Geo-Location Clustering Using K-Means Algorithm

Introduction:

Clustering, as defined by Stanford University, is a process of grouping and Set of data points into clusters such that points within the same cluster are positioned Comparable to each other while points from various clusters are different.

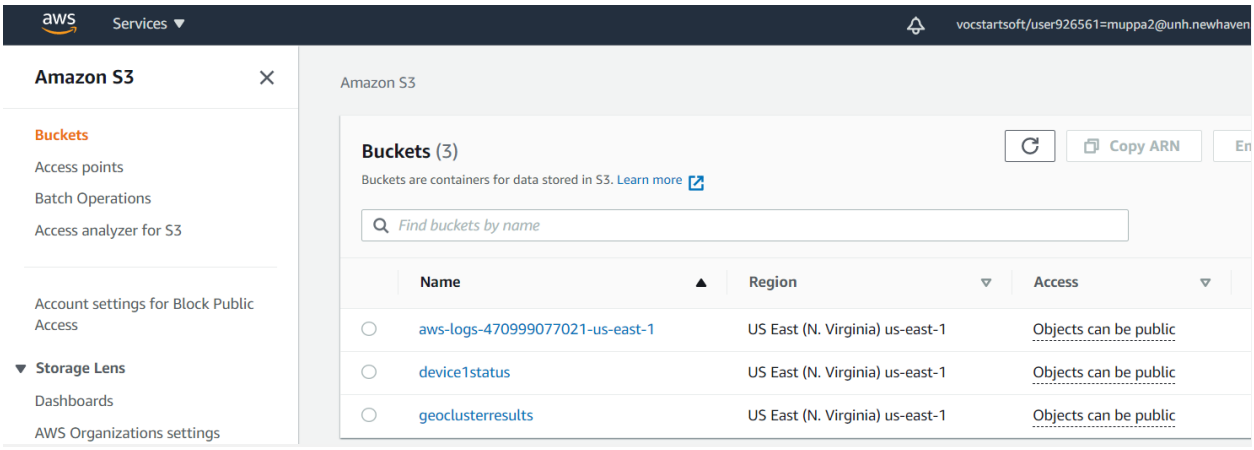
Clustering has many useful applications such as finding a group of consumers with common preferences, grouping documents based on the similarity of their contents, or finding spatial clusters of customers to improve logistics. More specific use cases are

- Marketing: given a large set of customer transactions, find customers with similar purchasing behaviors .
- Document classification: cluster web log data to discover groups of similar access patterns.
- Logistics: find the best locations for warehouses or shipping centers to minimize shipping times.

We will approach the clustering problem by implementing the k-means algorithm. k- means is a distance-based method that iteratively updates the location of k cluster centroids until convergence. The main user-defined ingredients of the k-means algorithm are the distance function (often Euclidean distance) and the number of clusters k. This parameter needs to be set according to the application or problem domain. (There is no magic formula to set k.) In a nutshell, k-means groups the data by minimizing the sum of squared distances between the data points and their respective closest centroid.

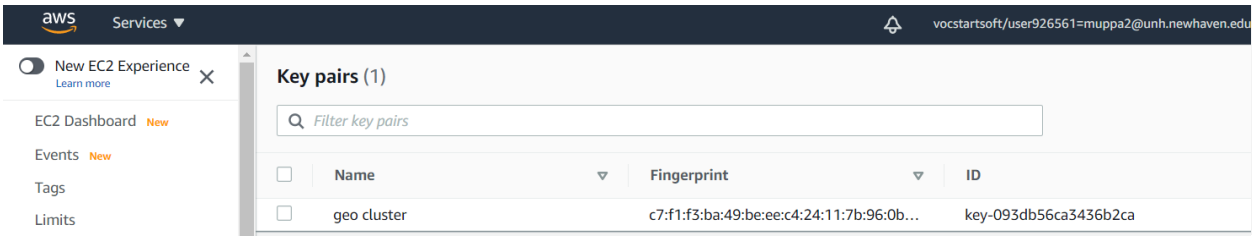
System Configuration:

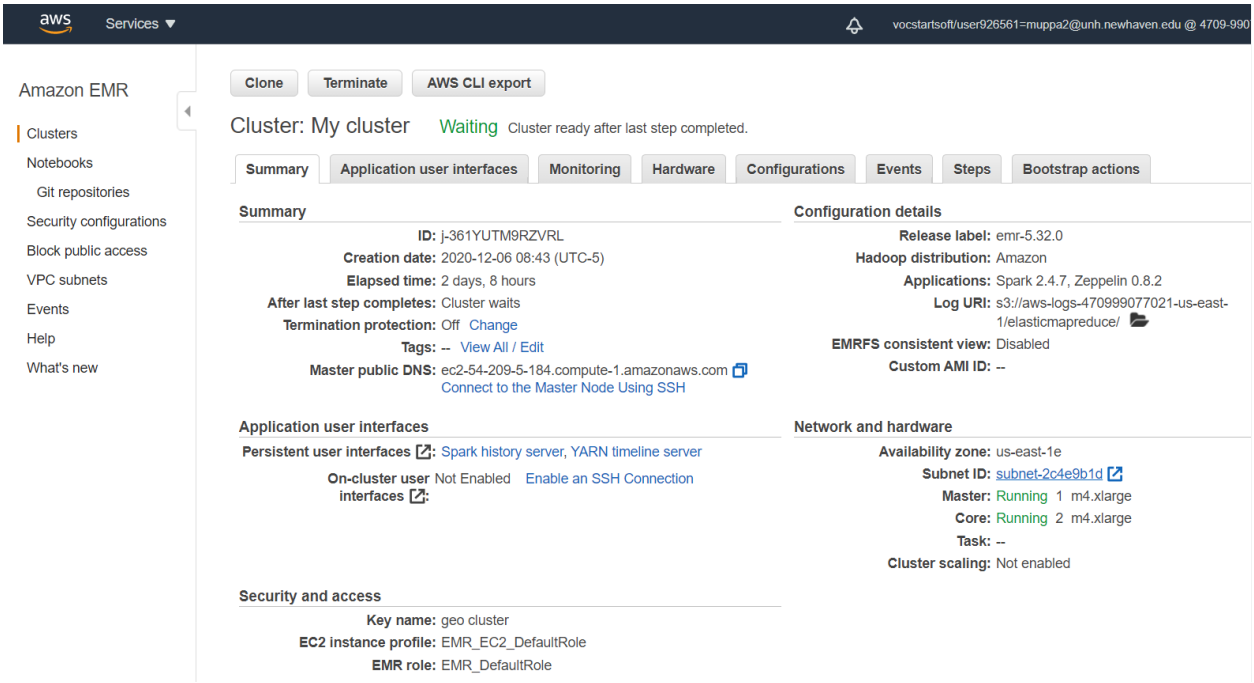
S3 Bucket Creation:



Creating EMR Cluster:

In the process of creating EMR cluster, created a key-pair, which then downloaded as .pem file. Using Putty SSH client created .ppk file from .pem file.





Data Preparation :

Data Preparation Before implementing the actual algorithm, we went through pre-processing step in order to convert the data into a standardized format for later processing. The following describes the pre-processing process for device status data:

1. Load the dataset
2. Determine which delimiter to use
3. Filter out any records which do not parse correctly; each record should have exactly 14 values
4. Extract the date, model, device ID, and latitude and longitude a. date: 1st field b. model: 2nd field c. device ID: 3rd field d. latitude: 13th field e. longitude: 14th field.
5. Store latitude and longitude as the first two fields
6. Filter out locations that have a latitude and longitude of 0
7. Split the model field that contains the device manufacturer and model name by spaces
8. Saved the extracted data to comma delimited text files in S3.

Location: 's3://geoclusterresults/devicedata.csv'

Data:

Mobilenet:

In [23]:
```
#Printing the data frame after droping zeros
devicedata
```

Out[23]:

	latitude	longitude	date	model	device ID
0	33.6894754264	-117.543308253	2014-03-15:10:10:20	F41L	8cc3b47e-bd01-4482-b500-28f2342679af
1	39.3635186767	-119.400334708	2014-03-15:10:10:20	F41L	707daba1-5640-4d60-a6d9-1d6fa0645be0
2	33.1913581092	-116.448242643	2014-03-15:10:10:20	Novelty Note 1	db66fe81-aa55-43b4-9418-fc6e7a00f891
3	33.8343543748	-117.330000857	2014-03-15:10:10:20	F41L	ffa18088-69a0-433e-84b8-006b2b9cc1d0
4	37.3803954321	-121.840756755	2014-03-15:10:10:20	F33L	66d678e6-9c87-48d2-a415-8d5035e54a23
...	...	...	...	...	...
65640	39.4463417571	-114.736213453	2014-03-15:10:49:30	F22L	40e61459-5448-4dc9-bb89-42e73a4e19cf
65641	38.4282665514	-121.25933863	2014-03-15:10:49:30	S2	b13ece99-62ab-4c9f-a366-6a06bd5e877f
65642	33.7778202246	-108.575470704	2014-03-15:10:49:30	F41L	32af1a0b-ca7f-4906-9772-9eb9435e7e4c
65643	38.2596913494	-122.295712621	2014-03-15:10:49:30	S1	a48a5559-d916-481b-84a9-5dce6272cce1
65644	34.2415255221	-118.23526739	2014-03-15:10:49:30	2	d86fbaa6-b71b-435f-a0bf-5304a202a70b

65645 rows × 5 columns

Synthetic:

In [38]:
```
#Displaying Sample_geo data
geo_2.head(5)
```

Out[38]:
```
[Row(Latitude='37.77253945', Longitude='-77.49954987', LocationID='1'),
 Row(Latitude='42.09013298', Longitude='-87.68915558', LocationID='2'),
 Row(Latitude='39.56341754', Longitude='-75.58753204', LocationID='3'),
 Row(Latitude='39.45302347', Longitude='-87.69374084', LocationID='4'),
 Row(Latitude='38.9537989', Longitude='-77.01656342', LocationID='5')]
```

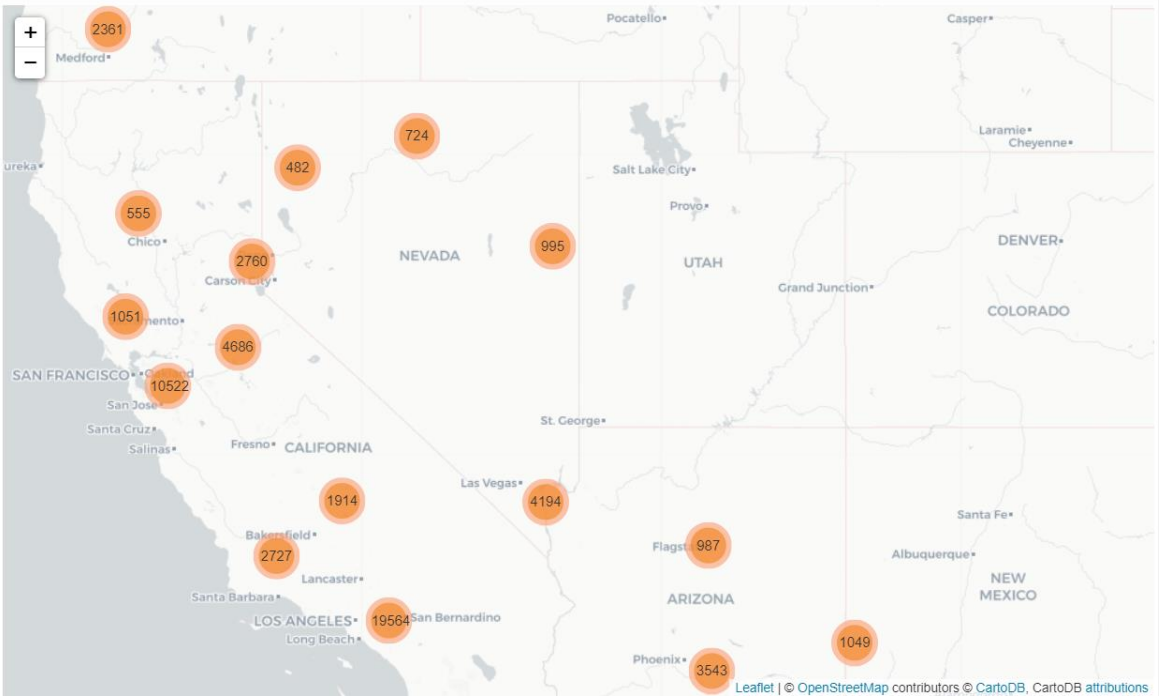
DBpedia:

	lat	long	name_of_page
0	36.7	3.216666666666667	<http://dbpedia.org/resource/Algeria>
1	42.5	1.5166666666666666	<http://dbpedia.org/resource/Andorra>
2	12.516666666666667	-70.03333333333333	<http://dbpedia.org/resource/Aruba>
3	-8.833333333333334	13.333333333333334	<http://dbpedia.org/resource/Angola>
4	41.333333333333336	19.8	<http://dbpedia.org/resource/Albania>

Data Visualization:

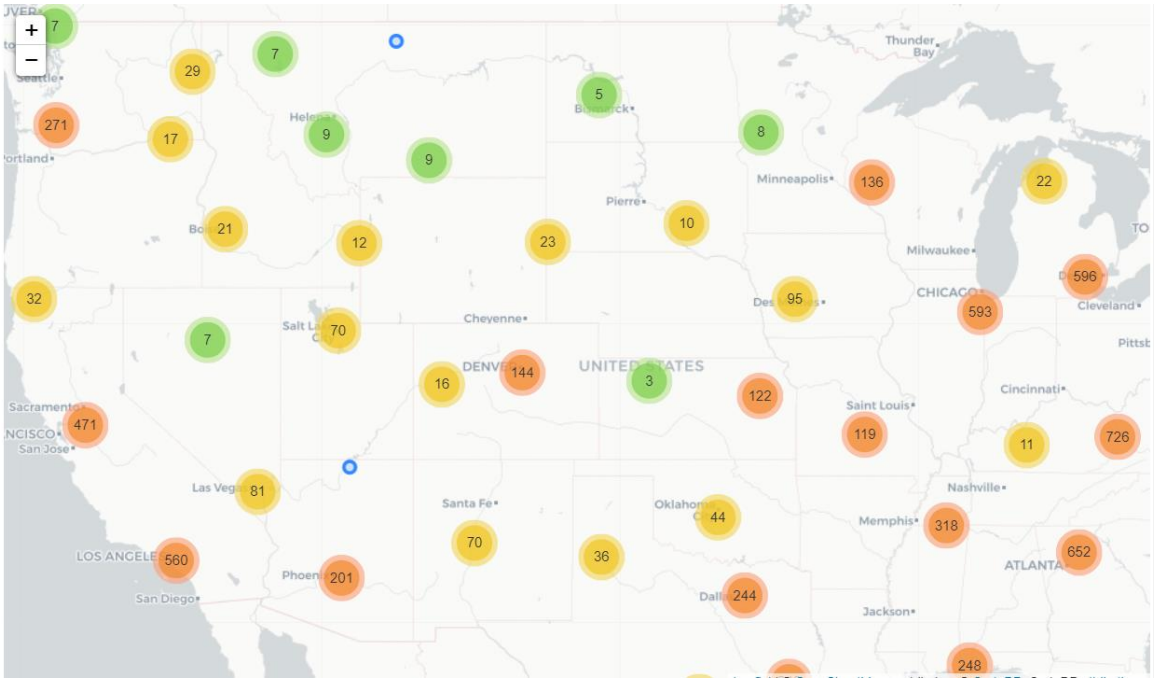
Mobilenet:

Out[32]:



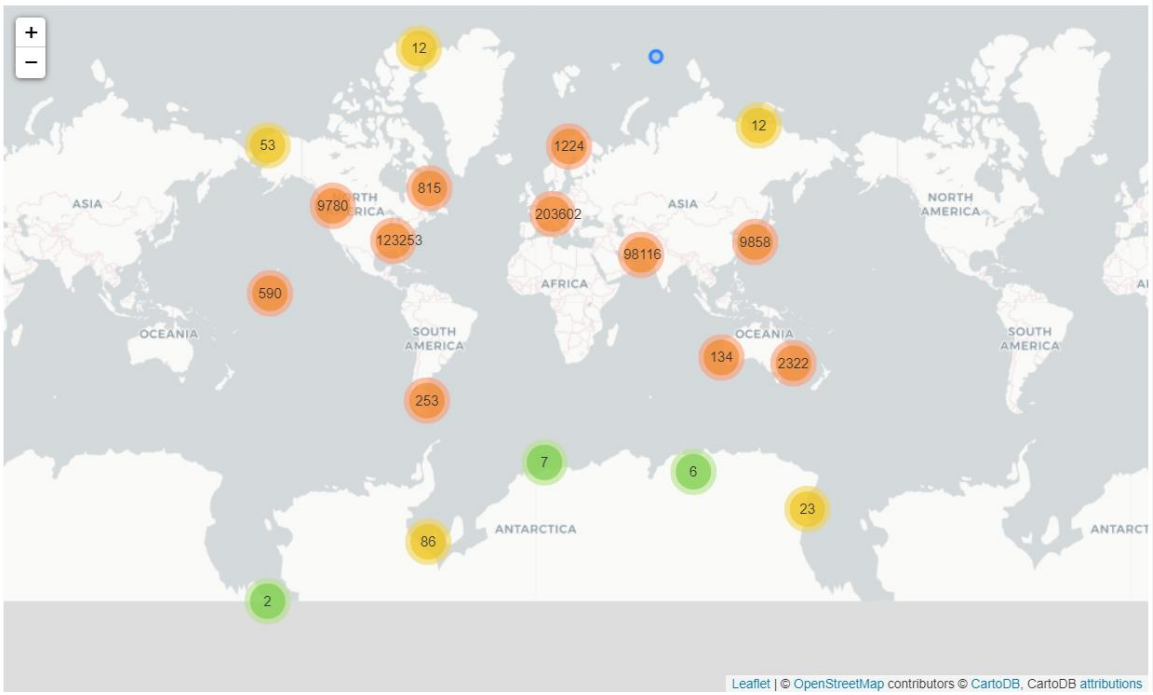
Synthetic:

Out[44]:



DBpedia:

Out[13]:


+
−
12
53
1224
12
815
9780
203602
ASIA
ASIA
NORTH AMERICA
123253
9858
98116
590
AFRICA
OCEANIA
SOUTH AMERICA
OCEANIA
SOUTH AMERICA
134
2322
253
7
6
23
ANTARCTICA
86
ANTARCT
2
Leaflet | © OpenStreetMap contributors © CartoDB, CartoDB attributions