CIS 4560 Term Project Tutorial

Authors: Jihyun Moon, James Logue, Hao Chen
Instructor: Jongwook Woo
Date: 12/15/2019

Lab Tutorial

NYC Ticket Analysis using Apache Hive

Objectives

In this hands-on lab, you will learn to:

- Set up a Hadoop cluster using Google Computing Services
- Set up authentication for cluster using SSH keys
- Move data from nodes into Hadoop file systems and back
- Hive commands to perform analysis
- Visualization

Platform Specifications

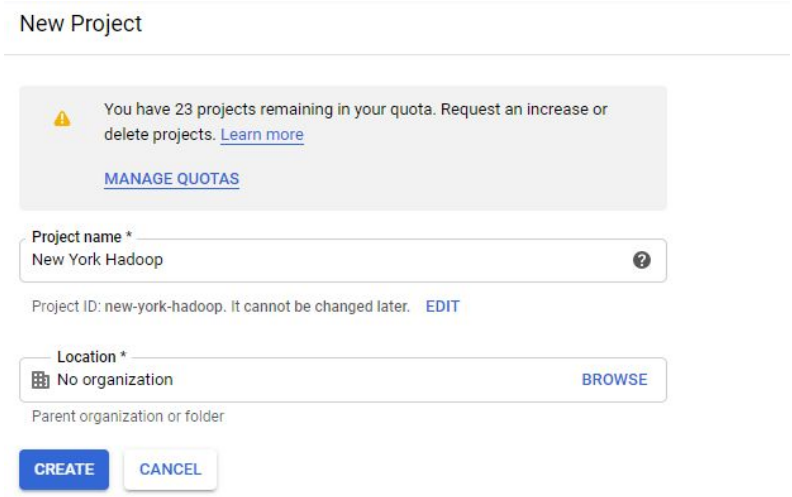
Google Dataproc

Master node
Standard (1 master, N workers)
Machine type
n1-standard-4 (4 vCPU, 15.0 GB memory)
Primary disk type
pd-standard
Primary disk size
500 GB
Worker nodes
2
Machine type
n1-standard-4 (4 vCPU, 15.0 GB memory)
Primary disk type
pd-standard
Primary disk size
500 GB
Local SSDs
0

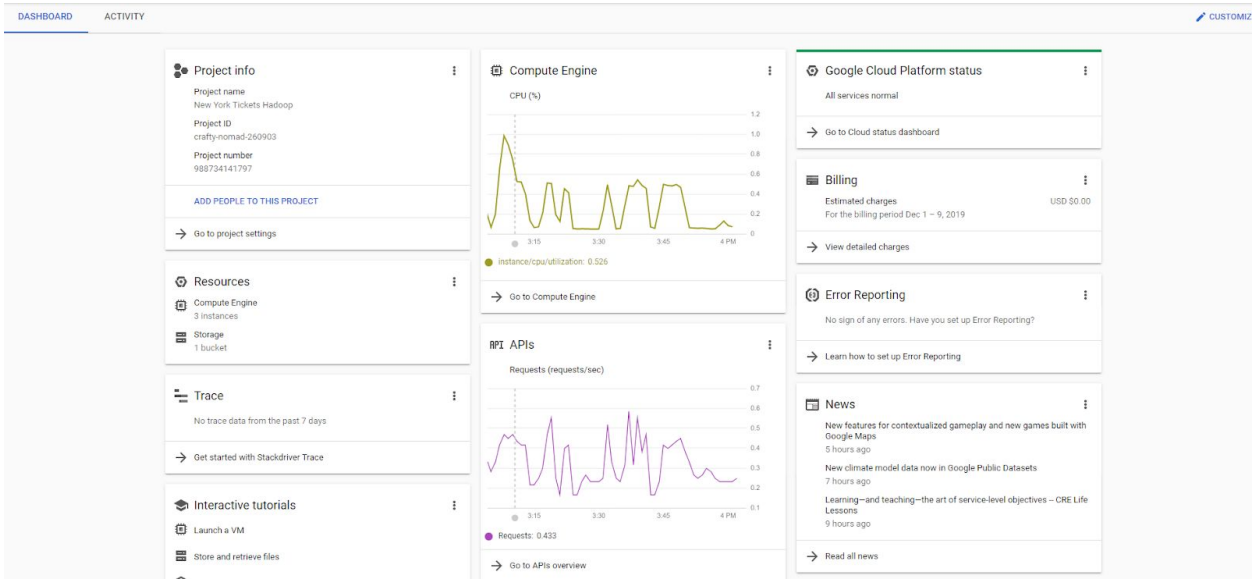
PART 1: Setting Up Cluster with Google Dataproc

1.Sign up for an account at https://cloud.google.com/

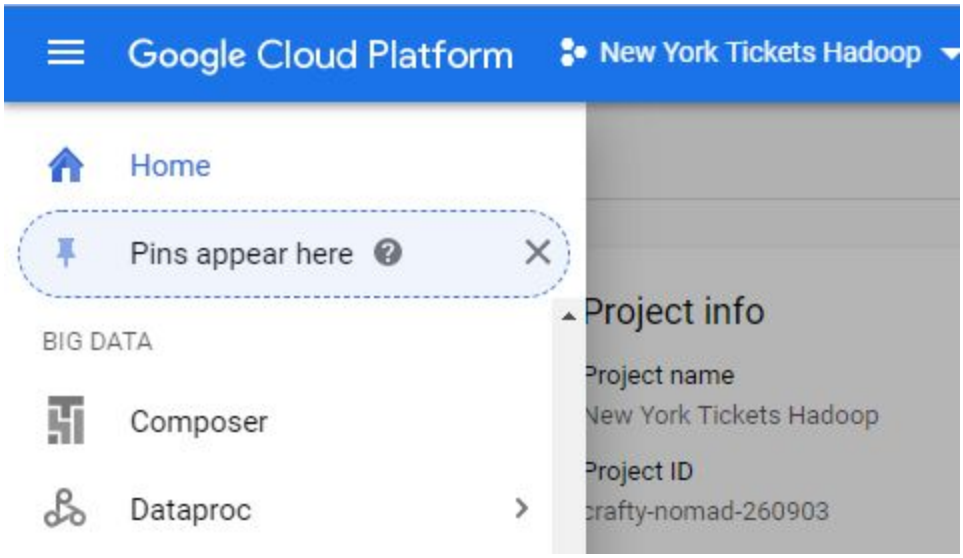
2.Upon logging in, create a project and name it accordingly



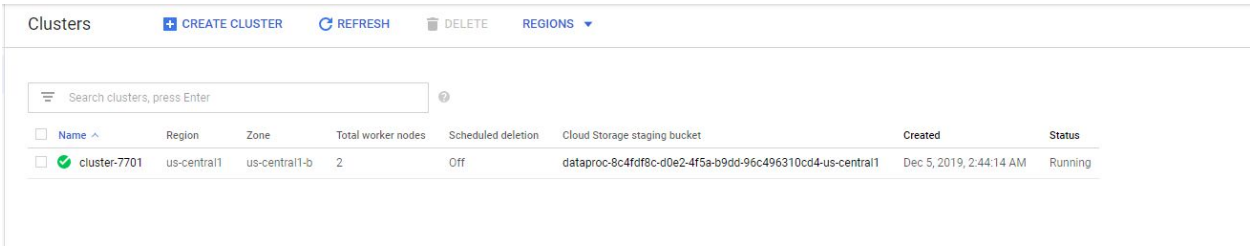
3.You will be greeted with a dashboard containing a lot of information



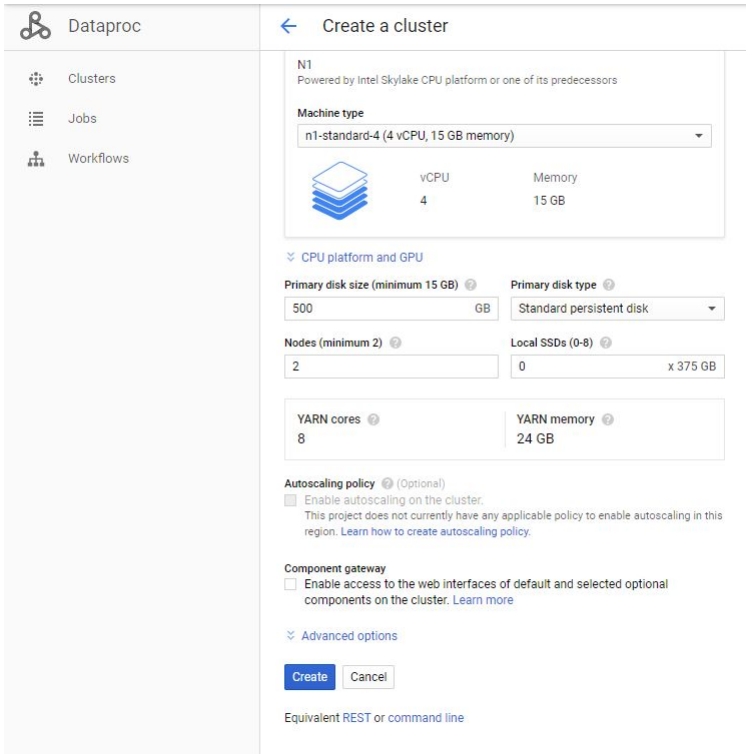
4.Using the navigation menu at the top left, access the Dataprocs tab



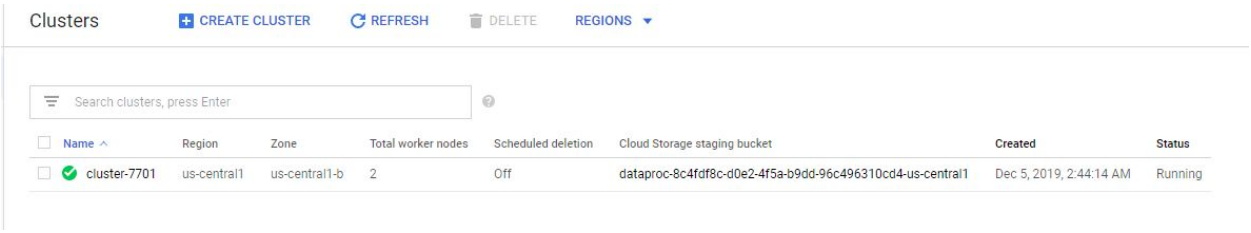
5. Create a cluster using the [Create Cluster] button



6.You can rename the cluster however you want and set hardware specifications on this page, when finished, click the [Create] button at the bottom

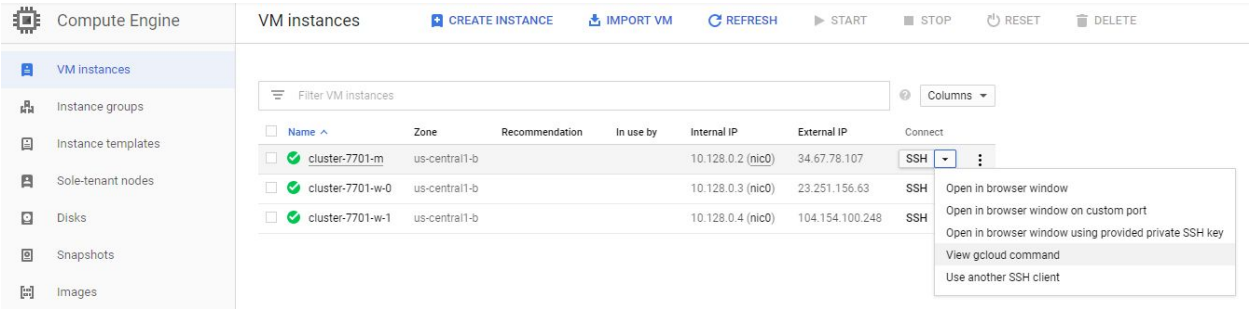


7.You now have set up a cluster and it is ready to use!



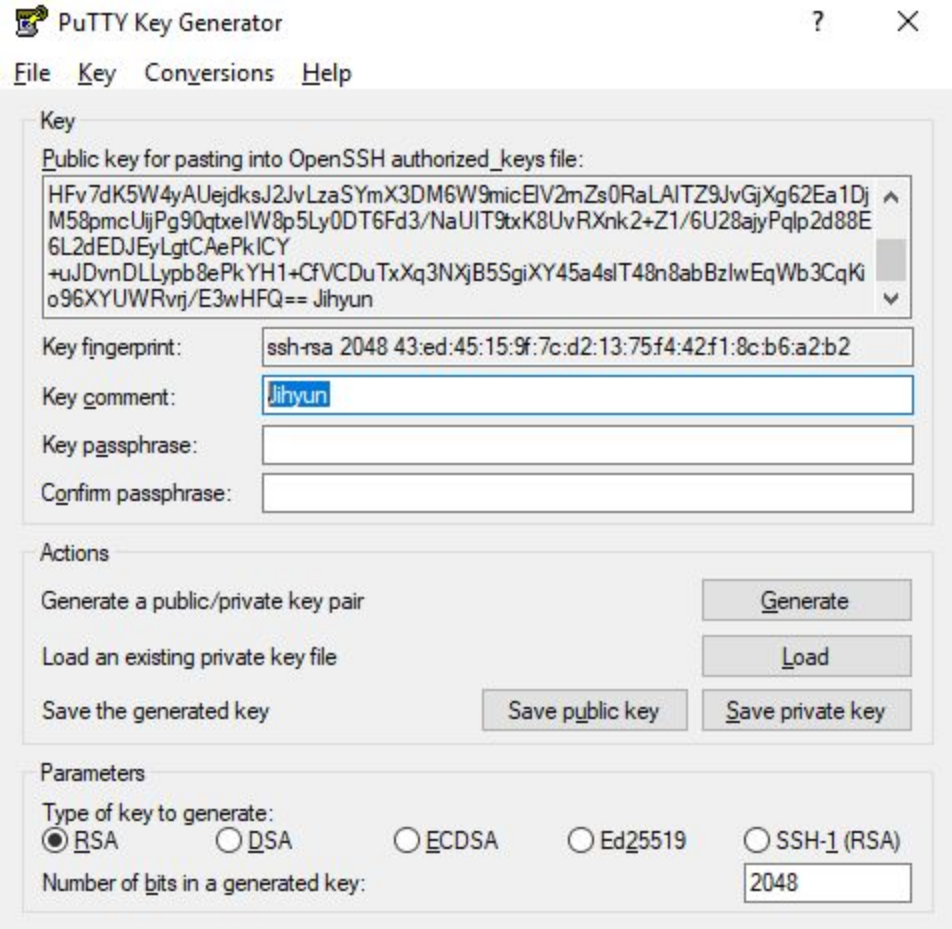
PART 2 : Setting up SSH Authentication

If you are setting this cluster up for use with others, you will need to set up SSH , if you are using this alone, you can just connect using the GCP console by navigating to Compute Engine>VM Instances>Connect

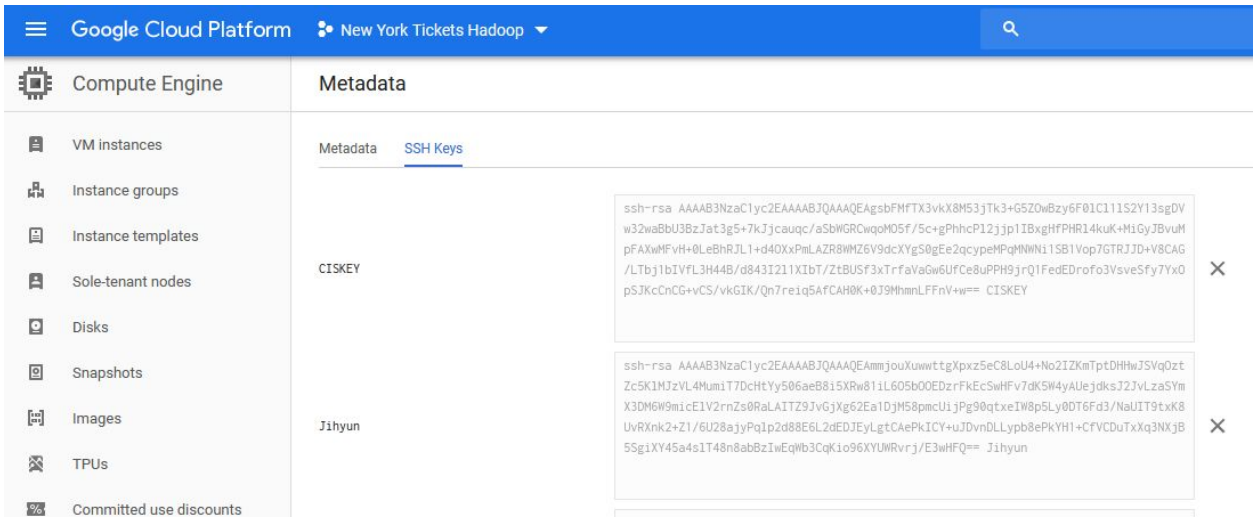


1. In order to connect using SSH, we will need to add the public keys into the metadata of the project. For this tutorial, we will be adding the keys on a project wide level and not an instance wide level.
2. Download and run puttygen.exe https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

3. Fill in the key comment as the username you will create to ssh into and press the generate button and save the private key in another location.

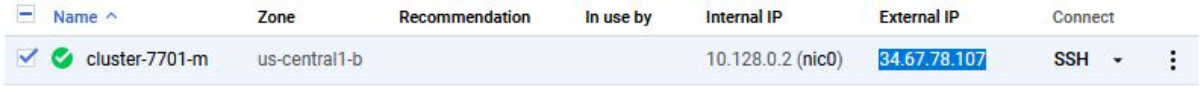


4. Copy and paste the public key above into Google Compute Engine under Metadata>Add SSH Keys

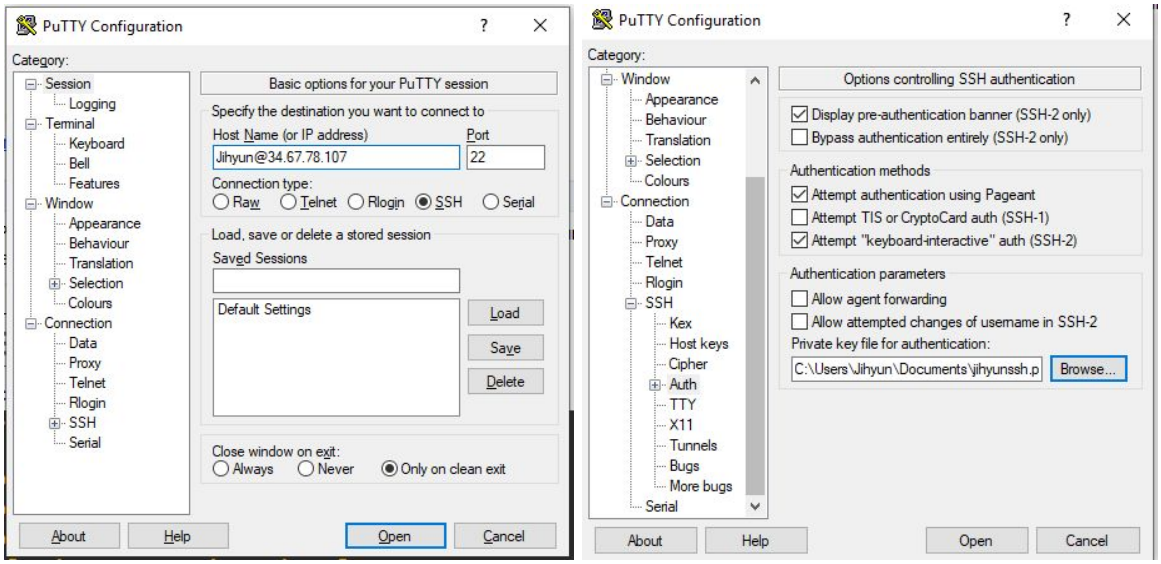


PART 3: Connect to Master Node using PUTTY

Download and open up putty.exe from
https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

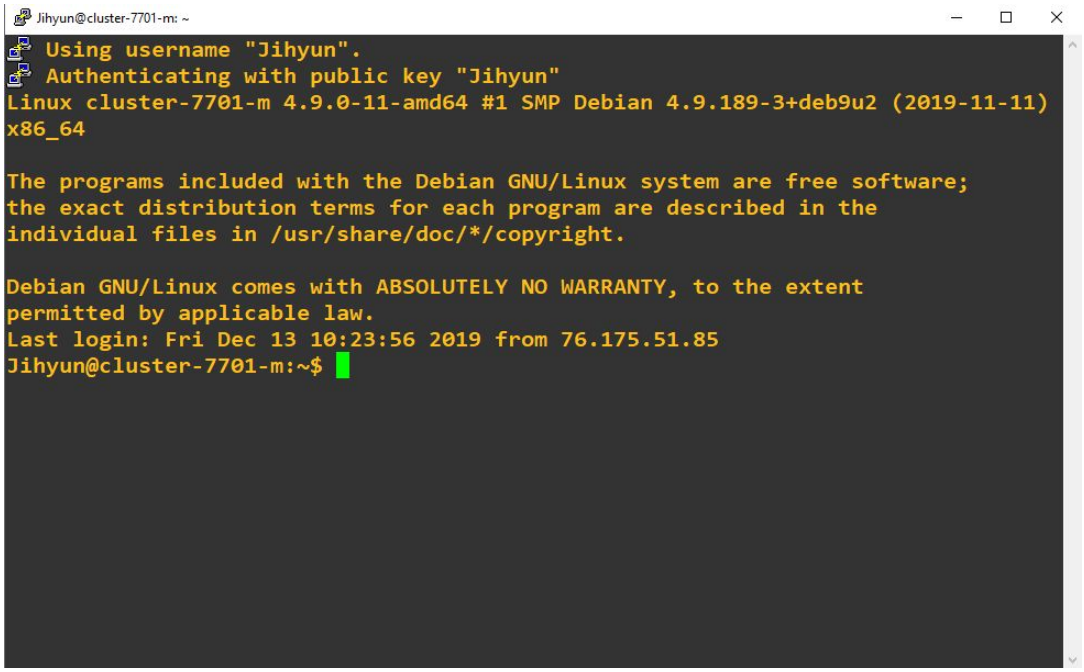


Using the External IP from the Compute Engine>VM Instances, use Putty to connect to your master node with the private key you saved earlier.



The host name should follow : [username]@externalip

Set the private key from above under Connection>SSH>Auth and connect by clicking [Open]



PART 4: Download data and set up for hdfs

Download datasets using the wget command

```
8  wget -O Parking_Violations_Issued_-_Fiscal_Year_2014.csv https://data.cityofnewyork.us/api/views/jt7v-77mi/rows.csv?accessType=DOWNLOAD
9  wget -O Parking_Violations_Issued_-_Fiscal_Year_2015.csv https://data.cityofnewyork.us/api/views/c284-tqph/rows.csv?accessType=DOWNLOAD
10 wget -O Parking_Violations_Issued_-_Fiscal_Year_2016.csv https://data.cityofnewyork.us/api/views/kiv2-tbus/rows.csv?accessType=DOWNLOAD
11 wget -O Parking_Violations_Issued_-_Fiscal_Year_2017.csv https://data.cityofnewyork.us/api/views/2bnn-yakx/rows.csv?accessType=DOWNLOAD
12
```

wget -O Parking_Violations_Issued_-_Fiscal_Year_2014.csv
https://data.cityofnewyork.us/api/views/jt7v-77mi/rows.csv?accessType=DOWNLOAD
wget -O Parking_Violations_Issued_-_Fiscal_Year_2015.csv
https://data.cityofnewyork.us/api/views/c284-tqph/rows.csv?accessType=DOWNLOAD
wget -O Parking_Violations_Issued_-_Fiscal_Year_2016.csv
https://data.cityofnewyork.us/api/views/kiv2-tbus/rows.csv?accessType=DOWNLOAD
wget -O Parking_Violations_Issued_-_Fiscal_Year_2017.csv
https://data.cityofnewyork.us/api/views/2bnn-yakx/rows.csv?accessType=DOWNLOAD

Check files are in local machine with the -ls command

```
Jihyun@cluster-7701-m:~$ ls
NYC
Parking_Violations_Issued_-_Fiscal_Year_2014.csv
Parking_Violations_Issued_-_Fiscal_Year_2015.csv
Parking_Violations_Issued_-_Fiscal_Year_2016.csv
Parking_Violations_Issued_-_Fiscal_Year_2017.csv
```

Merge all csv files into one using Paste and the wildcard *

```
Jihyun@cluster-7701-m:~$ paste *.csv > NewYorkCombined.csv
```

Double check that merged file exists using -ls

```
Jihyun@cluster-7701-m:~$ ls
NewYorkCombined.csv
NYC
Parking_Violations_Issued_-_Fiscal_Year_2014.csv
Parking_Violations_Issued_-_Fiscal_Year_2015.csv
Parking_Violations_Issued_-_Fiscal_Year_2016.csv
Parking_Violations_Issued_-_Fiscal_Year_2017.csv
Jihyun@cluster-7701-m:~$
```

Create a folder in hdfs using -mkdir

```
Jihyun@cluster-7701-m:~$ hdfs dfs -mkdir /NYC
```

Double check to see it is created using -ls

```
Jihyun@cluster-7701-m:~$ hdfs dfs -ls /
Found 5 items
drwxr-xr-x   - Jihyun hadoop          0 2019-12-06 11:06 /NYC
drwx------   - mapred hadoop          0 2019-12-05 10:45 /hadoop
drwxr-xr-x   - Jihyun hadoop          0 2019-12-05 13:00 /test
drwxrwxrwt   - hdfs   hadoop          0 2019-12-05 10:45 /tmp
drwxrwxrwt   - hdfs   hadoop          0 2019-12-05 10:45 /user
```

Move the combined csv file into hdfs using the -put command

```
Jihyun@cluster-7701-m:~$ hdfs dfs -put /home/Jihyun/NewYorkCombined.csv /NYC
```

Double check to see -put command went through successfully using -ls

```
Jihyun@cluster-7701-m:~$ hdfs dfs -ls /NYC
Found 1 items
-rw-r--r--   2 Jihyun hadoop 8462305792 2019-12-06 11:10 /NYC/NewYorkCombined.cs
v
```

Also create two more directories in hdfs called /Output/ and /Output/ticketsfinal/

```
Jihyun@cluster-7701-m:~$ hdfs dfs -mkdir /output
Jihyun@cluster-7701-m:~$ hdfs dfs -mkdir /output/ticketsfinal
```

Access apache hive by using the beeline command:
beeline -u jdbc:hive2://localhost:10000/default -n [username]

```
Jihyun@cluster-7701-m:~$ beeline -u jdbc:hive2://localhost:10000/default -n Jihy
un
Connecting to jdbc:hive2://localhost:10000/default
Connected to: Apache Hive (version 2.3.5)
Driver: Hive JDBC (version 2.3.5)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 2.3.5 by Apache Hive
```

Create a table from the combined csv file using a hive query

```
0: jdbc:hive2://localhost:10000/default> create table tickets
. . . . . .> (summons_number int, plate_id string, registration_state string, plate_type string, issue_date date, violation_code i
nt, vehicle_body_type string, vehicle_make string, issuing_agency string, street_code1 int, street_code2 int, street_code3 int,
. . . . . .> vehicle_expiration_date int, violation_location string, violation_precinct int, issuer_precinct int, issuer_code int,
 issuer_command string, issuer_squad string, violation_time string, time_first_observed string, violation_county string,
. . . . . .> violation_in_front_of_or_opposite string, house_number string, street_name string, intersecting_street string, date_f
irst_observed int, law_section int, sub_division string, violation_legal_code string, days_parking_in_effect string,
. . . . . .> from_hours_in_effect string, to_hours_in_effect string, vehicle_color string, unregistered_vehicle string, vehicle_ye
ar int, meter_number string, feet_from_curb int, violation_post_code string, violation_description string,
. . . . . .> no_standing_or_stopping_violation string, hydrant_violation string, double_parking_violation string)
. . . . . .> ROW FORMAT DELIMITED
. . . . . .> FIELDS TERMINATED BY ','
. . . . . .> STORED AS TEXTFILE
. . . . . .> TBLPROPERTIES("skip.header.line.count"="1");
```

create table tickets
(summons_number int, plate_id string, registration_state string, plate_type string, issue_date string, violation_code int, vehicle_body_type string, vehicle_make string, issuing_agency string, street_code1 int, street_code2 int, street_code3 int,
vehicle_expiration_date int, violation_location string, violation_precinct int, issuer_precinct int, issuer_code int, issuer_command string, issuer_squad string, violation_time string, time_first_observed string, violation_county string,
violation_in_front_of_or_opposite string, house_number string, street_name string, intersecting_street string, date_first_observed int, law_section int, sub_division string, violation_legal_code string, days_parking_in_effect string,

```
from_hours_in_effect string, to_hours_in_effect string, vehicle_color string,
unregistered_vehicle string, vehicle_year int, meter_number string, feet_from_curb int,
violation_post_code string, violation_description string,
no_standing_or_stopping_violation string, hydrant_violation string, double_parking_violation
string)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION "/Output/"
TBLPROPERTIES("skip.header.line.count"="1");
```

Load data into table using the command LOAD

```
0: jdbc:hive2://localhost:10000/default> LOAD DATA INPATH '/NYC/NewYorkCombined.csv' OVERWRITE INTO TABLE tickets;
No rows affected (1.542 seconds)
0: jdbc:hive2://localhost:10000/default>
```

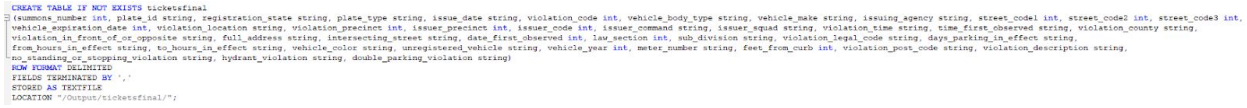
```
LOAD DATA INPATH '/NYC/NewYorkCombined.csv' OVERWRITE INTO TABLE tickets;
```

Run a test command (select first 20 records of vehicle colors)

```
0: jdbc:hive2://localhost:10000/default> select vehicle_color from tickets limit 20;
+----------------+
| vehicle_color  |
+----------------+
| BLACK          |
| BRN            |
| BLUE           |
| SILVR          |
| WHITE          |
| BLK            |
| YELLO          |
| BLK            |
| WH             |
| GREY           |
| BK             |
| ORANG          |
| SILVE          |
| GR             |
| WHITE          |
| BLU            |
| SILVE          |
| BR             |
| BLK            |
| WHITE          |
+----------------+
20 rows selected (2.382 seconds)
```

vehicle_year	int	
meter_number	string	
feet_from_curb	int	
violation_post_code	string	
violation_description	string	
no_standing_or_stopping_violation	string	
hydrant_violation	string	
double_parking_violation	string	
state	string	

Create a second table called ticketsfinal using CREATE TABLE that will hold concatenation and fill the new column with data and we will save the file in hdfs in a folder called Output/ticketsfinal/
check to see table is made using SHOW TABLES;



```
CREATE TABLE IF NOT EXISTS ticketsfinal
(summons_number int, plate_id string, registration_state string, plate_type string, issue_date string, violation_code int, vehicle_body_type string, vehicle_make string, issuing_agency string, street_code1 int, street_code2 int, street_code3 int,
vehicle_expiration_date int, violation_location string, violation_precinct int, issuer_precinct int, issuer_code int, issuer_command string, issuer_squad string, violation_time string, time_first_observed string, violation_county string,
violation_in_front_of_or_opposite string, full_address string, intersecting_street string, date_first_observed int, law_section int, sub_division string, violation_legal_code string, days_parking_in_effect string,
from_hours_in_effect string, to_hours_in_effect string, vehicle_color string, unregistered_vehicle string, vehicle_year int, meter_number string, feet_from_curb int, violation_post_code string, violation_description string,
no_standing_or_stopping_violation string, hydrant_violation string, double_parking_violation string)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION "/Output/ticketsfinal/";
```


CREATE TABLE IF NOT EXISTS ticketsfinal
(summons_number int, plate_id string, registration_state string, plate_type string, issue_date string, violation_code int, vehicle_body_type string, vehicle_make string, issuing_agency string, street_code1 int, street_code2 int, street_code3 int,
vehicle_expiration_date int, violation_location string, violation_precinct int, issuer_precinct int, issuer_code int, issuer_command string, issuer_squad string, violation_time string, time_first_observed string, violation_county string,
violation_in_front_of_or_opposite string, full_address string, intersecting_street string, date_first_observed int, law_section int, sub_division string, violation_legal_code string, days_parking_in_effect string,
from_hours_in_effect string, to_hours_in_effect string, vehicle_color string, unregistered_vehicle string, vehicle_year int, meter_number string, feet_from_curb int, violation_post_code string, violation_description string,
no_standing_or_stopping_violation string, hydrant_violation string, double_parking_violation string)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION "/Output/ticketsfinal/";

```
+-------------------+
|     tab_name      |
+-------------------+
| tickets           |
| tickets_summary   |
| ticketsfinal      |
| ticketstest       |
| ticketsv2         |
+-------------------+
```

Insert data into ticketsfinal using data from tickets with the INSERT OVERWRITE TABLE

```
INSERT OVERWRITE TABLE ticketsfinal
SELECT summons_number, plate_id, registration_state, plate_type, issue_date, violation_code, vehicle_body_type, vehicle_make, issuing_agency, street_code1, street_code2, street_code3,
vehicle_expiration_date, violation_location, violation_precinct, issuer_precinct, issuer_Code, issuer_command, issuer_squad, violation_time, time_first_observed, violation_county,
violation_in_front_of_or_opposite, CONCAT(house_number,' ',street_name,' ', case WHEN summons_number is NOT NULL then 'New York' ELSE 'New York' END) AS full_address, intersecting_street, date_first_observed, law_section,
sub_division, violation_legal_code, days_parking_in_effect,from_hours_in_effect, to_hours_in_effect, vehicle_color, unregistered_vehicle, vehicle_year, meter_number, feet_from_curb, violation_post_code, violation_description,
no_standing_or_stopping_violation, hydrant_violation, double_parking_violation
FROM tickets;
```

```
INSERT OVERWRITE TABLE ticketsfinal
SELECT summons_number, plate_id, registration_state, plate_type, issue_date,
violation_code, vehicle_body_type, vehicle_make, issuing_agency, street_code1,
street_code2, street_code3,
vehicle_expiration_date, violation_location, violation_precinct, issuer_precinct, issuer_Code,
issuer_command, issuer_squad, violation_time, time_first_observed, violation_county,
violation_in_front_of_or_opposite, CONCAT(house_number,' ',street_name,' ', case WHEN
summons_number is NOT NULL then 'New York' ELSE 'New York' END) AS full_address,
intersecting_street, date_first_observed, law_section,
 sub_division, violation_legal_code, days_parking_in_effect,from_hours_in_effect,
to_hours_in_effect, vehicle_color, unregistered_vehicle, vehicle_year, meter_number,
feet_from_curb, violation_post_code, violation_description,
no_standing_or_stopping_violation, hydrant_violation, double_parking_violation
FROM tickets;
```

Check to see that the street names are concatenated with the state name using SELECT.

```
0: jdbc:hive2://localhost:10000/default> select summons_number, full_address fro
m ticketsfinal limit 50;
+-----------------+-----------------------------------+
| summons_number  |           full_address            |
+-----------------+-----------------------------------+
| 1361929741      | 959 E 5 ST New York               |
| 1366962000      | 185 MARINE AVENUE New York        |
| 1342296187      | 60-25 56 ST New York              |
| 1342296199      | 60-12 56 ST New York              |
| 1342296217      | 54-14 ANDREWS AVE New York        |
| 1356906515      | 4165 BROADWAY New York            |
| 1337077380      | 99-01 34 AVE New York             |
| 1364523796      | 1017 THOMAS BOYLAND ST New York   |
| 1359914924      | 48 7 AVE New York                 |
| 1355498326      | 7003 FT HAMILTON PKWY New York    |
| 1361272259      | 205 W 39 ST New York              |
| 1360588267      |  I8OMASAMMK New York              |
| 1360588279      | 160 HAVEMEYER ST New York         |
| 1360016156      | 340 JAY ST New York               |
| 1255986920      |  SOUTH STREET New York            |
| 1359121262      | 149 36 124 ST New York            |
| 1350454229      | 669 DRAKE ST New York             |
| 1364684342      | 1622 W 125 New York               |
| 1365454538      | 49-11 BROADWAY New York           |
| 1357066697      | 98-27 50 AVE New York             |
| 1366144776      | 545 1 AVE New York                |
| 1347701394      | 273 MONROE ST New York            |
| 1347701400      | 273 MONROE ST New York            |
| 1359039533      | 450 55 ST New York                |
| 1358530051      | 22-03 93 ST New York              |
| 1364781992      | 241-15 NORTHERN BLVD New York     |
| 1357082800      | 46-01 108 ST New York             |
| 1356720614      | 87-77A PARSONS BLVD New York      |
```

PART 5 : DOWNLOAD FILE

Hdfs to find the file located in the saved location from the code in Output/ticketsfinal

```
Jihyun@cluster-7701-m:~$ hdfs dfs -ls /Output/ticketsfinal
Found 1 items
-rwxrwxrwt   2 Jihyun hadoop 2321270561 2019-12-09 23:49 /Output/ticketsfinal/00
0000_0
```

Move the file into the master node as a csv file using hdfs dfs -get

```
Jihyun@cluster-7701-m:~$ hdfs dfs -get /Output/ticketsfinal/000000_0 TicketsFinal.csv
```

-ls to verify the csv file is there

```
Jihyun@cluster-7701-m:~$ ls
NewYorkCombined.csv  Parking_Violations_Issued_-_Fiscal_Year_2014.csv  Parking_Violations_Issued_-_Fiscal_Year_2016.csv  TicketsFinal.csv
NYC                  Parking_Violations_Issued_-_Fiscal_Year_2015.csv  Parking_Violations_Issued_-_Fiscal_Year_2017.csv  Ticketsv2.csv
Jihyun@cluster-7701-m:~$
```

Using pscp, download the csv file from the master node onto your machine using the syntax Pscp -i [location of private key] [username]@(host ip):[source destination] (download destination)

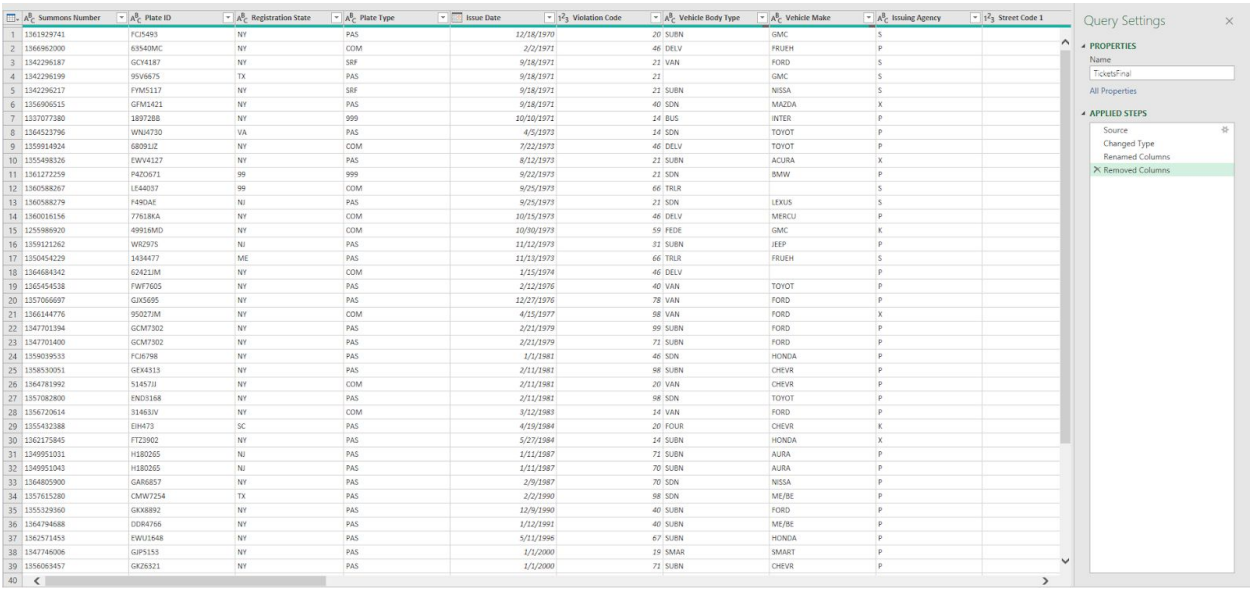
```
C:\Users\Jihyun>pscp -i C:/Users/Jihyun/Documents/jihyunssh.ppk Jihyun@34.67.78.107:TicketsFinal.csv .
TicketsFinal.csv          | 8864 kB | 805.8 kB/s | ETA: 00:46:42 |   0%
```

PART 6: Visualize using software

Import and Transform Data using Excel

-Rename the columns appropriately

-Remove columns will null values or no value



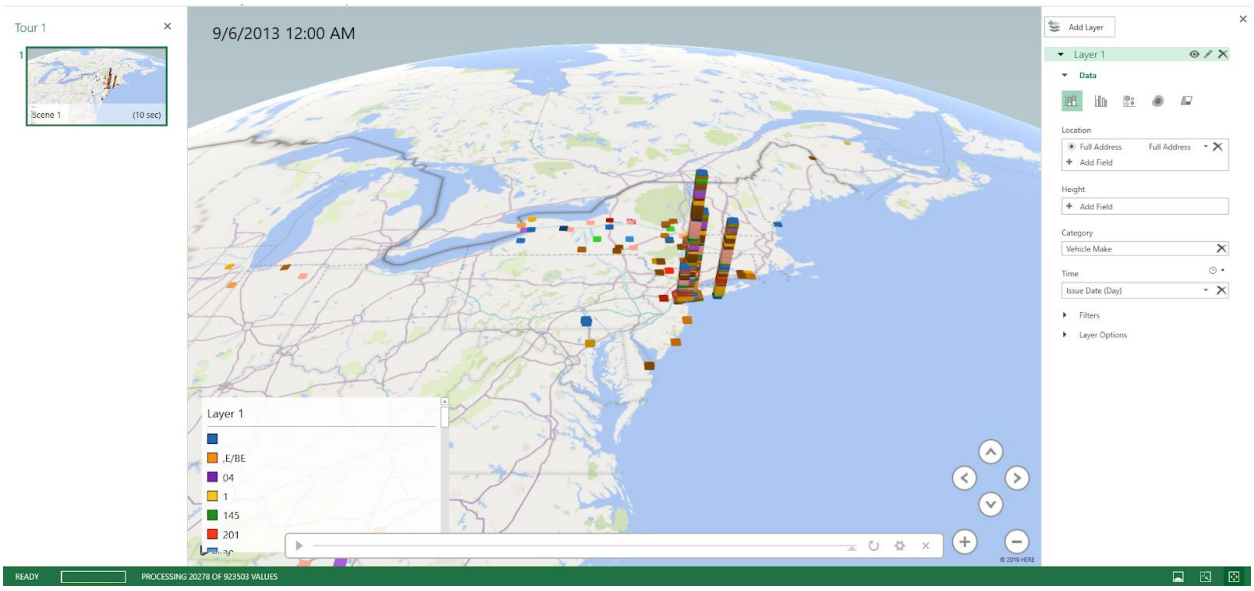
-Creating the visualization

Open up 3d Maps

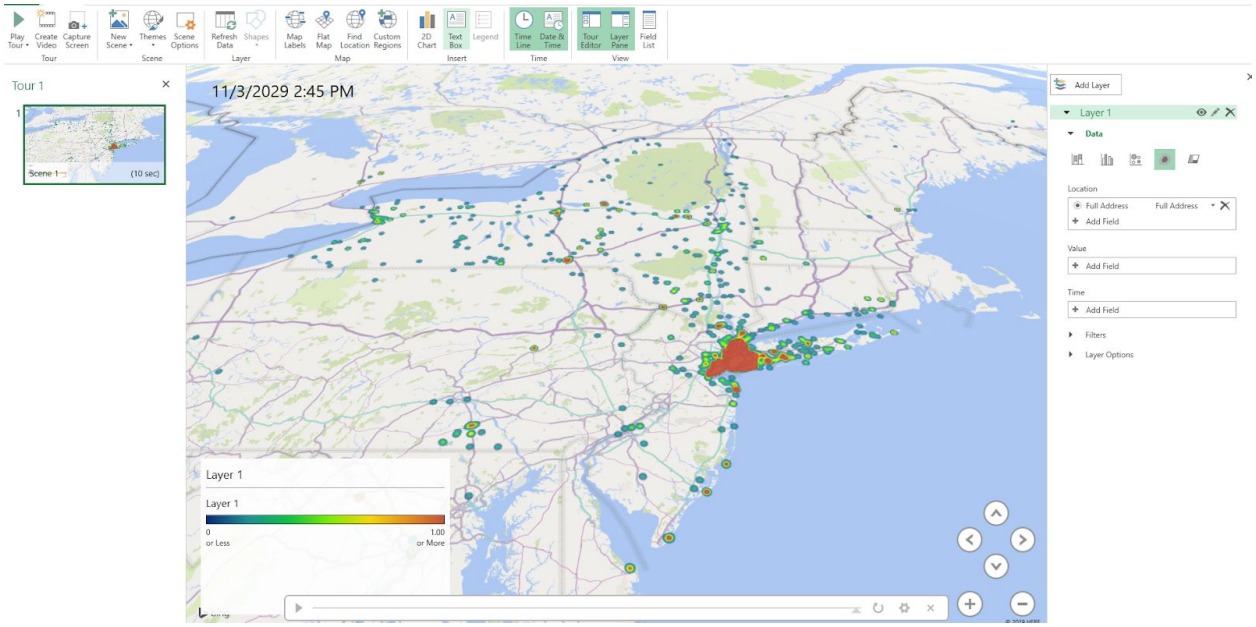
Set location to [full_address]

Set category to [vehicle_make]

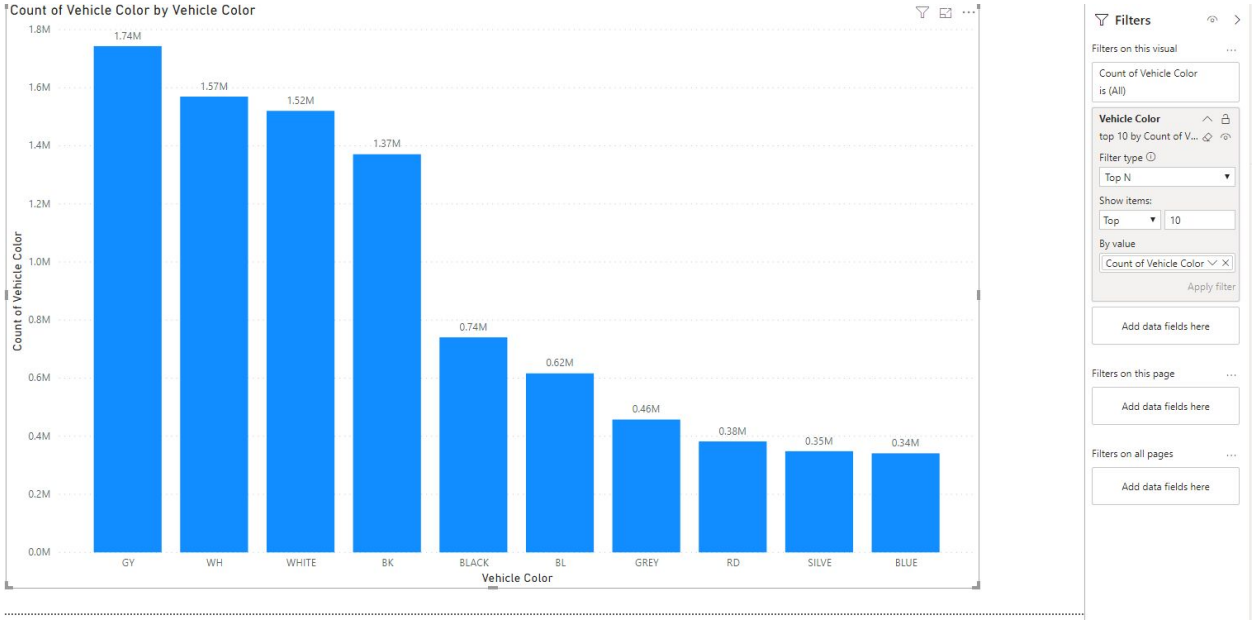
And time as issue_date (Day)



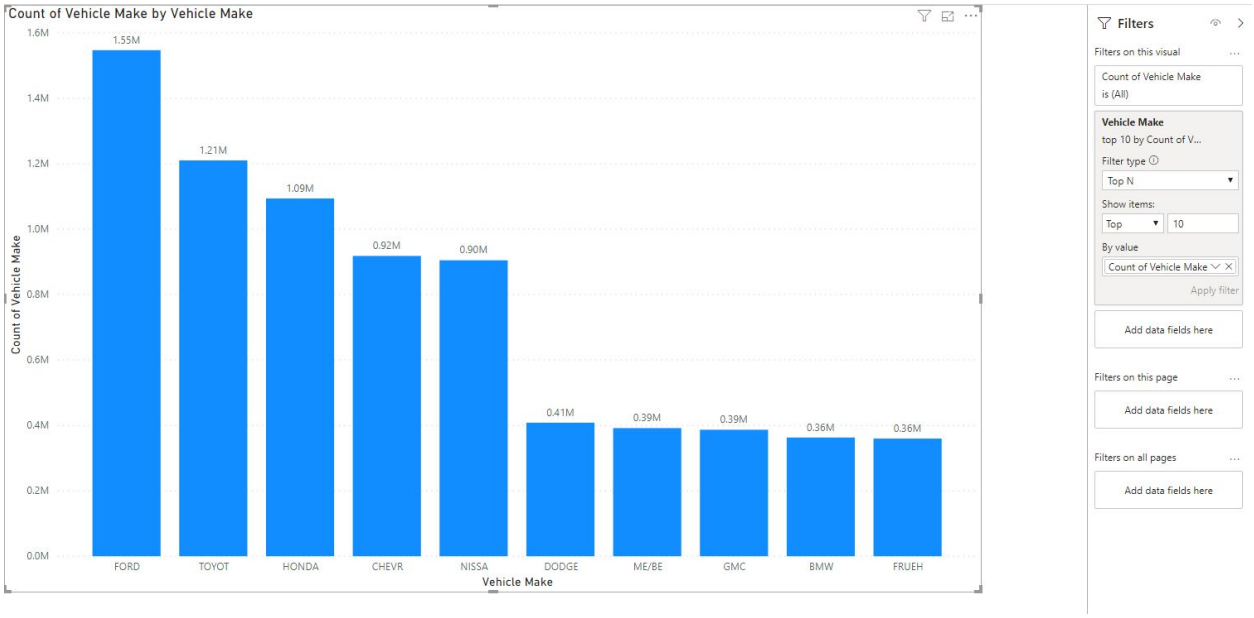
Change to a heat map to get a better idea of where tickets are being issued



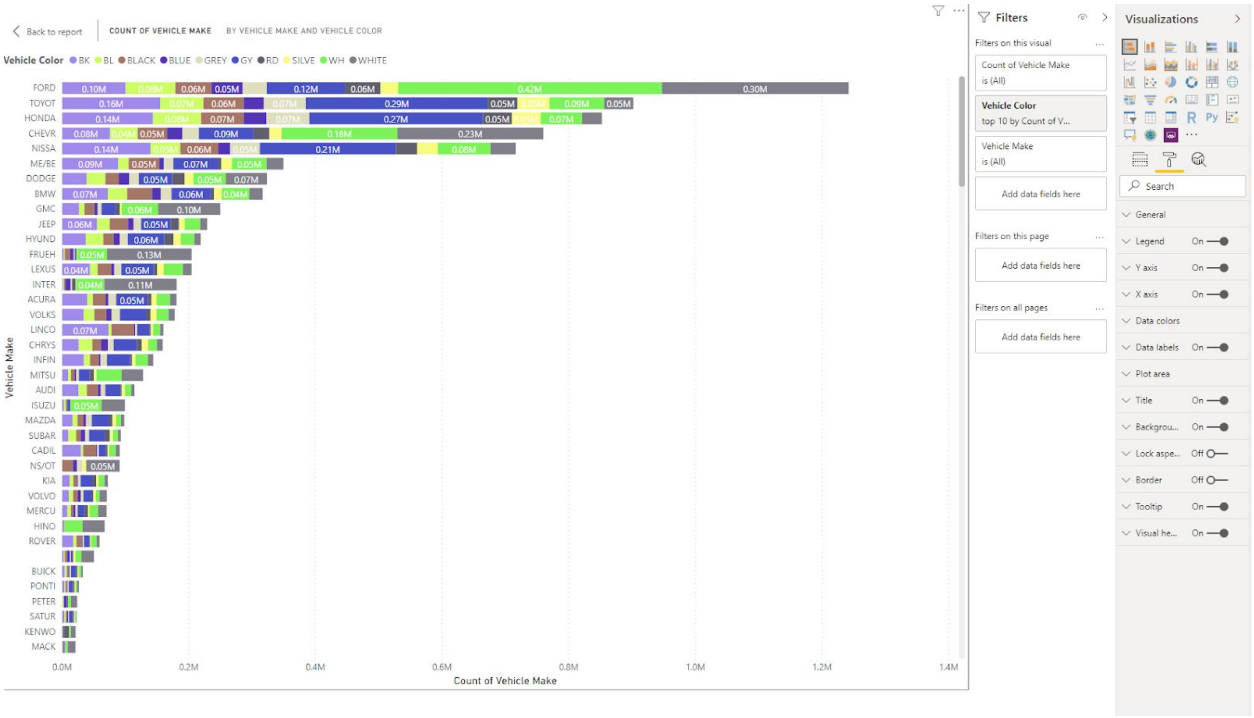
Using Power BI, we can use a bar chart and filter it to get a more in-depth view on the data



Set the x-axis as the vehicle color and the y-axis as the count for vehicle color.
Then set a filter with the vehicle color to only show the top 10 occurences.



Do the same as above but instead of vehicle color, use the vehicle make field.



Using both fields from the above graph, we can use vehicle color and vehicle make to determine which vehicle make had the most amount of occurences filtered with the color of the vehicle.

References

1. Data Source, https://data.cityofnewyork.us/City-Government/Parking-Violations-Issued-Fiscal-Year-2014/jt7v-77mi
2. Github, https://github.com/jhm916/new-york-ticket