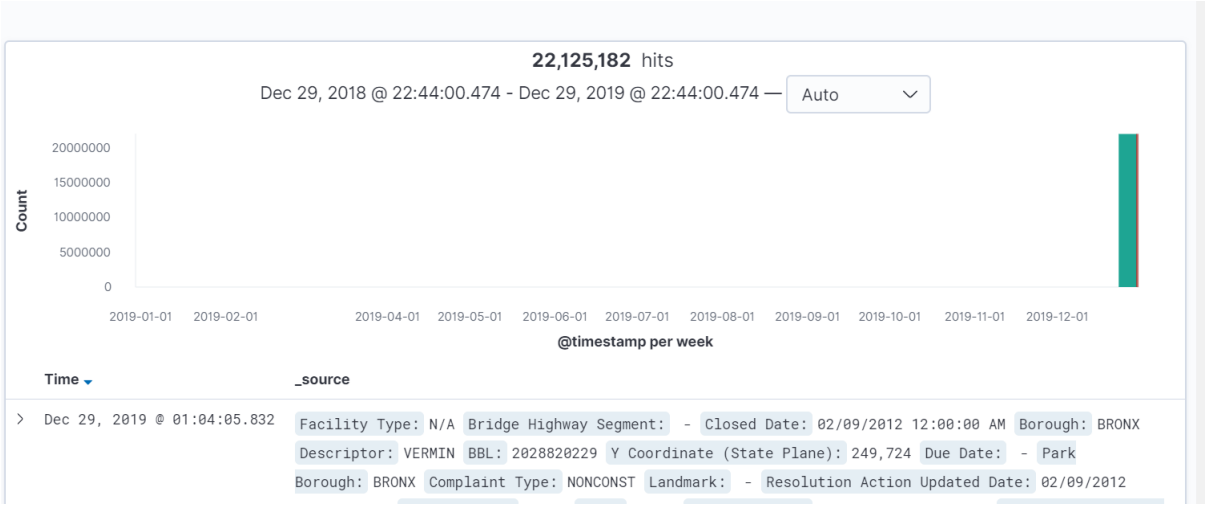
12. Checking if the documents are being created in Kibana or not:



13. Checking the progress of the data import into Kibana to see if the entire dataset was loaded or not. Full data successfully loaded:


```
{
  "count" : 22125182,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  }
}
```

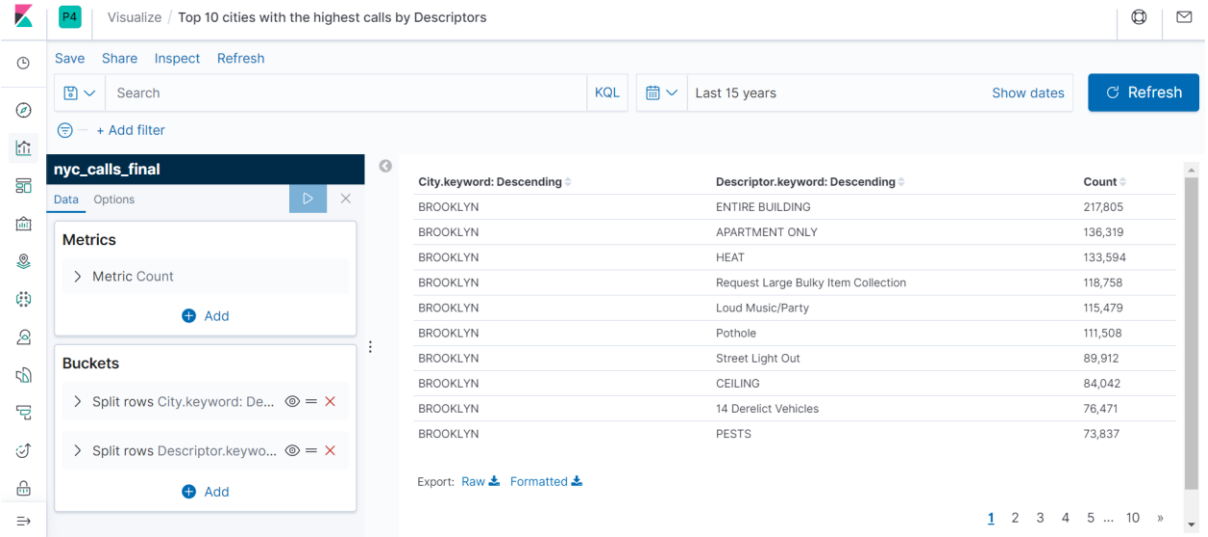
14. Checking if all the documents are visible in Kibana or not:

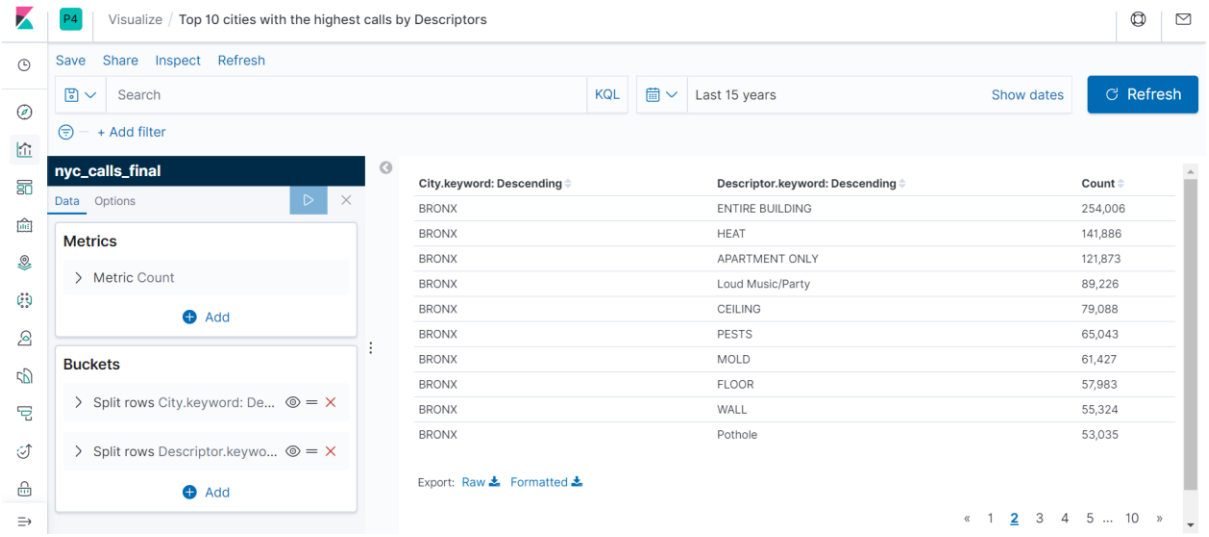


Analytical Questions:

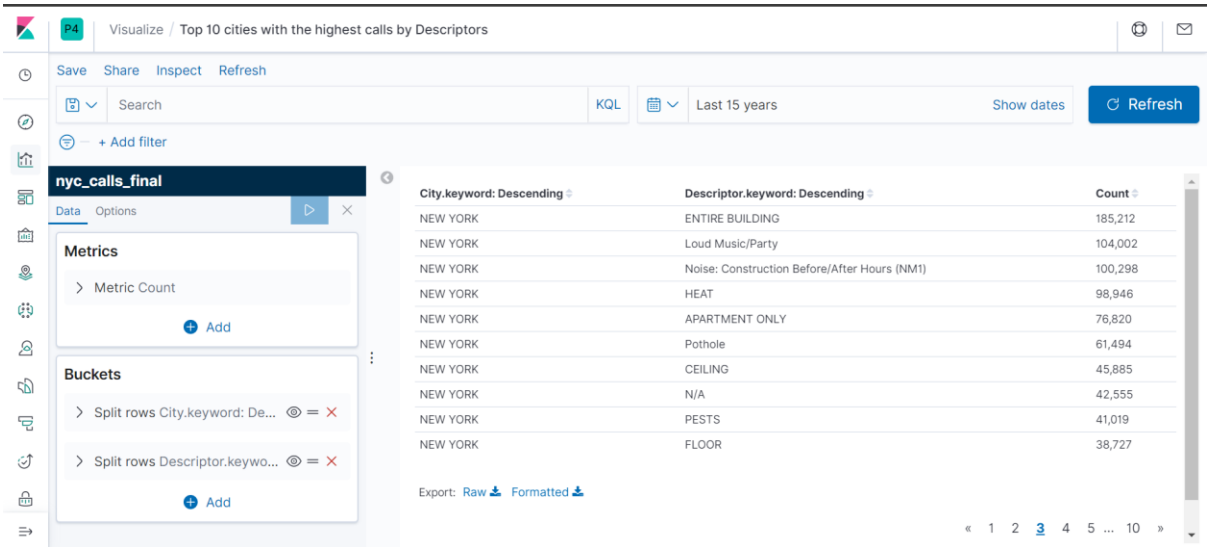
1. Create a table showing the top 10 cities with the highest calls alongside the count of top 10 complaint calls (by Descriptor) in each city.

Showing first few pages:



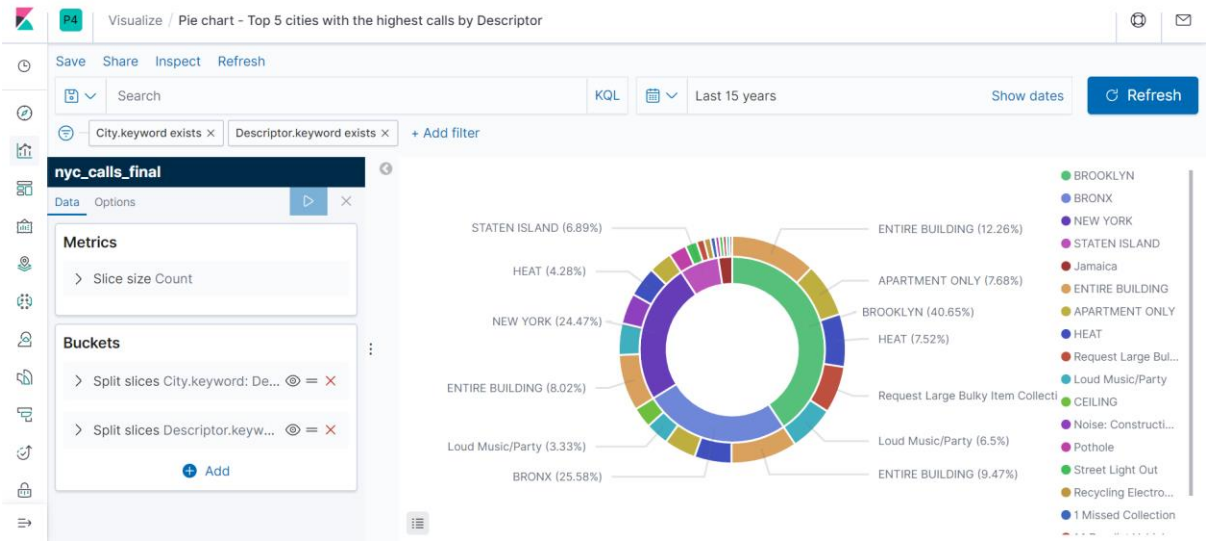


City.keyword: Descending	Descriptor.keyword: Descending	Count
BRONX	ENTIRE BUILDING	254,006
BRONX	HEAT	141,886
BRONX	APARTMENT ONLY	121,873
BRONX	Loud Music/Party	89,226
BRONX	CEILING	79,088
BRONX	PESTS	65,043
BRONX	MOLD	61,427
BRONX	FLOOR	57,983
BRONX	WALL	55,324
BRONX	Pothole	53,035

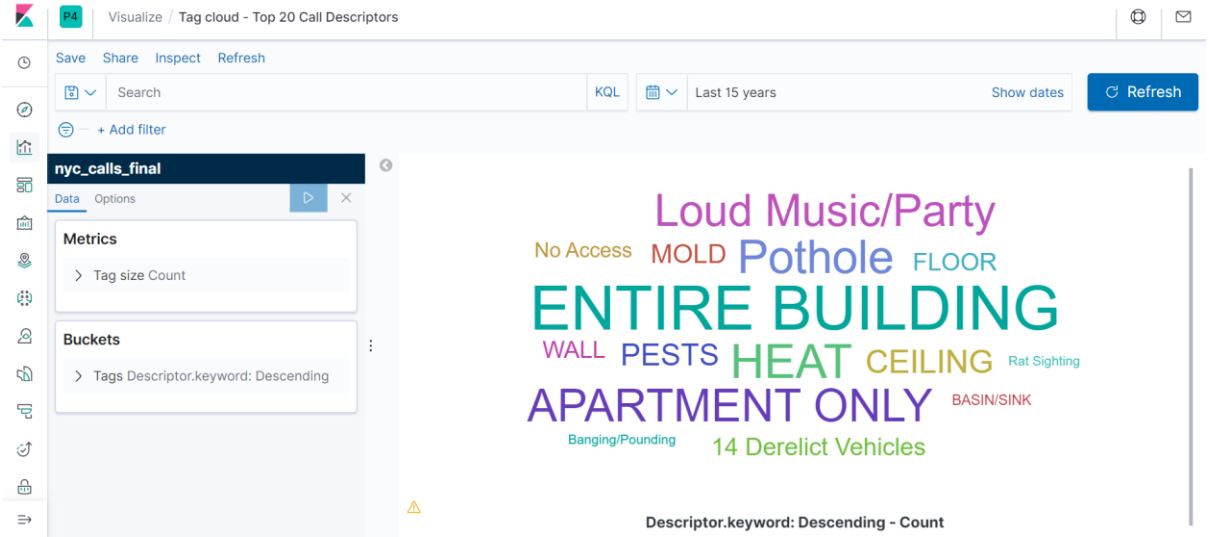


City.keyword: Descending	Descriptor.keyword: Descending	Count
NEW YORK	ENTIRE BUILDING	185,212
NEW YORK	Loud Music/Party	104,002
NEW YORK	Noise: Construction Before/After Hours (NM1)	100,298
NEW YORK	HEAT	98,946
NEW YORK	APARTMENT ONLY	76,820
NEW YORK	Pothole	61,494
NEW YORK	CEILING	45,885
NEW YORK	N/A	42,555
NEW YORK	PESTS	41,019
NEW YORK	FLOOR	38,727

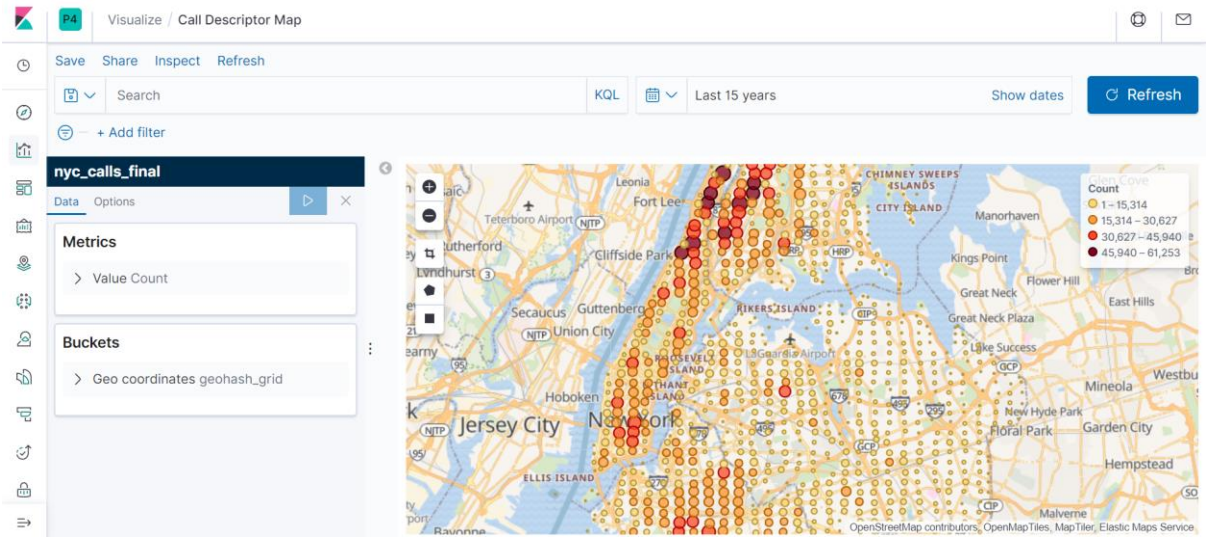
2. Create a pie chart showing the top 5 cities with the highest calls alongside the top five calls (Descriptor) in each city.



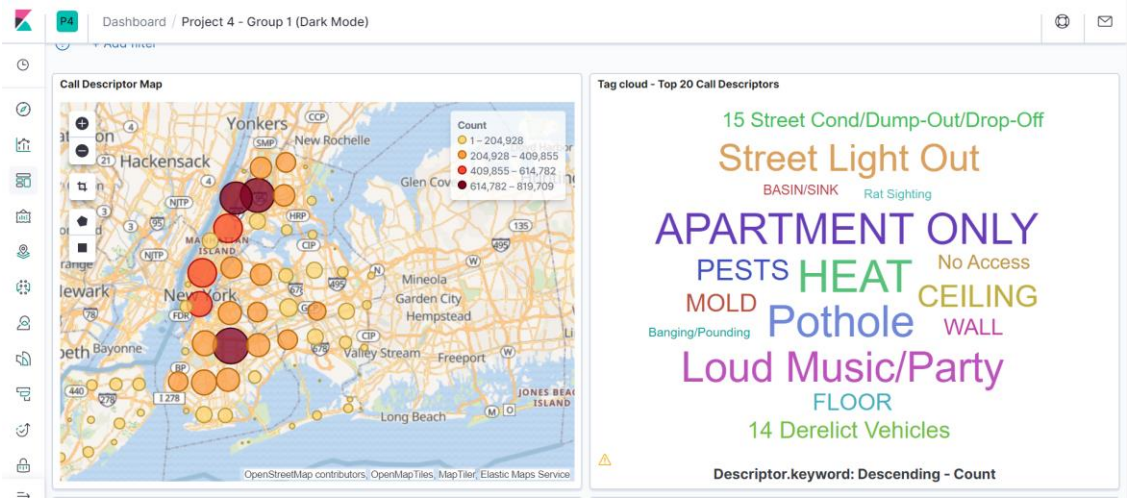
3. Create a tag cloud representing the top 20 call descriptors.

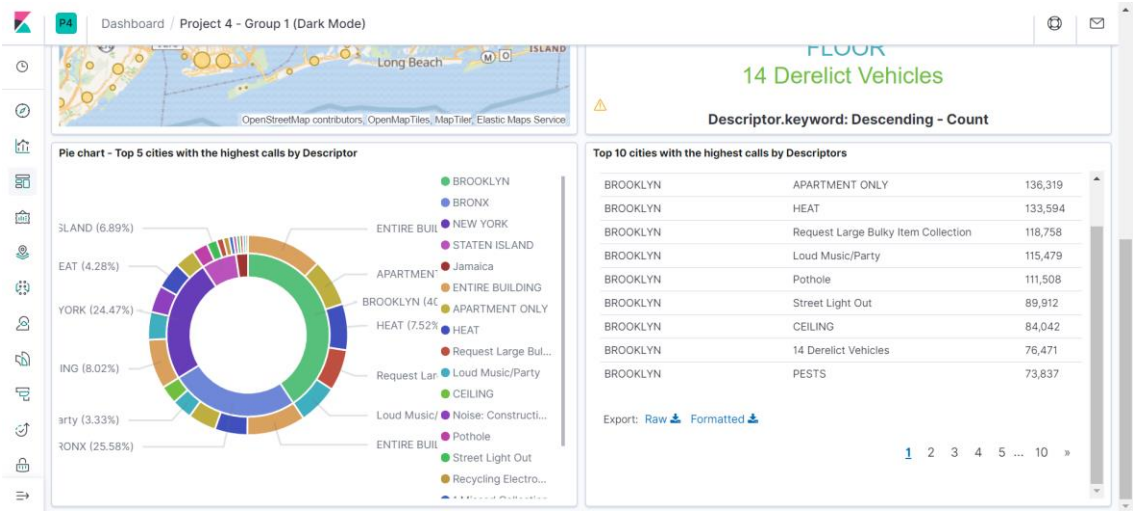


4. Create a coordinated map of all the major call descriptors in each city.



5. Create a dashboard for all visualizations of 1to 4 above.





Thus, Project 4 is completed successfully.

---End---