



INSIGHTS AND  
ANALYSIS USING  
ELASTICSEARCH,  
LOGSTASH AND  
KIBANA (ELK) STACK

BY:  
Joseph Gyamfi

CSDA1020 – Big Data Analytics Tools

Project 4

## **Background/Introduction**

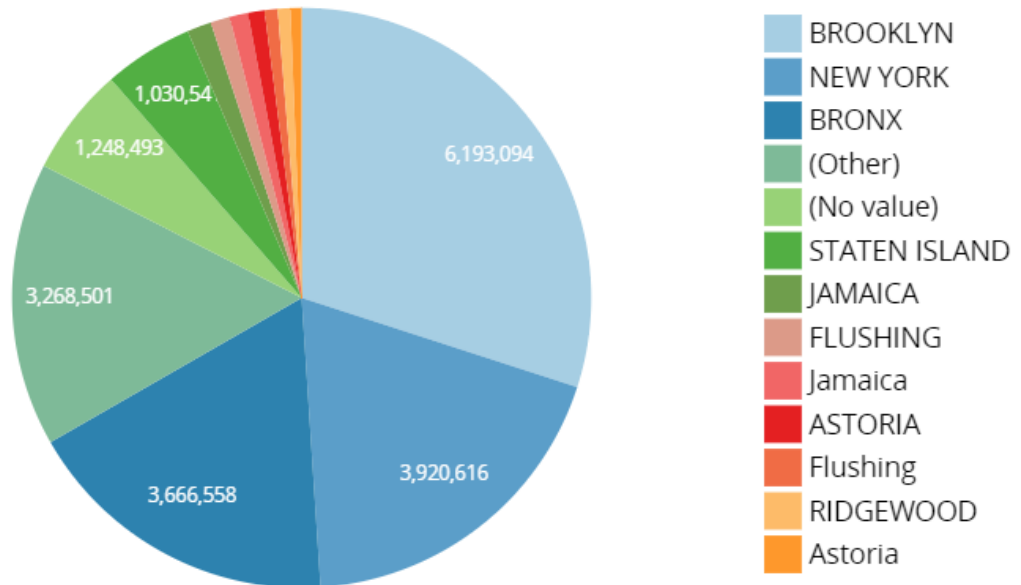
There are amazing benefits to real-time big data analytics. Real-time analysis allows organizations to develop more effective strategies in less time, offering deep insight into behavioural trends with which to improve performance. For the city of New York, the 311 number offers a valuable service to it's citizens by allowing agencies to manage workloads efficiently and improve City government through accurate, consistent measurement and analysis of service delivery. In order to aid New York in its mission, a powerful big data analytics tool, the ELK (Elasticsearch, Logstash, and Kibana) stack, was leveraged to provide valuable insights drawn from their service requests.

## **Data Description & Exploratory Analysis**

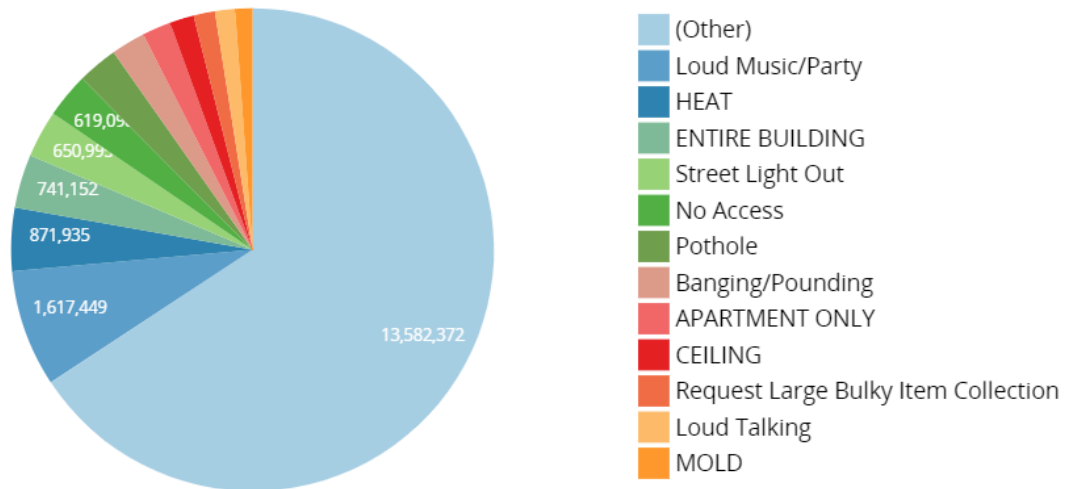
The data is drawn from NYC Open Data at: <https://nycopendata.socrata.com/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9/data>. This dataset includes all 311 service requests from 2010 to the present and is automatically updated daily containing approximately 20.7 million rows and 41 columns. A sample of the data types as well as some preliminary explorations are below:

Column Name	Description	Type	
Unique Key	Unique identifier of a Service Request (SR) in the open data...	Plain Text	T
Created Date	Date SR was created	Date & Time	📅
Closed Date	Date SR was closed by responding agency	Date & Time	📅
Agency	Acronym of responding City Government Agency	Plain Text	T
Agency Name	Full Agency name of responding City Government Agency	Plain Text	T
Complaint Type	This is the first level of a hierarchy identifying the topic of t...	Plain Text	T
Descriptor	This is associated to the Complaint Type, and provides furt...	Plain Text	T

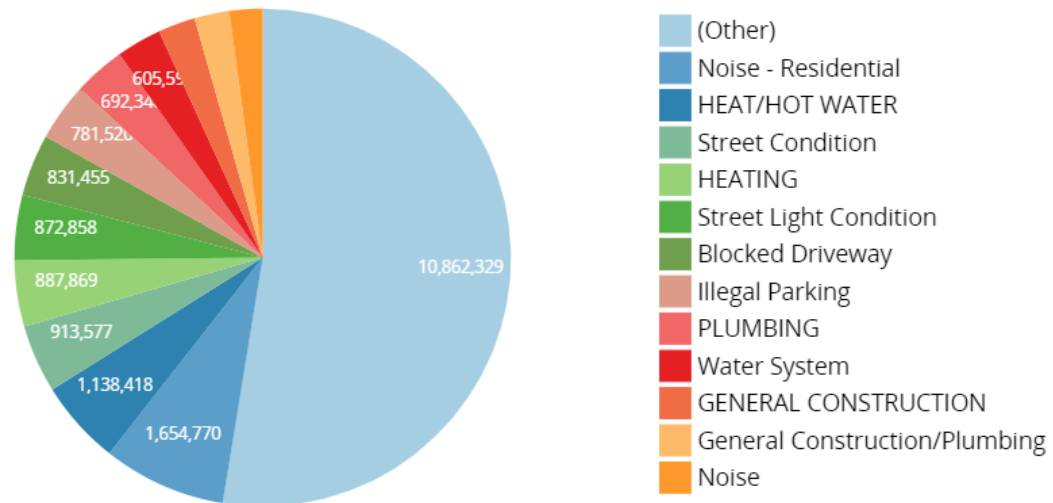
## Cities



## Descriptors



## Type of Complaint



## Methodology & Analysis

The Project was completed by utilizing an ELK stack through the creation of the Logstash configuration, geo-point template, firing Logstash to ingest the NYC 311 service request data into Elasticsearch and using Kibana to analyze and visualize the results as per the questions given.

The goal of this project is to answer the following analytical questions:

1. What are the top 10 cities with the highest calls vs. the top 10 complaints (by Descriptor) in each city? (Visualized with a table)
2. What are the top 5 cities with the highest calls vs. the top 5 complaints (by Descriptor) in each city? (Visualized with a Pie Chart)
3. What are the top 20 call descriptors? (Visualized through a tag cloud)
4. What are the locations of the major call descriptors in each city?
5. How does the total calls trend yearly (Visualized with Line Chart) – Bonus analytical question that I was interested in as I wanted to see if the number of call increased or decreased over the years.
6. Finally, a summary dashboard will show the above visualizations.

## Data Type Conversion and Manipulation

Appendix 4 shows the Logstash configuration file that was used to ingest data into elasticsearch. Typically, data is ingested as text format by default. To use the field (column) "Date Created" in a proper

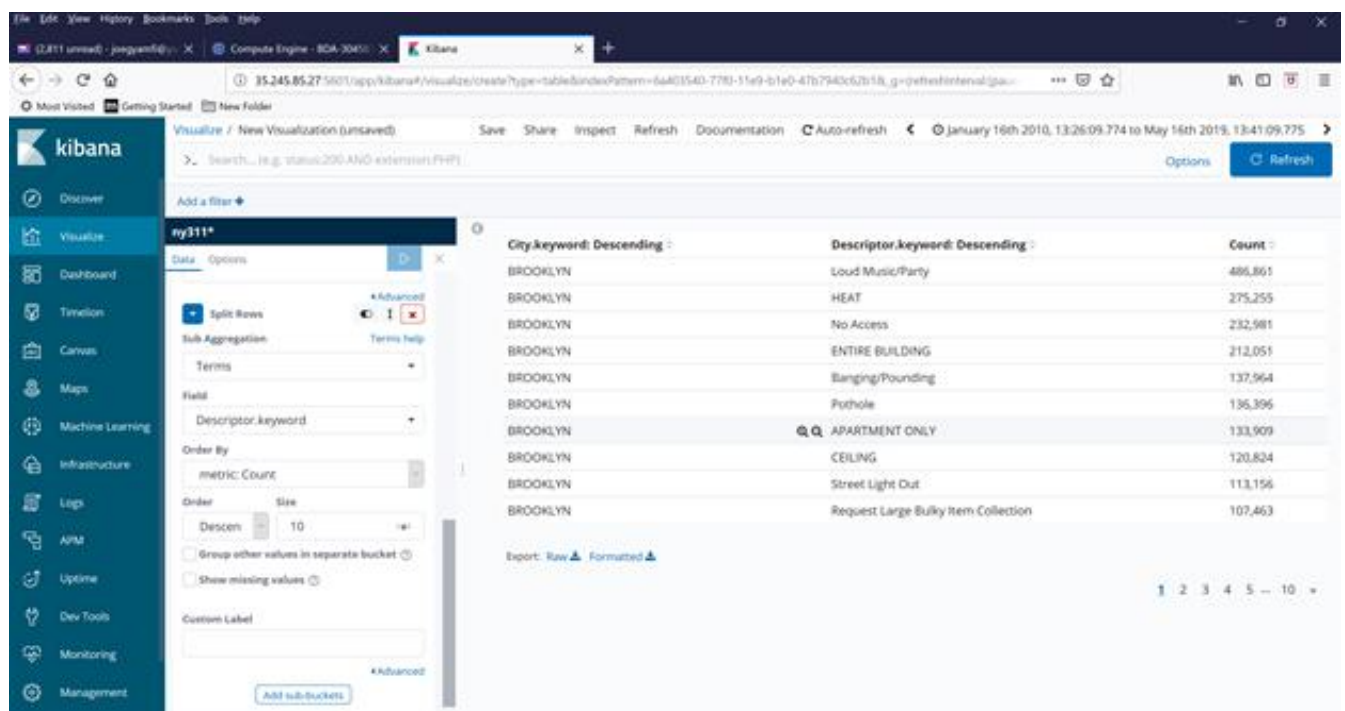
date format, a conversion was made and renamed as “Date”. Also, because the data in the “Location” column had a degree (°) symbol as part of the data, a decision was made to use the “Latitude” and “Longitude” fields and combine them into a new “Location” field that has a geo-point object type. Prior to combining them, the original location field was renamed as “Original Location”, and the “Latitude” and “Longitude” columns were converted into float types and then combined into a “Location” column.

Finally, preliminary analysis showed that the “City” column had some of the same cities duplicated in different formats as shown in the “Cities” pie chart under “Data Description & Exploratory Analysis” section above. For example, there were JAMAICA and Jamaica, FLUSHING and Flushing, ASTORIA and Astoria. Thus, the data in the “City” column was capitalized and any extra spaces removed so as to have only one city for those duplicated as shown in the filter plugin in the config file.

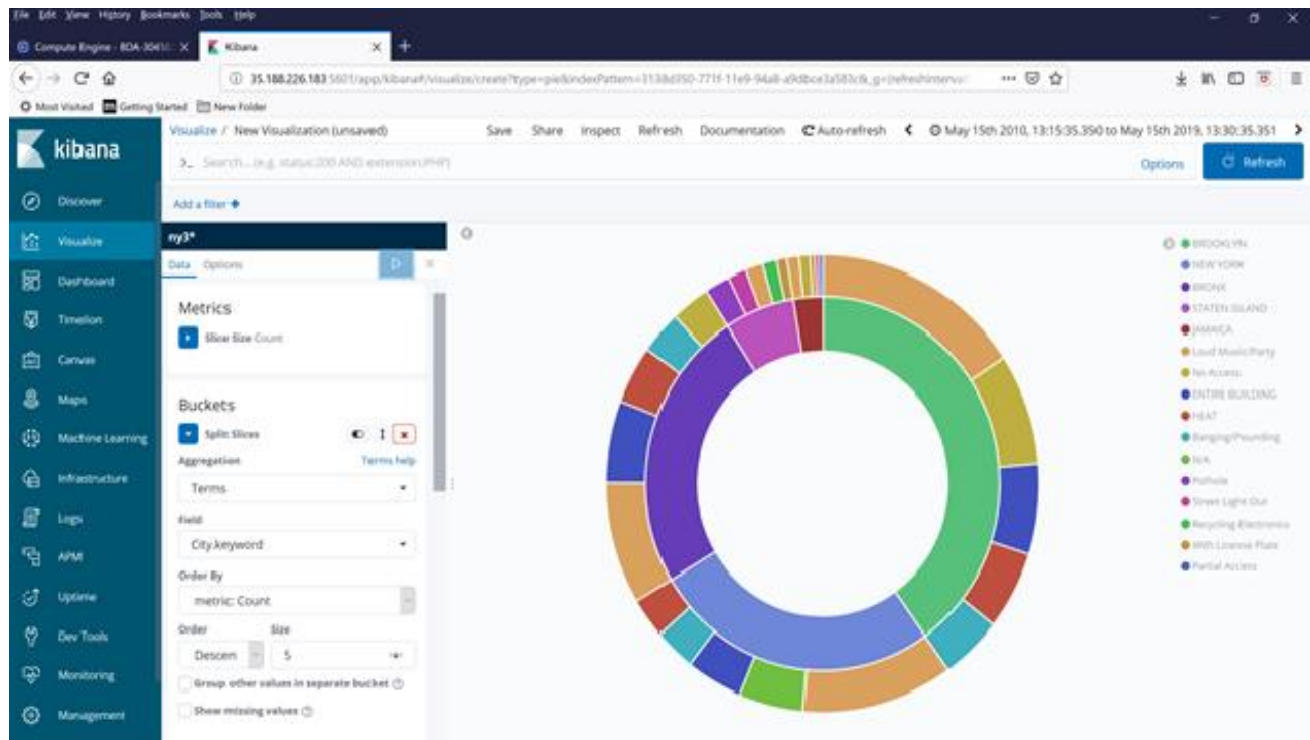
## Section A – Screenshots of tables and charts

The figures below show the screenshot of each visualization. The actual figures and analysis will follow after the screenshots.

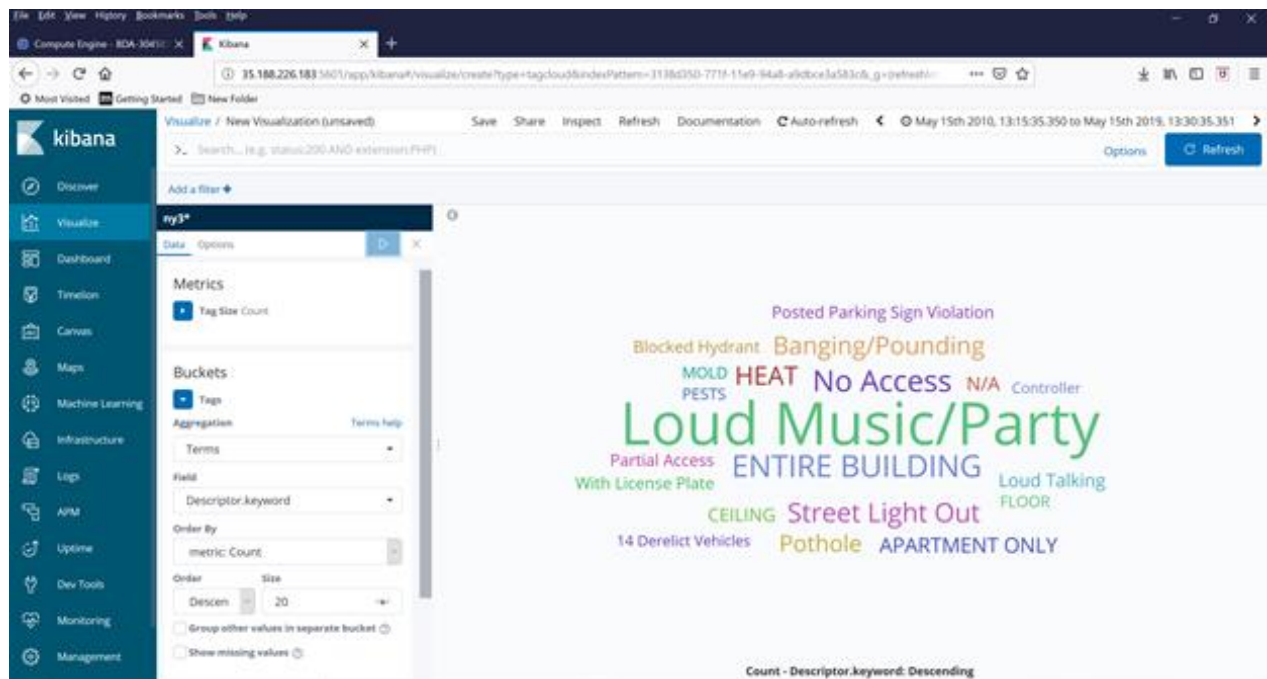
**Question 1: What are the top 10 cities with the highest calls vs. the top 10 complaints (by Descriptor) in each city? (Visualized with a table)**



**Question 2: What are the top 5 cities with the highest calls vs. the top 5 complaints (by Descriptor) in each city? (Visualized with a Pie Chart)**



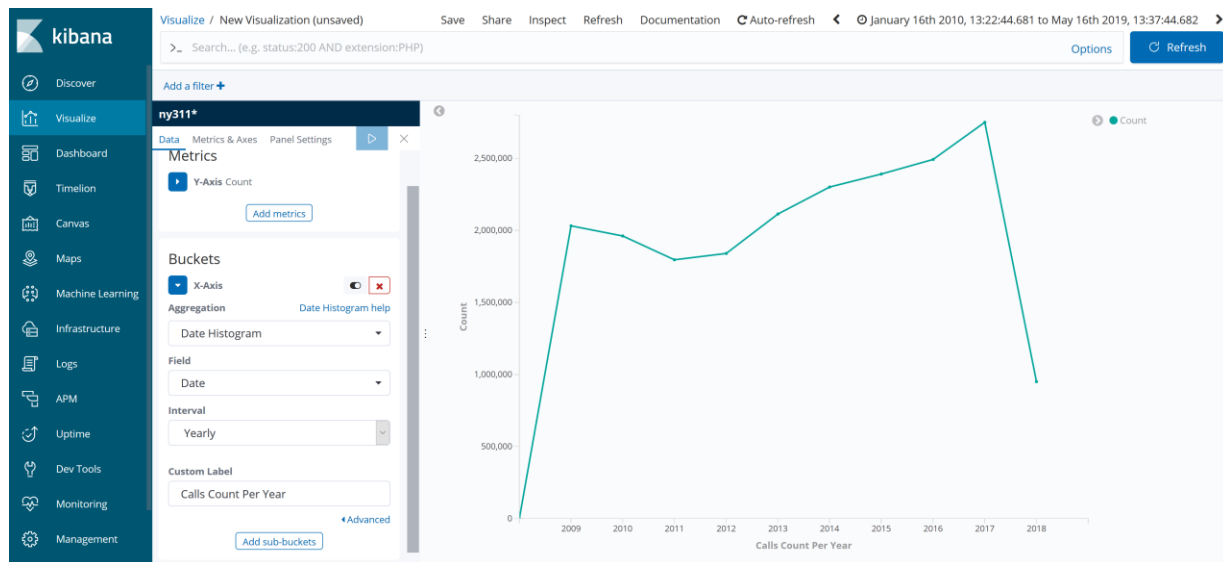
**Question 3: What are the top 20 call descriptors? (Visualized through a tag cloud)**



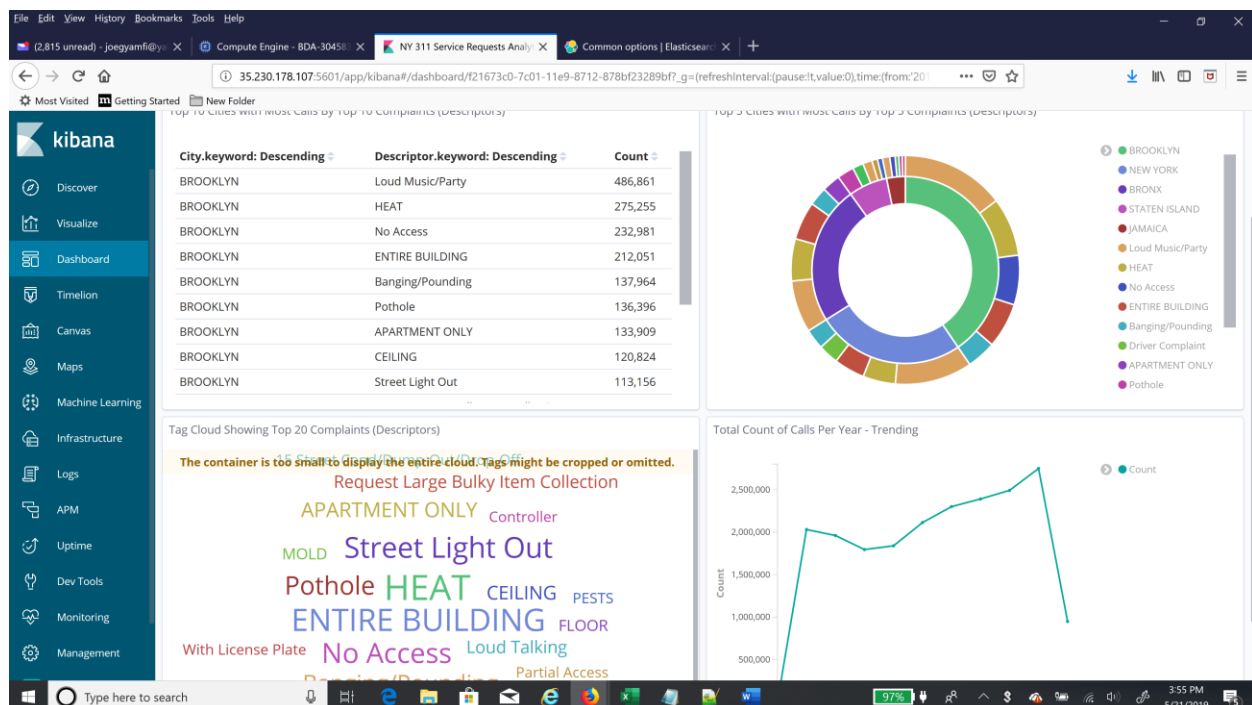
#### Question 4: What are the locations of the major call descriptors in each city?

Unfortunately, even though the latitude and longitude fields were converted from text to float, and subsequently used to create a new location field, the coordinate map visualization could not be generated.

#### Question 5 (Bonus) : What has been the trend of total calls over the years? (Visualized with Line Chart)



#### Question 6: Summary Dashboard



**Section B – Actual Figures and Analysis**

**a) Question 1: Data Table – Top 10 Cities With Most Calls Along With Top 10 Descriptors**

City.keyword: Descending	Descriptor.keyword: Descending	Count
BROOKLYN	Loud Music/Party	486,861
BROOKLYN	HEAT	275,255
BROOKLYN	No Access	232,981
BROOKLYN	ENTIRE BUILDING	212,051
BROOKLYN	Banging/Pounding	137,964
BROOKLYN	Pothole	136,396
BROOKLYN	APARTMENT ONLY	133,909
BROOKLYN	CEILING	120,824
BROOKLYN	Street Light Out	113,156
BROOKLYN	Request Large Bulky Item Collection	107,463
NEW YORK	Loud Music/Party	453,076
NEW YORK	HEAT	190,614
NEW YORK	ENTIRE BUILDING	180,240
NEW YORK	Driver Complaint	119,066
NEW YORK	Banging/Pounding	118,273
NEW YORK	Noise: Construction Before/After Hours (NM1)	106,276
NEW YORK	Loud Talking	95,483
NEW YORK	N/A	92,513
NEW YORK	Pothole	80,118
NEW YORK	APARTMENT ONLY	75,275



BRONX	Loud Music/Party	339,305
BRONX	HEAT	274,487
BRONX	ENTIRE BUILDING	248,378
BRONX	Banging/Pounding	121,774
BRONX	APARTMENT ONLY	119,751
BRONX	CEILING	109,118
BRONX	No Access	106,532
BRONX	MOLD	76,897
BRONX	FLOOR	72,887
BRONX	Pothole	65,797
STATEN ISLAND	Pothole	75,325
STATEN ISLAND	Street Light Out	48,029
STATEN ISLAND	Loud Music/Party	41,323
STATEN ISLAND	Recycling Electronics	24,824
STATEN ISLAND	Request Large Bulky Item Collection	22,803
STATEN ISLAND	With License Plate	22,276
STATEN ISLAND	1 Missed Collection	18,931
STATEN ISLAND	No Access	16,280
STATEN ISLAND	Dirty Water (WE)	16,092
STATEN ISLAND	Sewer Backup (Use Comments) (SA)	15,745
JAMAICA	Loud Music/Party	31,465
JAMAICA	No Access	23,253
JAMAICA	Street Light Out	15,404

JAMAICA	Pothole	14,913
JAMAICA	Sewer Backup (Use Comments) (SA)	13,269
JAMAICA	HEAT	13,251
JAMAICA	14 Derelict Vehicles	11,546
JAMAICA	With License Plate	10,877
JAMAICA	Illegal Conversion Of Residential Building/Space	9,690
JAMAICA	Partial Access	9,071
FLUSHING	No Access	20,444
FLUSHING	Pothole	15,244
FLUSHING	Loud Music/Party	12,476
FLUSHING	Street Light Out	12,293
FLUSHING	Partial Access	9,806
FLUSHING	HEAT	9,662
FLUSHING	ENTIRE BUILDING	9,538
FLUSHING	Banging/Pounding	8,635
FLUSHING	Request Large Bulky Item Collection	8,460
FLUSHING	Illegal Conversion Of Residential Building/Space	7,745
ASTORIA	Loud Music/Party	28,706
ASTORIA	No Access	21,007
ASTORIA	HEAT	10,021
ASTORIA	ENTIRE BUILDING	9,806
ASTORIA	Banging/Pounding	8,204
ASTORIA	Street Light Out	7,780

ASTORIA	Pothole	7,573
ASTORIA	Request Large Bulky Item Collection	7,096
ASTORIA	Partial Access	6,991
ASTORIA	Loud Talking	5,933
RIDGEWOOD	Loud Music/Party	16,869
RIDGEWOOD	No Access	14,740
RIDGEWOOD	Request Large Bulky Item Collection	11,842
RIDGEWOOD	Blocked Hydrant	8,298
RIDGEWOOD	HEAT	6,487
RIDGEWOOD	Street Light Out	6,143
RIDGEWOOD	Pothole	5,796
RIDGEWOOD	ENTIRE BUILDING	5,150
RIDGEWOOD	With License Plate	4,616
RIDGEWOOD	Partial Access	4,589
CORONA	No Access	21,699
CORONA	Loud Music/Party	17,111
CORONA	Partial Access	4,800
CORONA	HEAT	4,366
CORONA	ENTIRE BUILDING	4,054
CORONA	Street Light Out	3,858
CORONA	APARTMENT ONLY	3,571
CORONA	Pothole	3,297
CORONA	Banging/Pounding	2,728

CORONA	14 Derelict Vehicles	2,638
WOODSIDE	No Access	11,334
WOODSIDE	Loud Music/Party	10,659
WOODSIDE	ENTIRE BUILDING	5,584
WOODSIDE	Pothole	5,411
WOODSIDE	HEAT	5,042
WOODSIDE	Street Light Out	4,516
WOODSIDE	Partial Access	3,931
WOODSIDE	Loud Talking	3,211
WOODSIDE	Banging/Pounding	3,110
WOODSIDE	With License Plate	3,024

From the above table, the top ten (10) cities with the most calls in descending order are Brooklyn, New York, Bronx, Staten Island, Jamaica, Flushing, Astoria, Ridgewood, Corona and Woodside. Of these ten cities, Loud Music/Party came up as the top complaint in six (6) of them, No Access in three (3) and pothole in one (1).

The following tables provide further insights on this subset of data:

Row Labels	Sum of Count	%_of_Total
BROOKLYN	1,956,860	33.2%
BRONX	1,534,926	26.0%
NEW YORK	1,510,934	25.6%
STATEN ISLAND	301,628	5.1%
JAMAICA	152,739	2.6%
FLUSHING	114,303	1.9%
ASTORIA	113,117	1.9%
RIDGEWOOD	84,530	1.4%
CORONA	68,122	1.2%
WOODSIDE	55,822	0.9%
<b>Grand Total</b>	<b>5,892,981</b>	<b>100.0%</b>

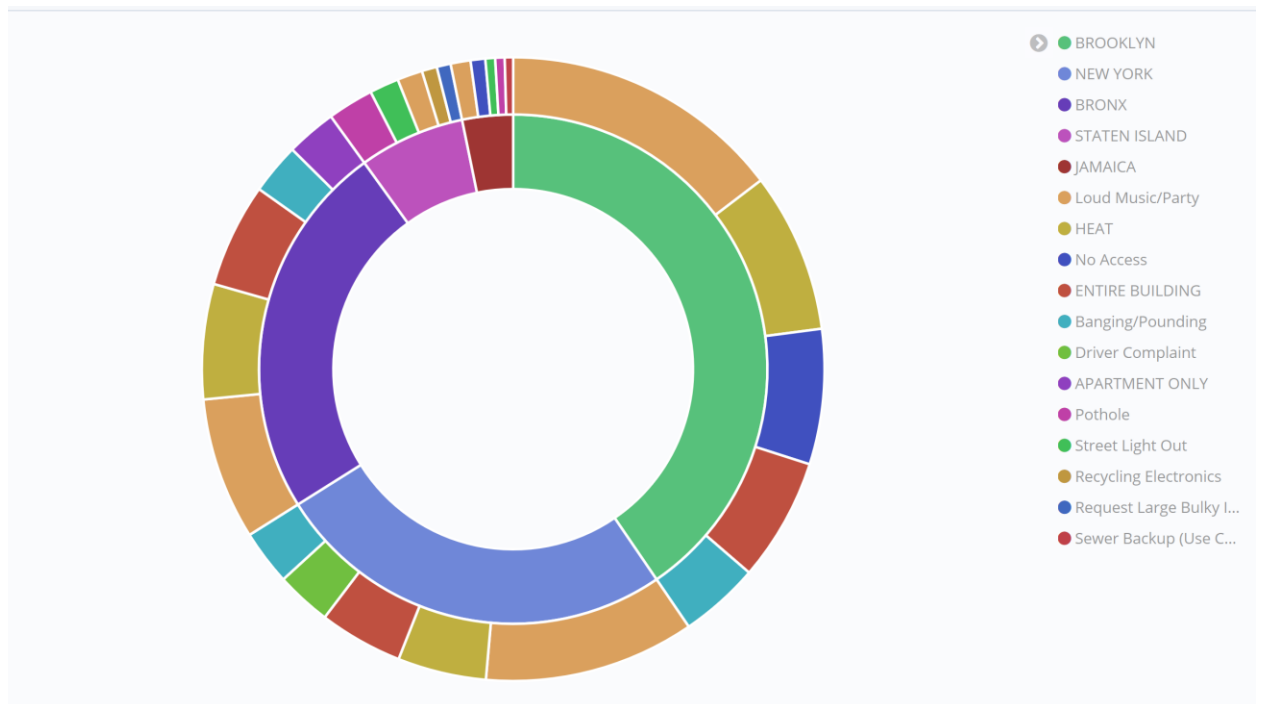
From the above table, the most complaints came from Brooklyn, followed by Bronx and New York cities with 33.2%, 26.0% and 25.6%, respectively, of total calls. These three cities accounted

for nearly 85% of the total calls for the top ten cities with most calls. This will serve as a useful guide to the city officials as to where they can allocate more resources.

Row Labels	Sum of Count	%_of_Total
Loud Music/Party	1,437,851	24.4%
HEAT	789,185	13.4%
ENTIRE BUILDING	674,801	11.5%
No Access	468,270	7.9%
Pothole	409,870	7.0%
Banging/Pounding	400,688	6.8%
APARTMENT ONLY	332,506	5.6%
CEILING	229,942	3.9%
Street Light Out	211,179	3.6%
Request Large Bulky Item Collection	157,664	2.7%
Driver Complaint	119,066	2.0%
Noise: Construction Before/After Hours (NM1)	106,276	1.8%
Loud Talking	104,627	1.8%
N/A	92,513	1.6%
MOLD	76,897	1.3%
FLOOR	72,887	1.2%
With License Plate	40,793	0.7%
Partial Access	39,188	0.7%
Sewer Backup (Use Comments) (SA)	29,014	0.5%
Recycling Electronics	24,824	0.4%
1 Missed Collection	18,931	0.3%
Illegal Conversion Of Residential Building/Space	17,435	0.3%
Dirty Water (WE)	16,092	0.3%
14 Derelict Vehicles	14,184	0.2%
Blocked Hydrant	8,298	0.1%
<b>Grand Total</b>	<b>5,892,981</b>	<b>100.0%</b>

From the above table, the top three complaint types were Loud Music/Party, Heat and Entire Building, accounting for nearly 50% of the total calls for the top ten cities with most calls and top ten calls. Again, a very useful guide when it comes to knowing the most prevalent complaint types and resources allocation planning.

**b) Question 2: Top 5 Cities With The Most Calls With The Top 5 Descriptors**



The chart shows that the top five cities with the most calls are Brooklyn, New York, Bronx, Staten Island and Jamaica in descending order of call count. The top five complaint types for each city are also shown in the chart. For example, In Brooklyn, the top five complaint types are Loud Music/Party, Heat, No Access, Entire Building and Banging/Pounding. The chart thus visually confirms the data and analysis in the Data Table.

c) Question 3: Top 20 Call Descriptors Using Tag Cloud

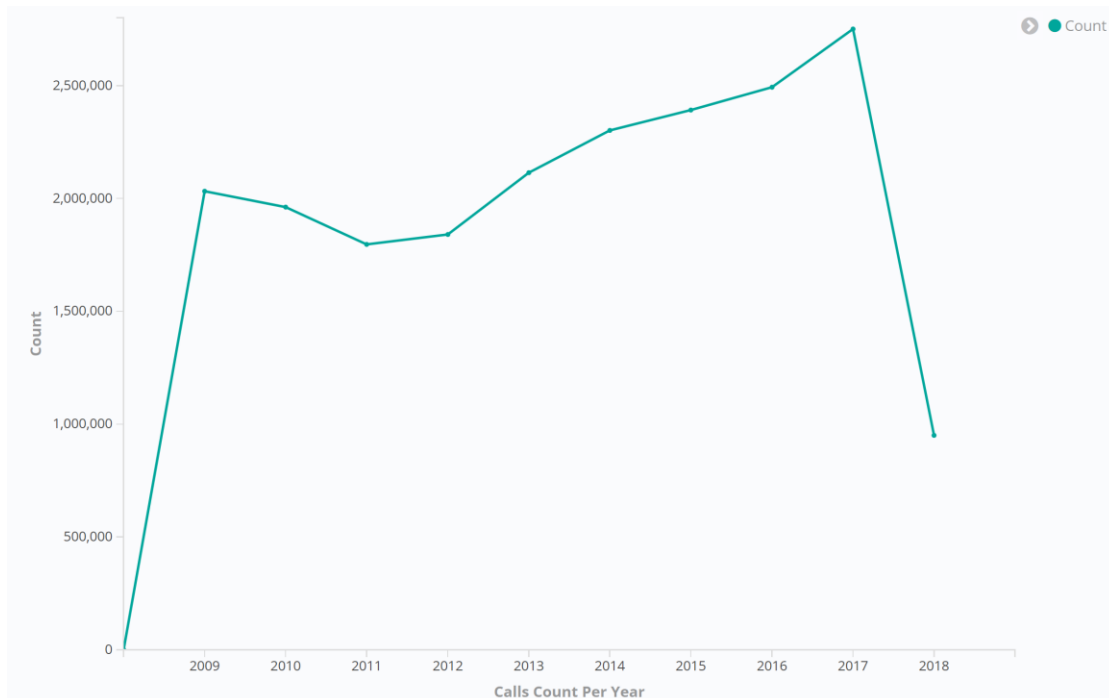


The above visualization also supports our earlier analysis. The larger the word (or phrase), the higher the count. Thus, the tag cloud shows that Loud Music/Party, Heat and Entire Building were the most complaint types.

d) Question 4: Map

*As explained earlier, the Coordinate Map could not be generated even though the latitude and longitude were properly converted to number (float) type prior to combining them into a location geo\_point object. In addition, the geo-point template was run in kibana before data was ingested in elasticsearch via logstash.*

**e) Question 5 (Bonus): The trend of total calls over the years**



The line chart shows the total count of calls per year from 2010 to 2019. Please note that the X-axis got shifted for some unknown reason that I couldn't identify. The first data point should start from 2010 and each one should be shifted one year to the right accordingly so that the last data point falls on 2019. In Kibana, when you hover around each dot (data point), it gives you the correct reading.

There were 2,031,828 total calls in 2010, decreasing to 1,796,282 in 2012, and then trending upward every year until it peaked at 2,711,850 in 2018. Current total calls count as of May 16, 2019 stood at 950,157.

## **Conclusion**

The objectives of the project were met. In particular, elk stack was demonstrated as a very powerful tool for big data analytics. Various analytical questions were answered that would be useful for the New York city authorities. Using such a tool to help analyze such a large and complex data in real time will enable the city respond in real time to various complaints. In addition, such analysis will be useful in effective resource planning and management whereby limited resources will be directed where needed most. For example, looking at the trend of calls, what types of calls are most prevalent and where they originate from, the city can create a plan to address them accordingly.

Moreover, in real time, knowing where calls are coming from and what type of complaints will also enable government and security officials respond appropriately with the right resources and adequate measures. Overall, the project has been a huge success apart from the coordinate map that could not be generated. I will continue to work on it for my own learning experience.

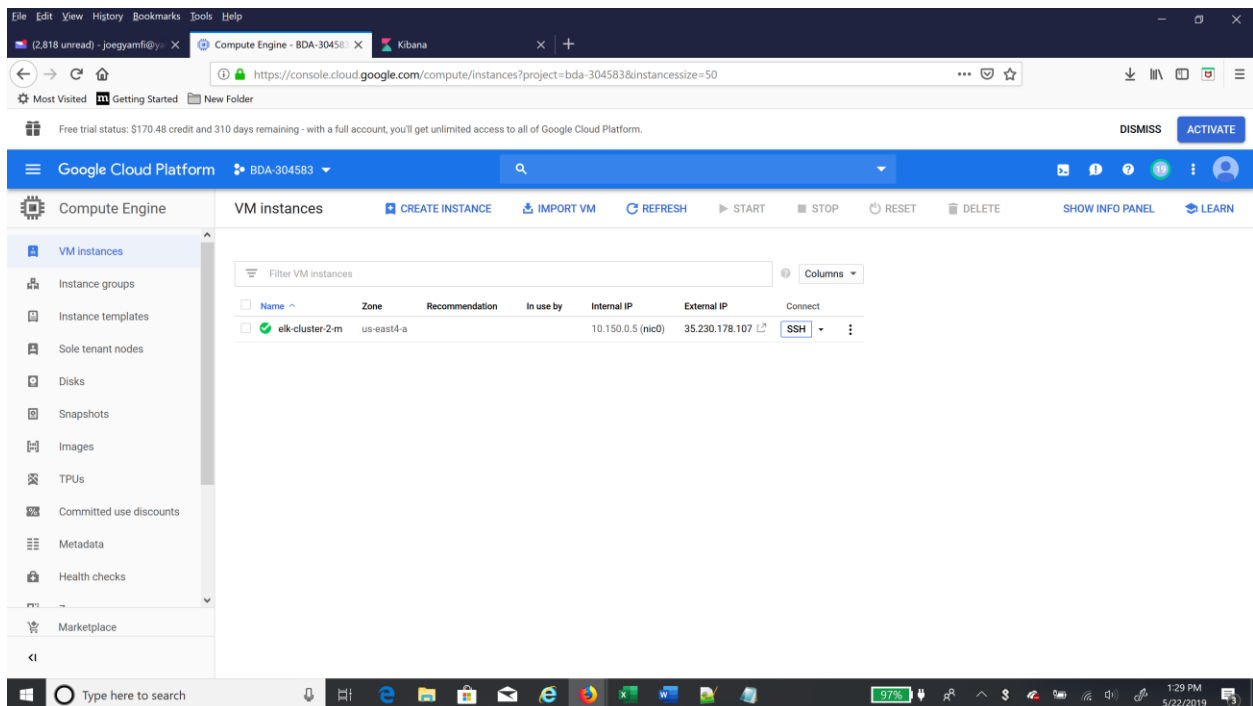


## References

1. New York Service Requests Data from 2010 to 2019 (NYC Open Data):  
<https://nycopendata.socrata.com/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9>
2. CSDA 1020 Big Data Analytics Tools Course Materials
3. Elasticsearch: A Complete Guide (Learning Path, End-to-end Search and Analytics). *Dixit, B. et al., 2017 Packt Publishing.*

## Appendix:

### Appendix 1: Create elkcluster & VM instances

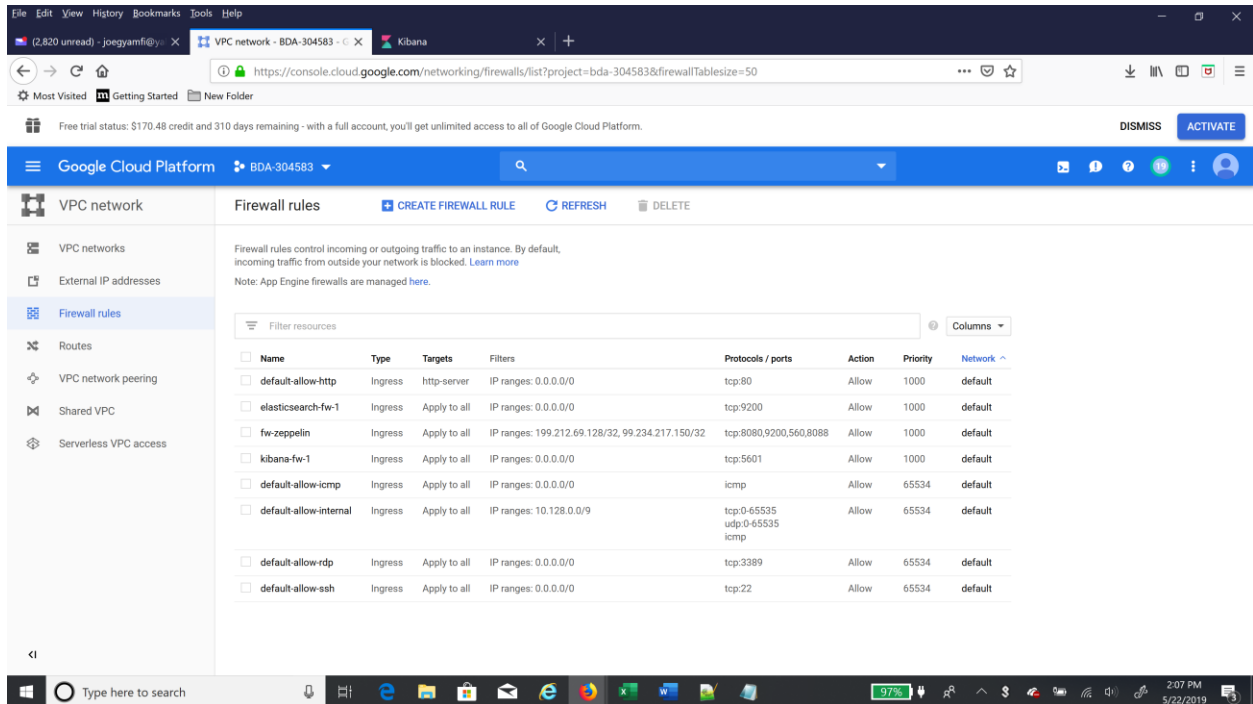


The screenshot displays the Google Cloud Platform console interface. The top navigation bar shows the 'Compute Engine' section selected. The left sidebar lists various resources, with 'VM instances' highlighted. The main content area shows a table of VM instances. A single instance, 'elk-cluster-2-m', is listed with the following details:

Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
elk-cluster-2-m	us-east4-a			10.150.0.5 (nic0)	35.230.178.107	SSH

The bottom of the image shows a Windows taskbar with the search bar and several application icons. The system clock indicates the time is 1:29 PM on 5/22/2019.

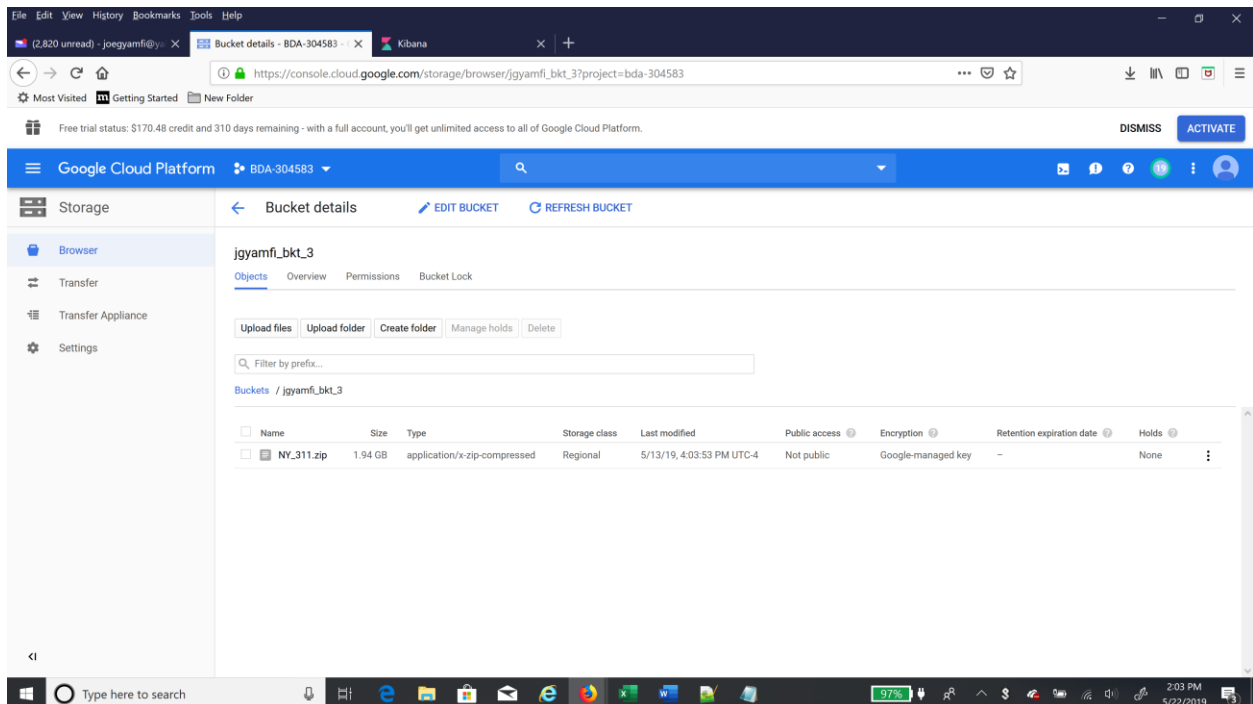
## Appendix 2: Firewall Rules



The screenshot shows the Google Cloud Platform console for project 'BDA-304583'. The 'Firewall rules' page is active, displaying a list of rules. The left sidebar shows the navigation menu with 'Firewall rules' selected. The main content area includes a 'Filter resources' search bar and a table of firewall rules. The table has columns for Name, Type, Targets, Filters, Protocols / ports, Action, Priority, and Network. The rules listed are: default-allow-http, elasticsearch-fw-1, fw-zeppelin, kibana-fw-1, default-allow-icmp, default-allow-internal, default-allow-rdp, and default-allow-ssh.

Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network
default-allow-http	Ingress	http-server	IP ranges: 0.0.0.0/0	tcp:80	Allow	1000	default
elasticsearch-fw-1	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:9200	Allow	1000	default
fw-zeppelin	Ingress	Apply to all	IP ranges: 199.212.69.128/32, 99.234.217.150/32	tcp:8080,9200,560,8088	Allow	1000	default
kibana-fw-1	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:5601	Allow	1000	default
default-allow-icmp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	icmp	Allow	65534	default
default-allow-internal	Ingress	Apply to all	IP ranges: 10.128.0.0/9	tcp:0-65535 udp:0-65535 icmp	Allow	65534	default
default-allow-rdp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:3389	Allow	65534	default
default-allow-ssh	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:22	Allow	65534	default

## Appendix 3: Storage bucket with NY 311 service requests data



The screenshot shows the Google Cloud Platform console for project 'BDA-304583'. The 'Storage' page is active, displaying the 'Bucket details' for 'jgyamfi\_bkt\_3'. The left sidebar shows the navigation menu with 'Storage' selected. The main content area includes tabs for 'Objects', 'Overview', 'Permissions', and 'Bucket Lock'. The 'Objects' tab is active, showing a list of objects. The table has columns for Name, Size, Type, Storage class, Last modified, Public access, Encryption, Retention expiration date, and Holds. The object 'NY\_311.zip' is listed with a size of 1.94 GB and a storage class of Regional.

Name	Size	Type	Storage class	Last modified	Public access	Encryption	Retention expiration date	Holds
NY_311.zip	1.94 GB	application/x-zip-compressed	Regional	5/13/19, 4:03:53 PM UTC-4	Not public	Google-managed key	-	None

#### **Appendix 4: Logstash NY311.Config file code**

```
input{
  file{
    path => "/home/gyamfiyaw31/NY_311.csv"
    start_position => "beginning"
    "sinedb_path" => "/dev/null"
  }
}

filter {
  # Step 1 drop the csv header line
  if [message] =~ /^#/ {
    drop {}
  }

  # Step 2 split columns
  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)", "Open Data Channel Type",
      "Park Facility Name", "Park Borough", "Vehicle Type", "Taxi Company Borough",
      "Taxi Pick Up Location", "Bridge Highway Name", "Bridge Highway Direction",
      "Road Ramp", "Bridge Highway Segment", "Latitude", "Longitude", "Location"]
  }

  # Step 3
```

# Rename Location (in degrees) and move latitude and longitude into a new location object for defined geo\_point type in ES

# Convert latitude and longitude data types into floats prior to creating the location geo\_point object

```
mutate {  
  rename => ["Location", "Original Location"]  
}  
mutate {  
  convert => ["Latitude", "float"]  
}  
mutate {  
  convert => ["Longitude", "float"]  
}  
mutate {  
  rename => ["Latitude", "[location][lat]", "Longitude", "[location][lon]"]  
}  
mutate {  
  uppercase => ["City"]  
  strip => ["City"]  
}  
  
# Step 4 Convert date data types to proper formats  
date {  
  locale => "eng"  
  match => ["Created Date", "MM/dd/yyyy HH:mm:ss aa", "ISO8601"]  
  target => "Date"  
  remove_field => ["Created Date"]  
}
```

```
}  
output {  
  elasticsearch {  
    hosts => "localhost"  
    index => "ny311data"  
    document_type => "locality"  
  }  
  # stdout {}  
}
```

### **Appendix 5: NY311 geo-point PUT Template**

PUT /\_template/ny311data

```
{  
  "order": 0,  
  "template": "ny311data*",  
  "mappings": {  
    "_default_": {  
      "properties": {  
        "Location": {  
          "type": "geo_point"  
        }  
      }  
    }  
  }  
}
```