

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

New Project

 You have 23 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)
[MANAGE QUOTAS](#)

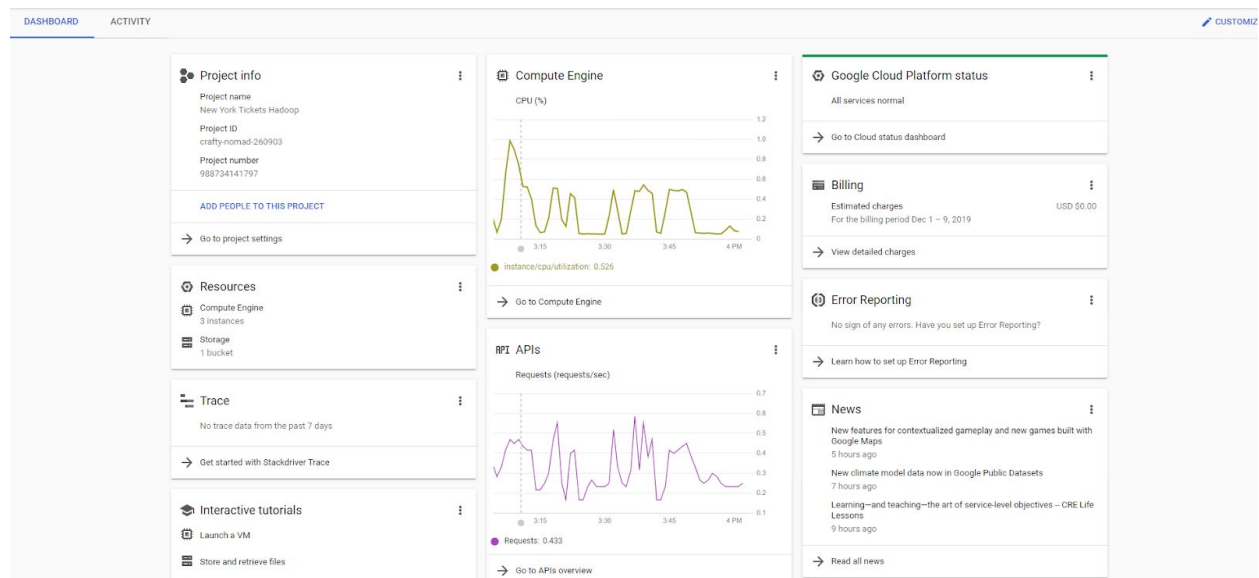
Project name *
New York Hadoop 

Project ID: new-york-hadoop. It cannot be changed later. [EDIT](#)

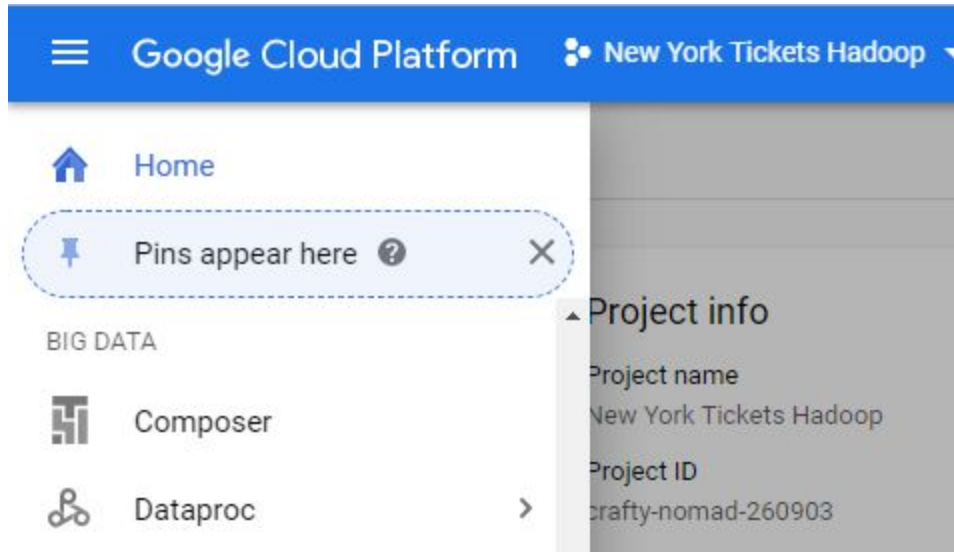
Location *
 No organization [BROWSE](#)
Parent organization or folder

[CREATE](#) [CANCEL](#)

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



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



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

Clusters + CREATE CLUSTER REFRESH DELETE REGIONS								
<input type="text" value="Search clusters, press Enter"/>								
<input type="checkbox"/> Name	Region	Zone	Total worker nodes	Scheduled deletion	Cloud Storage staging bucket	Created	Status	
<input checked="" type="checkbox"/> cluster-7701	us-central1	us-central1-b	2	Off	dataproc-8c4fd8c-d0e2-4f5a-b9dd-96c496310cd4-us-central1	Dec 5, 2019, 2:44:14 AM	Running	

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

Dataproc

← Create a cluster

N1

Powered by Intel Skylake CPU platform or one of its predecessors

Machine type

n1-standard-4 (4 vCPU, 15 GB memory)

vCPU

4

Memory

15 GB

CPU platform and GPU

Primary disk size (minimum 15 GB)

500 GB

Primary disk type

Standard persistent disk

Nodes (minimum 2)

2

Local SSDs (0-8)

0 x 375 GB

YARN cores

8

YARN memory

24 GB

Autoscaling policy (Optional)

☐ Enable autoscaling on the cluster.

This project does not currently have any applicable policy to enable autoscaling in this region. [Learn how to create autoscaling policy.](#)

Component gateway

☐ Enable access to the web interfaces of default and selected optional components on the cluster. [Learn more](#)

Advanced options

Create

Cancel

Equivalent REST or command line

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

Clusters							
<div>CREATE CLUSTER REFRESH DELETE REGIONS</div>							
<div>Search clusters, press Enter</div>							
<input type="checkbox"/> Name ^	Region	Zone	Total worker nodes	Scheduled deletion	Cloud Storage staging bucket	Created	Status
<input type="checkbox"/> cluster-7701	us-central1	us-central1-b	2	Off	dataproc-8c4fdf8c-d0e2-4f5a-b9dd-96c496310cd4-us-central1	Dec 5, 2019, 2:44:14 AM	Running

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

Compute Engine							
VM instances							
<div>CREATE INSTANCE IMPORT VM REFRESH START STOP RESET DELETE</div>							
<div>Filter VM instances Columns</div>							
<input type="checkbox"/> Name ^	Zone	Recommendation	In use by	Internal IP	External IP	Connect	
<input type="checkbox"/> cluster-7701-m	us-central1-b			10.128.0.2 (nic0)	34.67.78.107	SSH	
<input type="checkbox"/> cluster-7701-w-0	us-central1-b			10.128.0.3 (nic0)	23.251.156.63	SSH	
<input type="checkbox"/> cluster-7701-w-1	us-central1-b			10.128.0.4 (nic0)	104.154.100.248	SSH	

Open in browser window

Open in browser window on custom port

Open in browser window using provided private SSH key

View gcloud command

Use another SSH client

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>

- 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.

PutTY Key Generator ? X

File Key Conversions Help

Key

Public key for pasting into OpenSSH authorized_keys file:

```
HFv7dK5W4yAUejdksJ2JvLzaSYmX3DM6W9micEIV2mZs0RaLAITZ9JvGjXg62Ea1DjM58pmcUijPg90qtXeIW8p5Ly0DT6Fd3/NaUIT9txK8UvRXnk2+Z1/6U28ajyPqlp2d88E6L2dEDJeyLgtCAePkICY+uJDvnDLLypb8ePkYH1+CfVCDuTxXq3NXjB5SgiXY45a4slT48n8abBzIwEqWb3CqKi096XYUWRvrj/E3wHFQ== Jihyun
```

Key fingerprint: ssh-rsa 2048 43:ed:45:15:9f:7c:d2:13:75:f4:42f1:8c:b6:a2:b2

Key comment: Jihyun

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair Generate

Load an existing private key file Load

Save the generated key Save public key Save private key

Parameters

Type of key to generate:
☒ RSA ☐ DSA ☐ ECDSA ☐ Ed25519 ☐ SSH-1 (RSA)

Number of bits in a generated key: 2048

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

Google Cloud Platform New York Tickets Hadoop

Compute Engine Metadata

VM instances Instance groups Instance templates Sole-tenant nodes Disks Snapshots Images TPUs Committed use discounts

Metadata SSH Keys

CISKEY

```
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAAQEAgsbFMFTX3vkX8M53jTk3+G5Z0wBzy6F01C11S2Y13sgDVw32waBbU3BzJat3g5+7kJjcauqc/aSbWGRcwqoM05f/5c+gPhhcP12j3p1IBxgHFPHR14kuK+MiGyJBvuHpfAXwMFvH+0LeBhRjL1+d40XxPmLAZR8WHz6V9dcXYgS0gEe2qcypmPqWwNi1S81Vop7GTRJJD+V8CAG/LTbj1bIVFL3H44B/d843I211XIbT/ZtBU5f3XtRfaVaGw6UfCe8uPPH9jrQ1FedEDrofo3VsveSfy7Yx0pSJkcCnCG+vCS/vkGfK/Qn7rei95AfCAH8K+0J9MhmnLFFnV+w== CISKEY
```

Jihyun

```
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAAQEAmmjouXuwttgXpxz5eC8LoU4+No21ZKmTptDHHwJ5Vq0ztZc5K1MJzVL4MumiT7DCHtYy506aeB815XRw811L605b00EdzrFkEcSwHFv7dK5W4yAUejdksJ2JvLzaSYmX3DM6W9micE1V2rnZs0RaLAITZ9JvGjXg62Ea1DjM58pmcUijPg90qtXeIW8p5Ly0DT6Fd3/NaUIT9txK8UvRXnk2+Z1/6U28ajyPqlp2d88E6L2dEDJeyLgtCAePkICY+uJDvnDLLypb8ePkYH1+CfVCDuTxXq3NXjB5SgiXY45a4slT48n8abBzIwEqWb3CqKi096XYUWRvrj/E3wHFQ== Jihyun
```

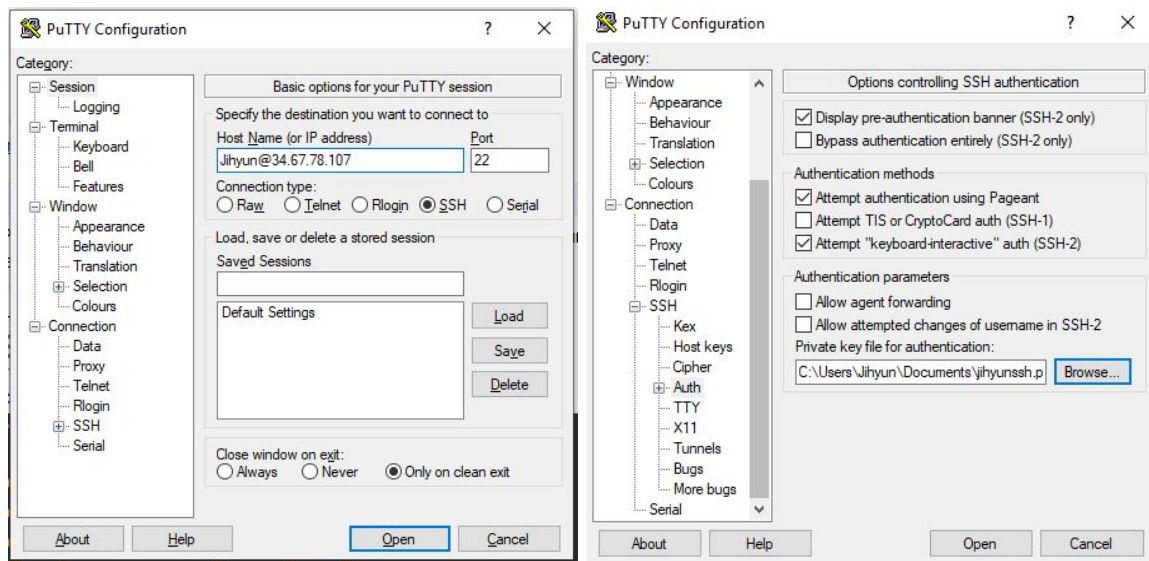
PART 3: Connect to Master Node using PUTTY

Download and open up putty.exe from

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
✓ cluster-7701-m	us-central1-b			10.128.0.2 (nic0)	34.67.78.107	SSH

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]

```
Jihyun@cluster-7701-m: ~  
Using username "Jihyun".  
Authenticating with public key "Jihyun"  
Linux cluster-7701-m 4.9.0-11-amd64 #1 SMP Debian 4.9.189-3+deb9u2 (2019-11-11)  
x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Fri Dec 13 10:23:56 2019 from 76.175.51.85  
Jihyun@cluster-7701-m:~$
```


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.csv
```

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 Jihyun
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 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
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 from 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/000000_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 with null values or no value

	Summons Number	Plate ID	Registration State	Plate Type	Issue Date	Violation Code	Vehicle Body Type	Vehicle Make	Issuing Agency	Street Code 1
1	1361929741	FCZ483	NY	PAS	12/7/1970	20 SUBN	GMC	S		
2	136660200	63540MC	NY	COM	2/2/1971	46 DELV	FRUEH	P		
3	1342298187	GCY4187	NY	SBP	9/18/1971	21 VAN	FORD	S		
4	1342298199	9518675	TX	PAS	9/18/1971	21	GMC	S		
5	1342298217	FYN5117	NY	SBP	9/18/1971	21 SUBN	NISSA	S		
6	135609015	GFN1421	NY	PAS	9/18/1971	40 SON	MAZDA	X		
7	135707380	1897285	NY	999	10/10/1972	34 BUS	INTER	P		
8	1364522796	WNU4730	VA	PAS	4/5/1992	14 SON	TOYOT	P		
9	1359914924	6809112	NY	COM	7/22/1973	46 DELV	TOYOT	P		
10	1355498326	EWV4127	NY	PAS	8/72/1973	21 SUBN	ACURA	X		
11	1361272259	P420X71	99	999	9/22/1973	21 SON	BMW	P		
12	1360588287	LE44037	99	COM	9/25/1973	66 TRLR	S	S		
13	1360588279	F48046	NI	PAS	9/25/1973	21 SON	LEPUS	S		
14	1360018156	776018A	NY	COM	10/15/1973	46 DELV	MERCU	P		
15	1355886920	49918MD	NY	COM	10/10/1978	59 PRIDE	GMC	K		
16	1359121262	WU2975	NI	PAS	11/21/1973	31 SUBN	JEOP	P		
17	1350454229	1434477	ME	PAS	11/21/1973	66 TRLR	FRUEH	S		
18	1364684342	62442LM	NY	COM	1/15/1974	46 DELV		P		
19	1365481538	FW77605	NY	PAS	2/12/1976	40 VAN	TOYOT	P		
20	1357066697	GIX6695	NY	PAS	12/2/1976	78 VAN	FORD	P		
21	1363144776	9502794	NY	COM	4/12/1977	38 VAN	FORD	X		
22	1347713194	GCMT302	NY	PAS	2/12/1979	99 SUBN	FORD	P		
23	1347710400	GCMT302	NY	PAS	2/12/1979	71 SUBN	FORD	P		
24	1358019533	FCB798	NY	PAS	1/12/1981	46 SON	HONDA	P		
25	1358530051	GEK4313	NY	PAS	2/11/1981	88 SUBN	CHEVR	P		
26	1364781992	5145711	NY	COM	2/11/1981	20 VAN	CHEVR	P		
27	1357028600	END3168	NY	PAS	2/11/1981	88 SON	TOYOT	P		
28	1356720614	31483V	NY	COM	3/12/1983	14 VAN	FORD	P		
29	135543388	DM479	SC	CHEVR	4/12/1984	20 FOUR	CHEVR	K		
30	1362175815	FTZ2902	NY	PAS	5/27/1984	14 SUBN	HONDA	X		
31	1349951011	H180205	NI	PAS	1/11/1987	71 SUBN	AURA	P		
32	1349951043	H180205	NI	PAS	1/11/1987	70 SUBN	AURA	P		
33	1364805900	GAB8857	NY	PAS	2/9/1987	70 SON	NISSA	P		
34	1357615280	CMW7254	TX	PAS	2/2/1990	98 SON	ME/BE	P		
35	1355329560	GKX8892	NY	PAS	12/9/1990	40 SUBN	FORD	P		
36	1364794888	DDH4766	NY	PAS	1/12/1991	40 SUBN	ME/BE	P		
37	1362571853	FNU1648	NY	PAS	5/11/1996	87 SUBN	HONDA	P		
38	1367740006	GPY1513	NY	PAS	1/12/2000	19 SMAR	SMART	P		
39	1364063457	GK28321	NY	PAS	1/12/2000	71 SUBN	CHEVR	P		

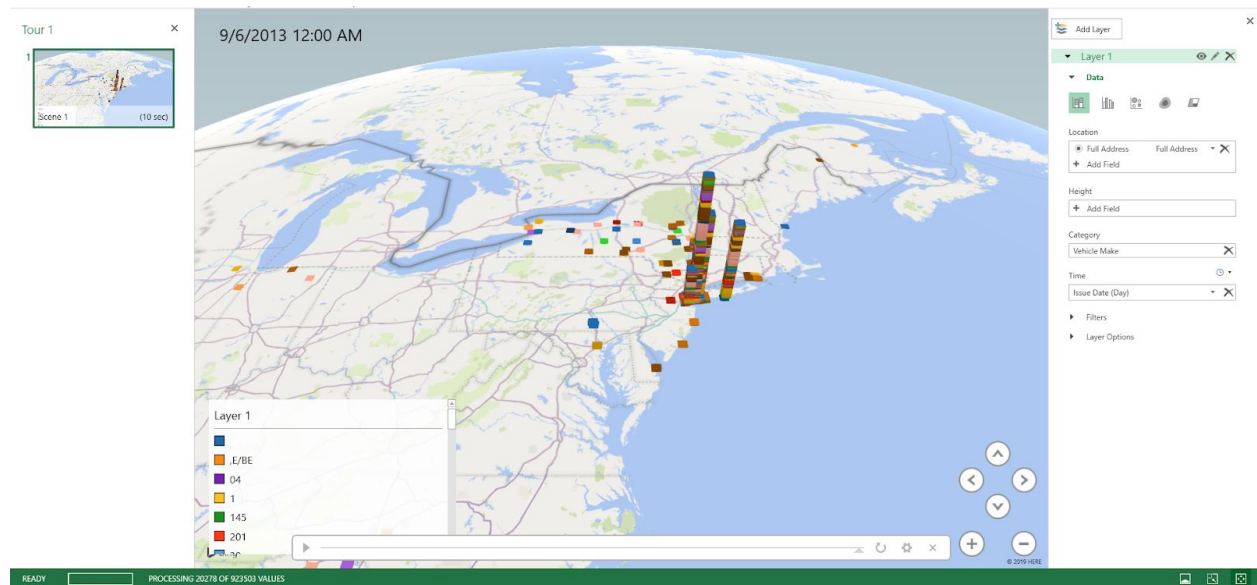
-Creating the visualization

Open up 3d Maps

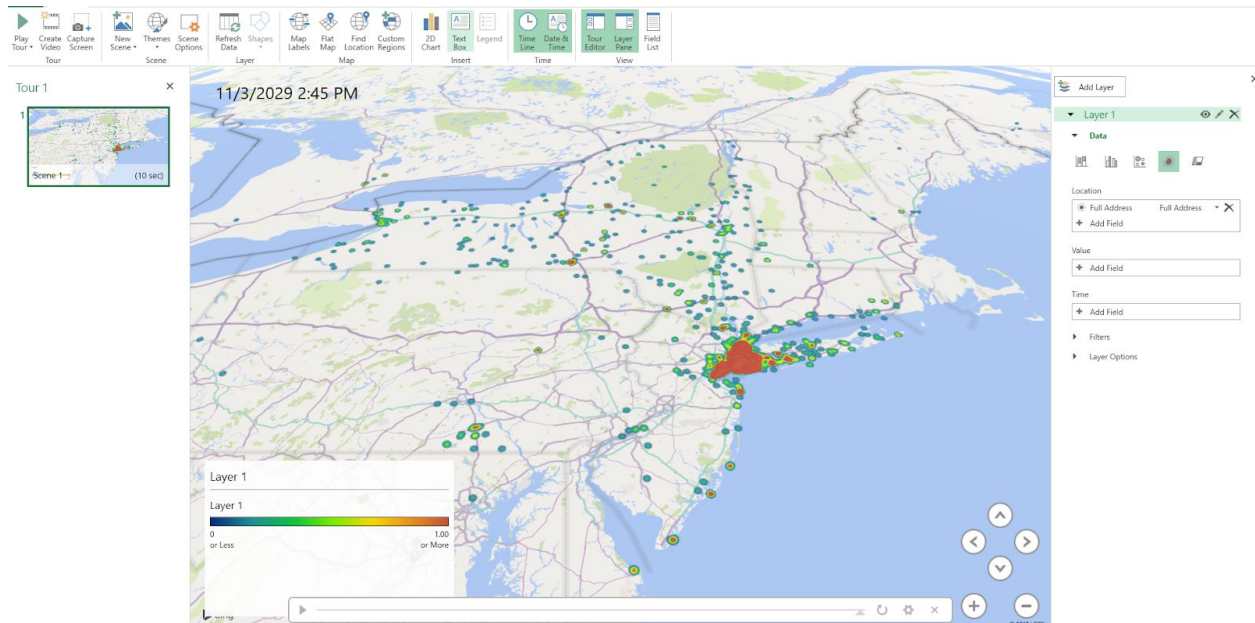
Set location to [full_address]

Set category to [vehicle_make]

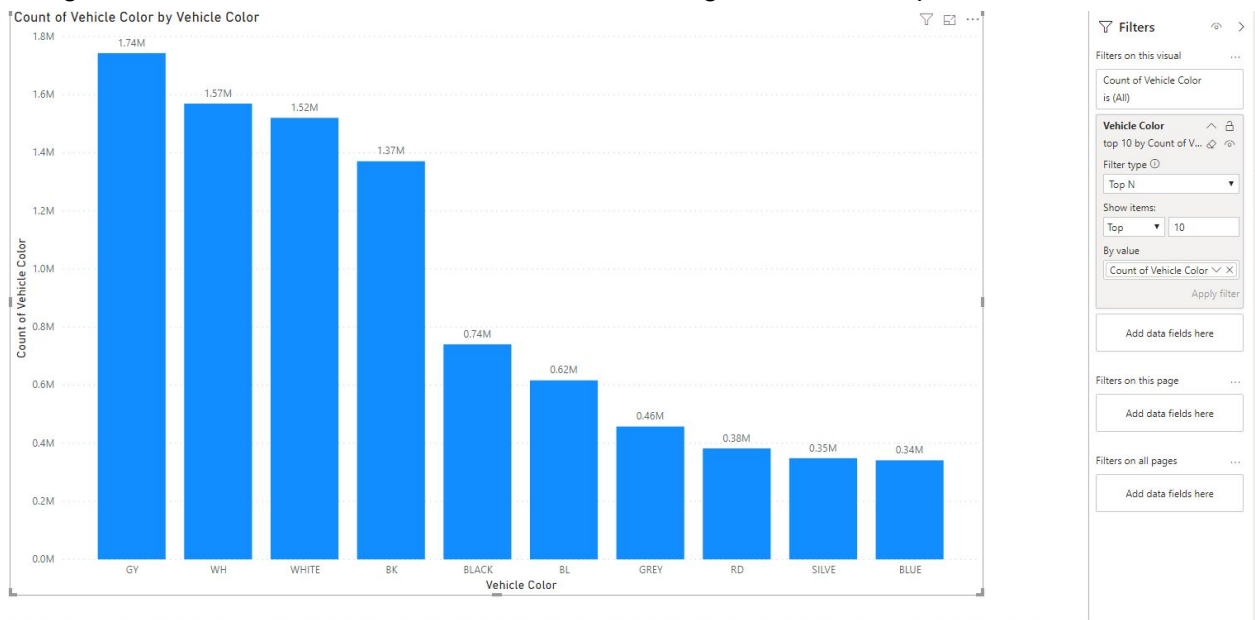
And time as issue_date (Day)



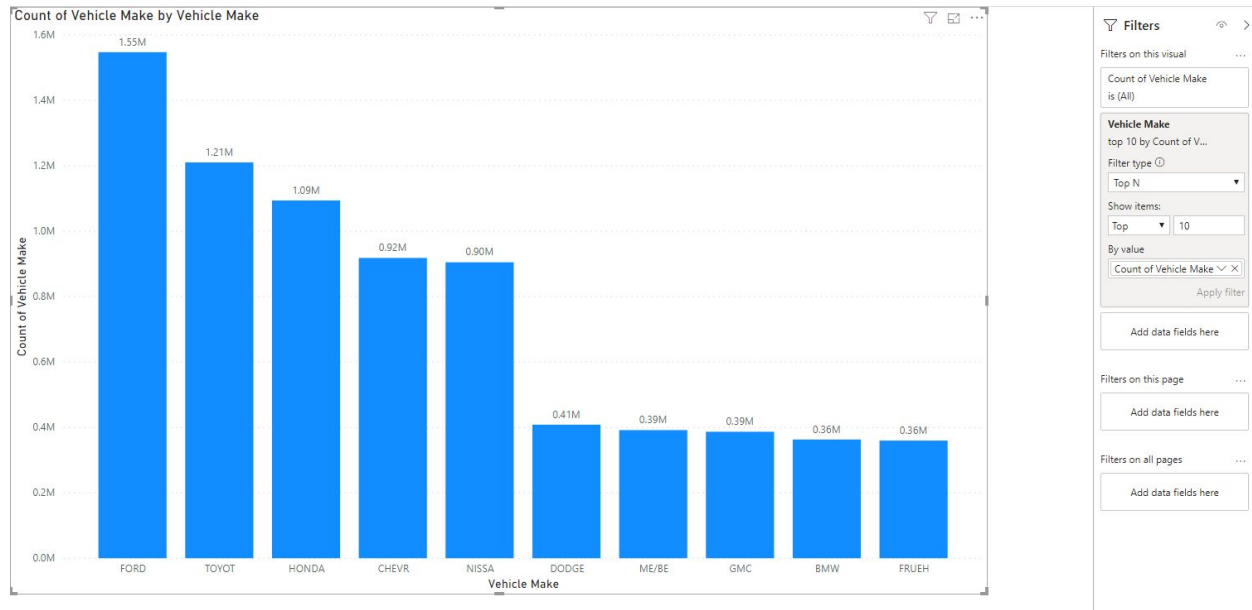
Change to a heat map to get a better idea of where tickets are being issued



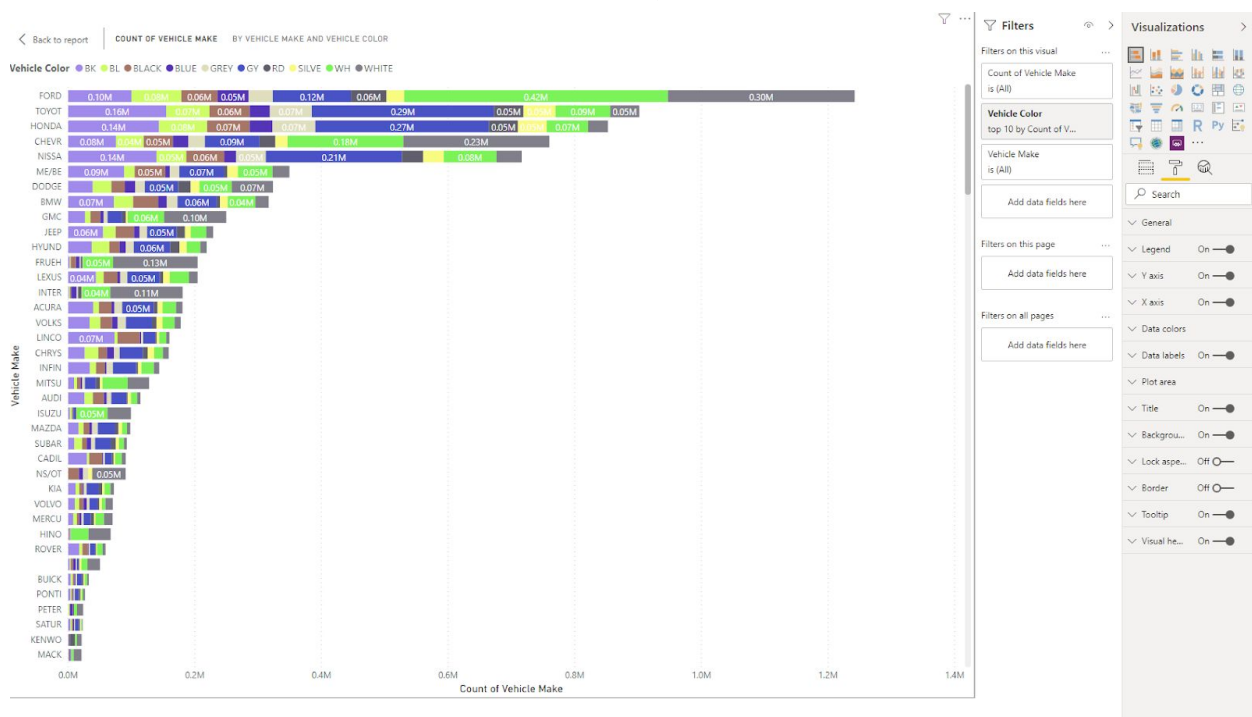
Using Power BI, we can use a bar chart and filter it to get a more in-depth view on the data



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 occurrences.



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 occurrences 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>