Learning and Knowledge Management

PROCESSING BIGDATA
On
AWS-EMR
Lab Guide

Developed & Tested

By

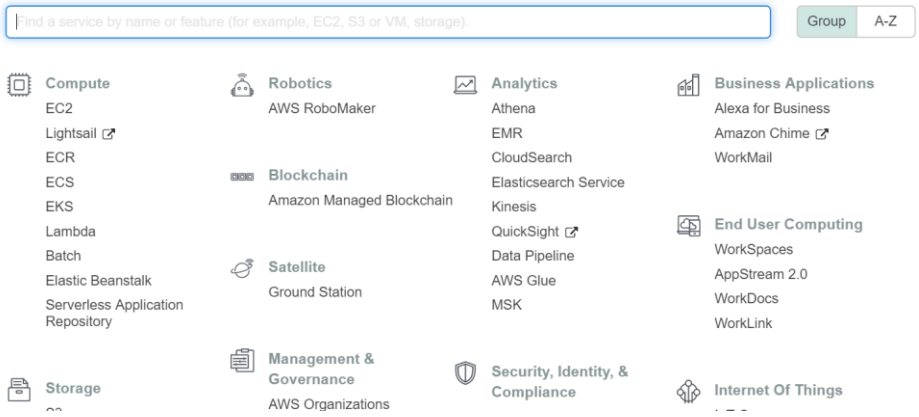
Karthigayen.Y

LKM, Accenture - ATCI

Creating EMR Cluster

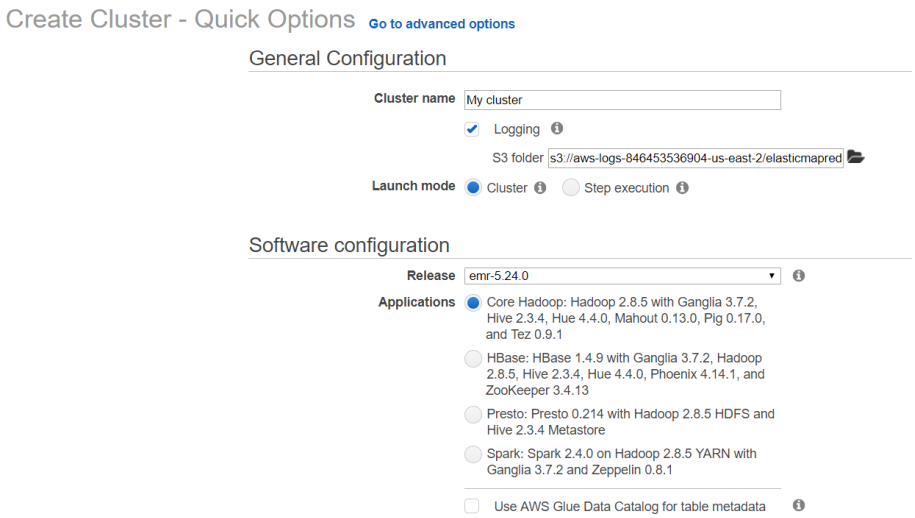
Steps to create an EMR cluster.

Step 1 : From the services menu, under Analytics services, select EMR as shown below.

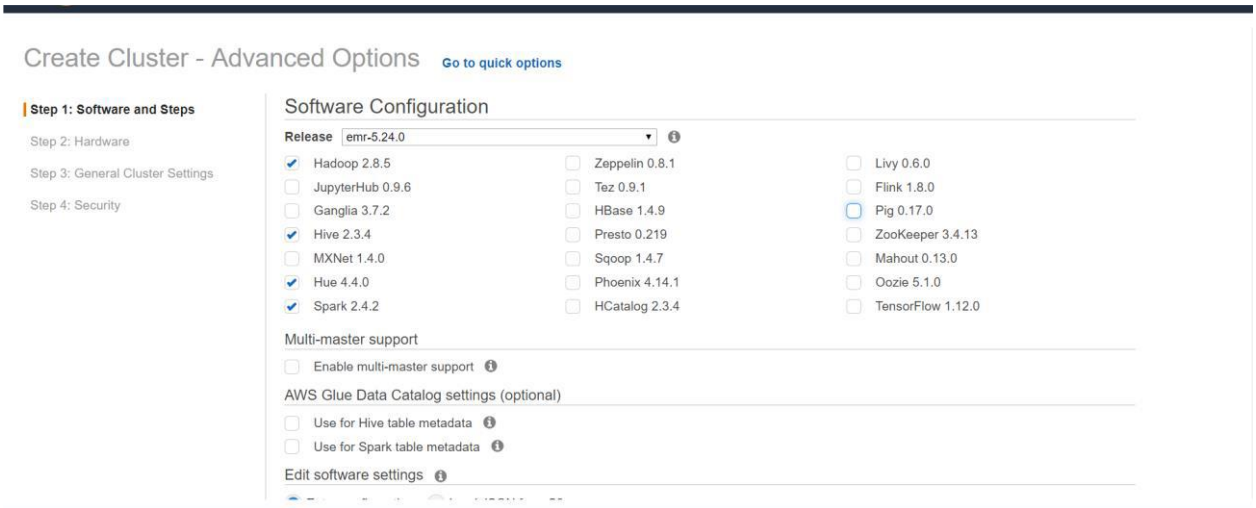


Step 2 : Click on Create Cluster button as shown below.

Step 3 : On the Create cluster – Quick option, click on Go to advanced options.

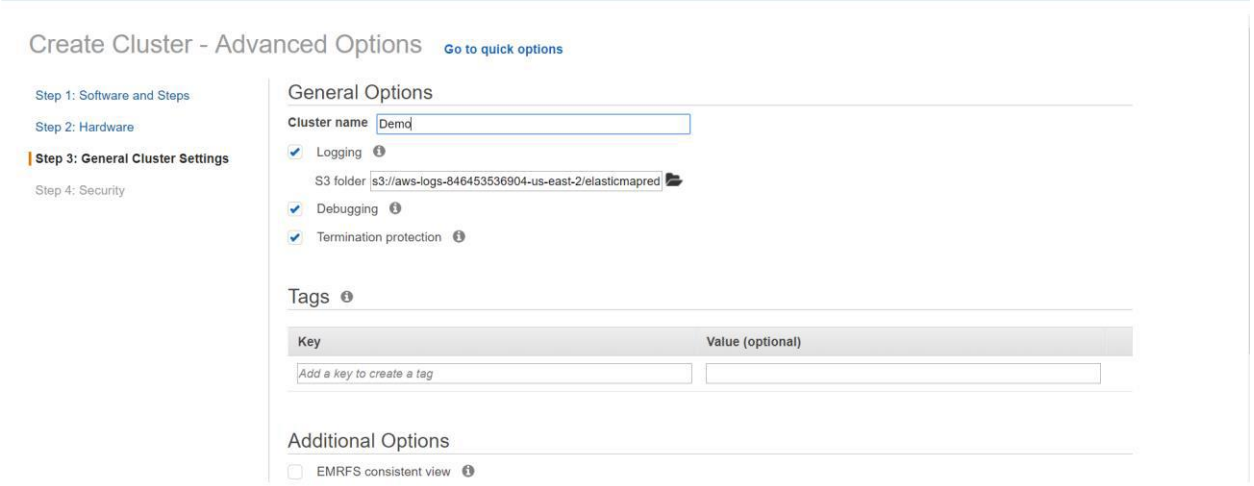


Step 4 : On the Create Cluster – Advanced Options, select Hadoop, Hive, Hue, and Spark components and click on Next button.

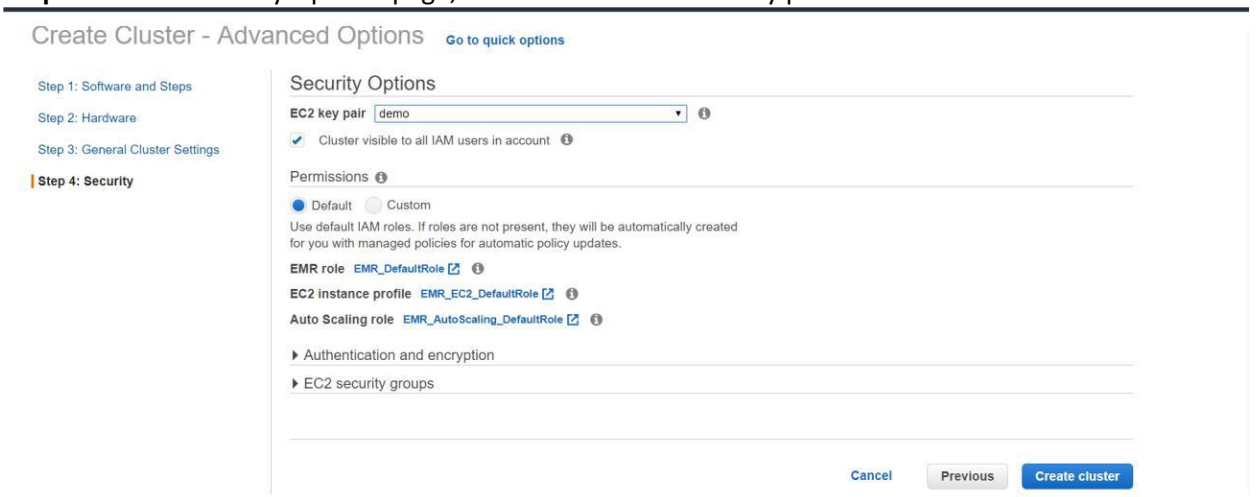


Step 5 : On the Hardware Configuration page, click on Next button.

Step 6 : On the General Options page, specify the cluster name as "Demo" as shown below and click on "Next".

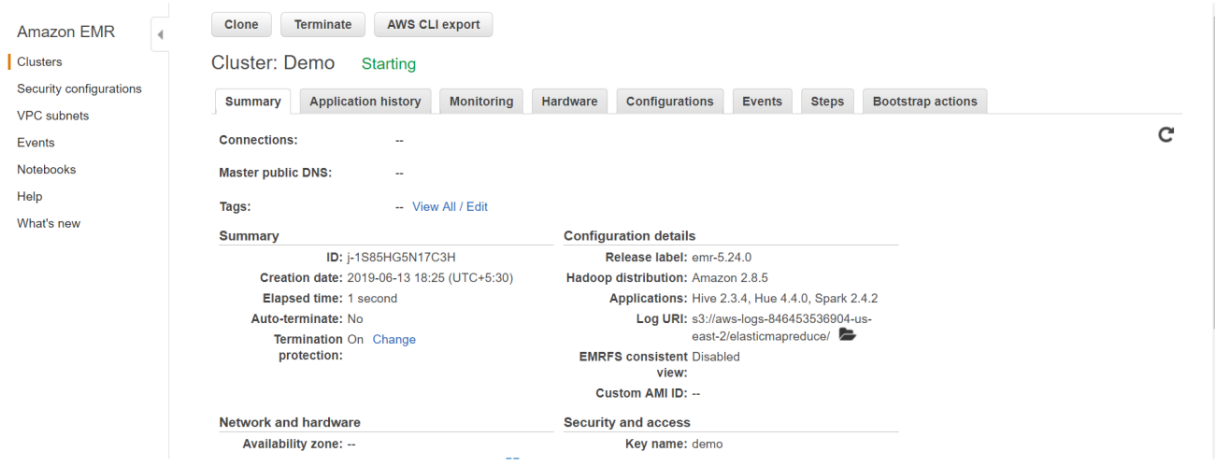


Step 7 : On the Security Options page, select the "demo" EC2 key pair and click on Create Cluster button.



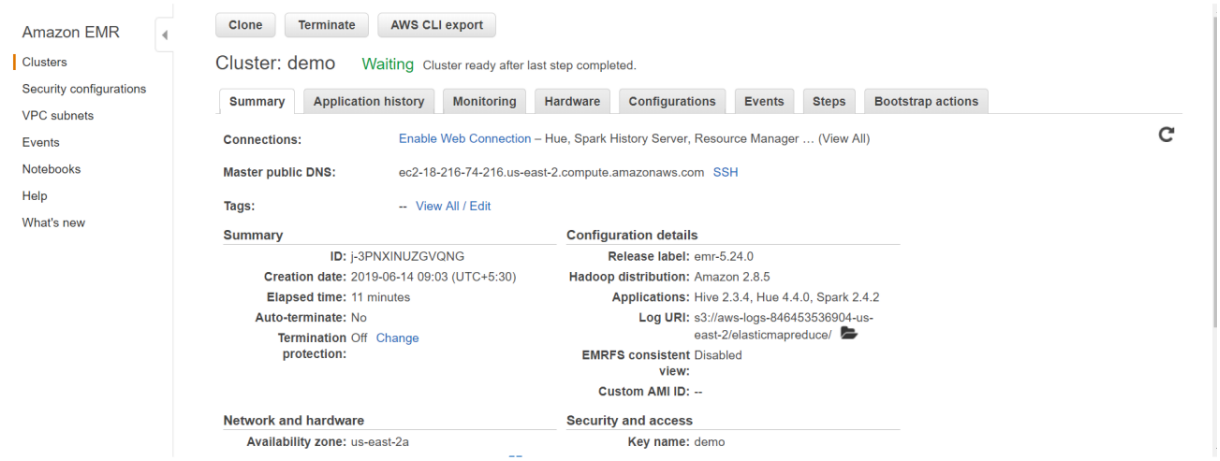
Note: If you don't have key pair, create a new key pair – **Services->EC2->KeyPair**. (Now we can directly create a **.PPK** file on the console which automatically gets downloaded to the Windows host machine as well.)

Step 8 : Now, you can see the Cluster: Demo status as "Starting" as shown below.



Step 9 : The cluster creation process will take around 10 to 15 minutes.

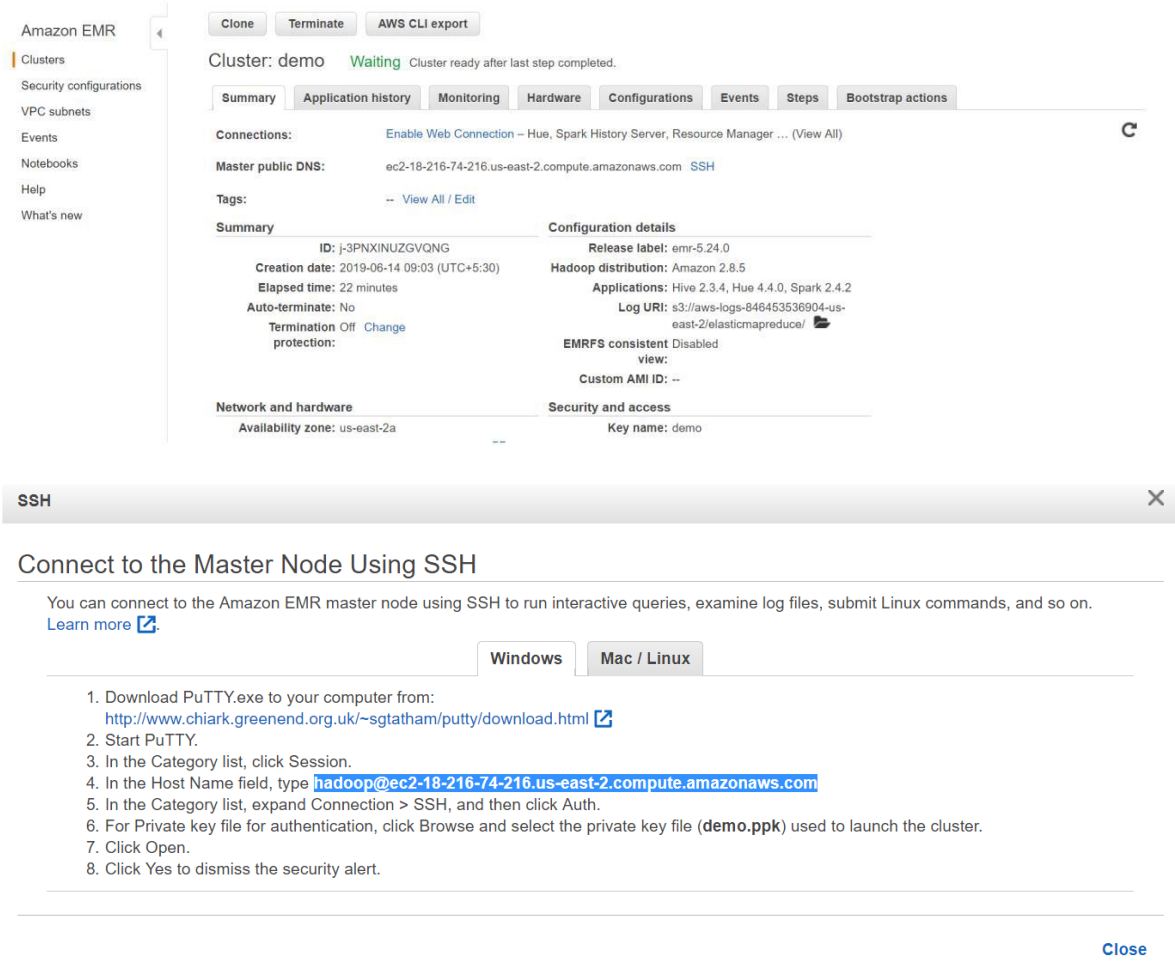
Step 10 : After few minutes, you can see the cluster status as "Waiting". It says that the cluster has been created successfully.



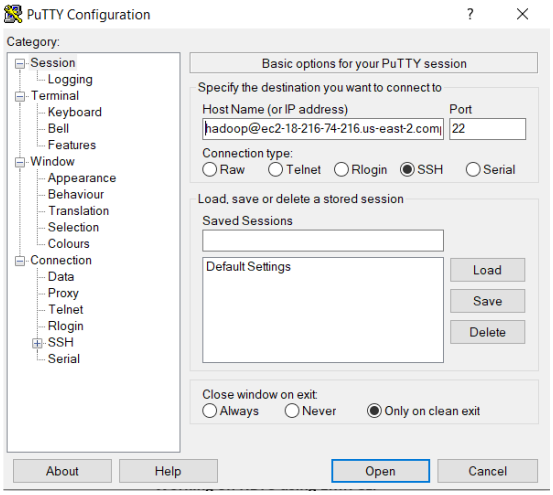
Connecting to EMR Cluster using Secure Shell (SSH)

Steps to connect to EMR cluster using SSH.

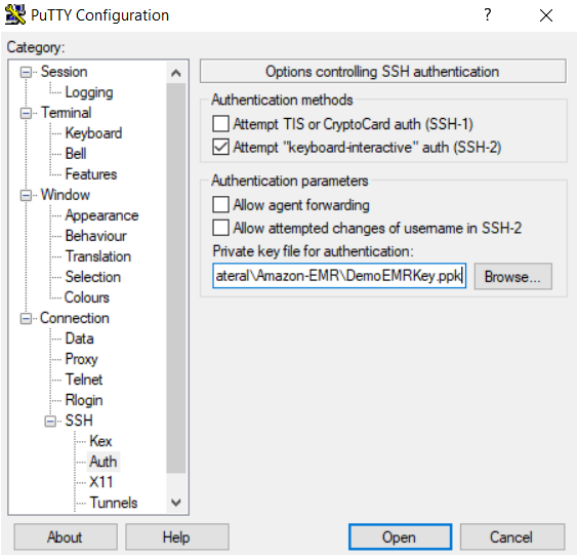
Step 1 : On the Clusters page click on SSH, copy the master host name.



Step 2 : Open putty / MobaXterm and type the host name as shown below. (**Note:** add **hadoop@** in front of the Host Name)

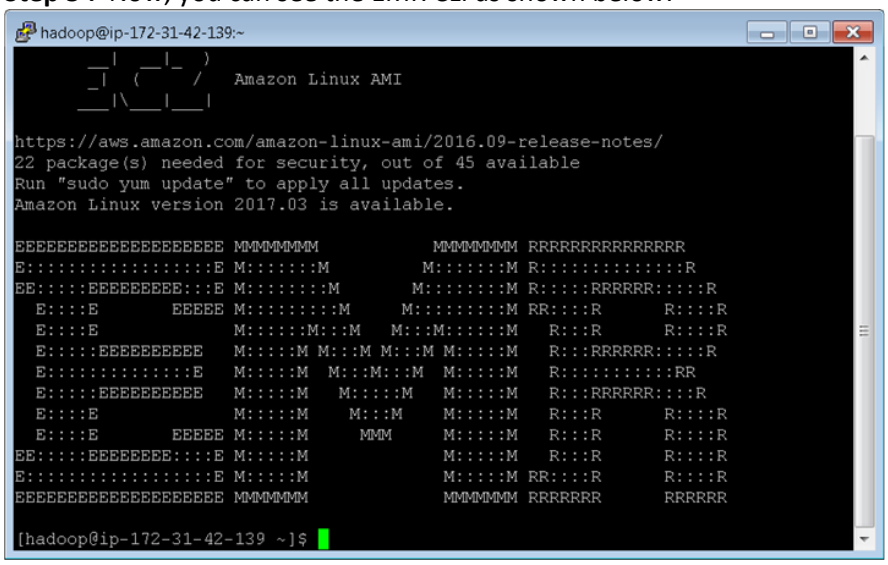


Step 3 : Next, expand SSH tab, select Auth and browse the Demo.ppk file as shown below.

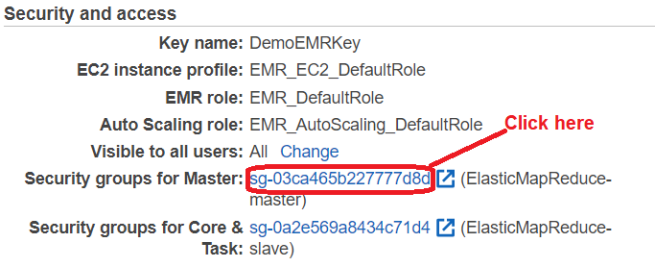


Step 4 : Next, click on open to connect to the EMR cluster.

Step 5 : Now, you can see the EMR CLI as shown below.



Note : if the putty doesn't connect , From the **Summary Tab** of the cluster - Edit the inbound rules of the Master Node and add SSH rule for your host machine IP -



Working on HDFS using EMR CLI

In this lab, we are going to discuss how to work with HDFS commands using EMR CLI.
1. To create the directory “demo”.

```
[hadoop@ip-172-31-26-223 ~]$ hadoop fs -mkdir /demo
[hadoop@ip-172-31-26-223 ~]$
```

2. To List the HDFS root directory.

```
[hadoop@ip-172-31-26-223 ~]$ hadoop fs -ls /
Found 5 items
drwxr-xr-x - hdfs hadoop 0 2019-06-17 09:16 /apps
drwxr-xr-x - hadoop hadoop 0 2019-06-17 09:32 /demo
drwxrwxrwt - hdfs hadoop 0 2019-06-17 09:18 /tmp
drwxr-xr-x - hdfs hadoop 0 2019-06-17 09:16 /user
drwxr-xr-x - hdfs hadoop 0 2019-06-17 09:16 /var
 [hadoop@ip-172-31-26-223 ~]$
```

3. To create a sample file in local filesystem.

```
[hadoop@ip-172-31-26-223 ~]$ cat > sample
welcome to hadoop session
hadoop session
hadoop definitive guide
[hadoop@ip-172-31-26-223 ~]$
```

4. To push the file named “sample” to HDFS demo directory using “put” command.

```
[hadoop@ip-172-31-26-223 ~]$ hadoop fs -put sample /demo/sample
[hadoop@ip-172-31-26-223 ~]$
```

5. To list HDFS demo directory.

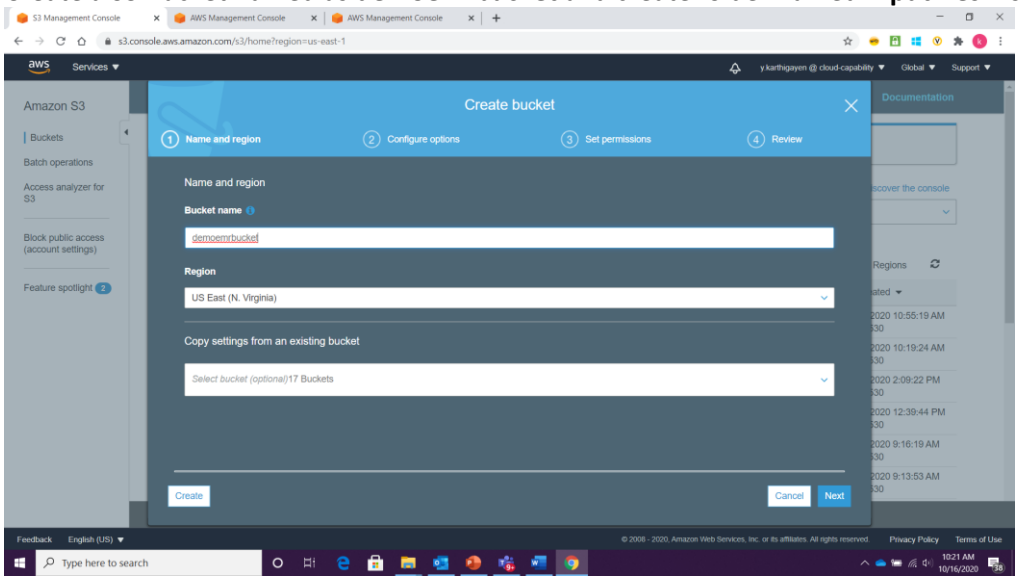
```
[hadoop@ip-172-31-26-223 ~]$ hadoop fs -ls /demo
Found 1 items
-rw-r--r-- 1 hadoop hadoop 65 2019-06-17 09:37 /demo/sample
[hadoop@ip-172-31-26-223 ~]$
```

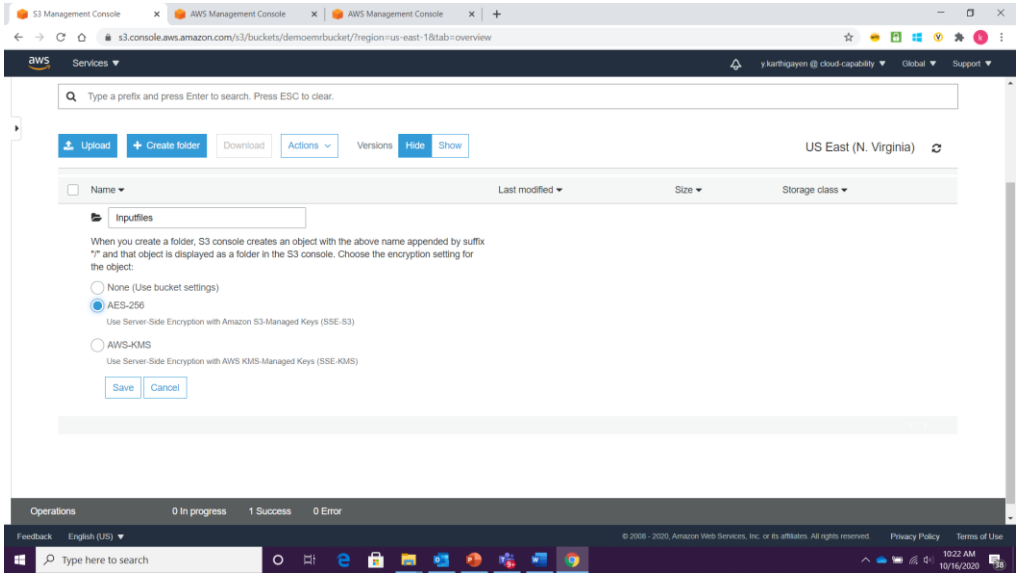
6. Retrieving the sample file content from HDFS demo directory to local filesystem using “get” command.

```
[hadoop@ip-172-31-26-223 ~]$ hadoop fs -cat /demo/sample
welcome to hadoop session
hadoop session
hadoop definitive guide
[hadoop@ip-172-31-26-223 ~]$
```

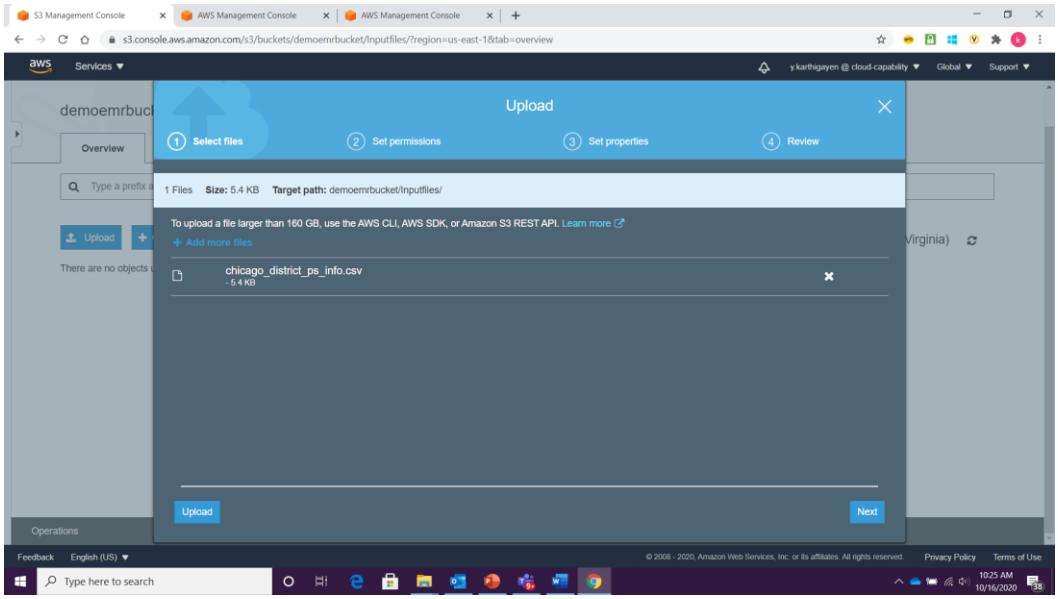
Injecting a File from AWS-S3 into EMR-HDFS :

- Create a S3 Bucket named as **demoemrbucket** and create folder named **Inputfiles** inside the bucket





- Now upload the files required for the case study into the Inputfiles folder



- Now connect with the SSH Terminal and create a directory named **EMR-Inputfiles on HDFS**

```
[hadoop@ip-172-31-47-106 ~]$ hadoop fs -mkdir /EMR-Inputfiles
[hadoop@ip-172-31-47-106 ~]$ hadoop fs -ls /
Found 5 items
drwxr-xr-x   - hadoop hadoop          0 2020-10-16 06:06 /EMR-Inputfiles
drwxr-xr-x   - hdfs   hadoop          0 2020-10-16 05:28 /apps
drwxrwxrwt   - hdfs   hadoop          0 2020-10-16 05:29 /tmp
drwxr-xr-x   - hdfs   hadoop          0 2020-10-16 05:28 /user
drwxr-xr-x   - hdfs   hadoop          0 2020-10-16 05:28 /var
[hadoop@ip-172-31-47-106 ~]$
```

- Now execute the below command to Inject the Input file from S3 into HDFS
- **hadoop distcp s3://demoemrbucket10479255kar/Inputfiles/chicago_crime_dataset.csv /Hivedemos**
- The above command will run a MR Job to copy the csv file from S3 to HDFS

```
[hadoop@ip-172-31-47-106 ~]$ hadoop fs -ls /EMR-Inputfiles
Found 1 items
-rw-r--r--   1 hadoop hadoop       5545 2020-10-16 05:53 /EMR-Inputfiles/chicago_district_ps_info.csv
[hadoop@ip-172-31-47-106 ~]$
```

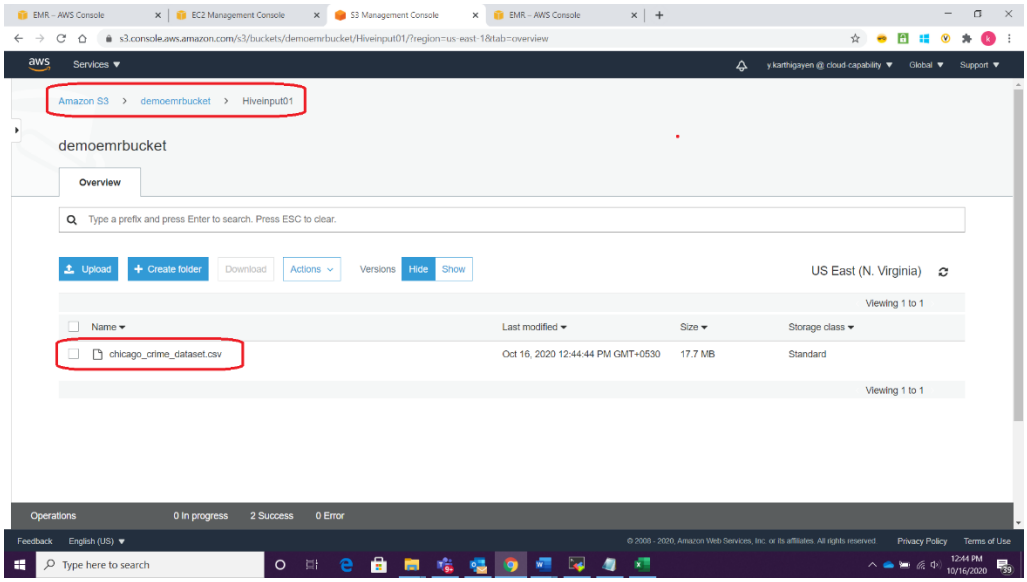
```
[hadoop@ip-172-31-47-106 ~]$ hadoop fs -cat /EMR-Inputfiles/chicago_district_ps_info.csv
DISTRICT_CODE,DISTRICT_NAME,ADDRESS,CITY,STATE,ZIP,WEBSITE,PHONE,FAX,TTY,X_COORDINATE,Y_COORDINATE,LATITUDE,LONGI
TUDE,LOCATION
1,Central,1718 S State St,Chicago,IL,60616,http://home.chicagopolice.org/community/districts/1st-district-central
/,312-745-4290,312-745-3694,312-745-3693,1176569.052,1891771.704,41.85837259,-87.62735617,"(41.8583725929, -87.62
7356171)"
2,Wentworth,5101 S Wentworth Ave,Chicago,IL,60609,http://home.chicagopolice.org/community/districts/2nd-district-
wentworth/,312-747-8366,312-747-5396,312-747-6656,1175864.837,1871153.753,41.80181109,-87.63056018,"(41.801811091
2, -87.6305601801)"
3,Grand Crossing,7040 S Cottage Grove Ave,Chicago,IL,60637,http://home.chicagopolice.org/community/districts/3rd-
district-grand-crossing/,312-747-8201,312-747-5479,312-747-9168,1182739.183,1858317.732,41.76643089,-87.60574786,
"(41.7664308925, -87.6057478606)"
4,South Chicago,2255 E 103rd St,Chicago,IL,60617,http://home.chicagopolice.org/community/districts/4th-district-s
outh-chicago/,312-747-7581,312-747-5276,312-747-9169,1193131.299,1837090.265,41.70793329,-87.56834912,"(41.707933
2906, -87.5683491228)"
5,Calumet,727 E 111th St,Chicago,IL,60628,http://home.chicagopolice.org/community/districts/5th-district-calumet/
,312-747-8210,312-747-5935,312-747-9170,1183305.427,1831462.313,41.69272336,-87.60450587,"(41.6927233639, -87.604
5058667)"
6,Gresham,7808 S Halsted St,Chicago,IL,60620,http://home.chicagopolice.org/community/districts/6th-district-gresh
am/,312-745-3617,312-745-3649,312-745-3639,1172283.013,1853022.646,41.75213684,-87.64422891,"(41.7521368378, -87.
6442289066)"
```

Solving the Case-Study Uses Cases Using the Data Processing Frameworks on Amazon-EMR

Hadoop & Hive Processing Framework :

Use Case 1 : Create report on total number of crime cases on each day from crimes dataset

Steps 1: Upload the required input files into S3 bucket inside the respective folders



Step2 : Inject the inputfile from S3 into HDFS inside the Hivedemos Folder

```
[hadoop@ip-172-31-47-106 ~]$
[hadoop@ip-172-31-47-106 ~]$ hadoop fs -mkdir /Hivedemos
[hadoop@ip-172-31-47-106 ~]$ hadoop distcp  s3://demoemrbucket/Hiveinput01/chicago_crime_dataset.csv /Hivedemos
```

Step3 : get into the Hive prompt and create a Database named **daonemr**

- **set hive.cli.print.current.db=true**

```
hive> create database DAonEMR;
OK
Time taken: 0.626 seconds
hive> show databases;
OK
daonemr
default
Time taken: 0.137 seconds, Fetched: 2 row(s)
hive> use daonemr;
OK
Time taken: 0.037 seconds
hive> set hive.cli.print.current.db=true;
hive (daonemr)>
```

Step4 : Now let us create the schema for the Crime dataset input file on Hive.

create EXTERNAL table IF NOT EXISTS CrimeData (ID INT, CaseNo STRING, DateofCrime DATE, Block STRING, IUCR_Code STRING, Location_Desc STRING, Arrest STRING, Domestic STRING, Beat_Num INT, District_Code INT, Ward_No INT, Community_Code INT, FBI_Code STRING, X_Coord INT, Y_Coord INT, Year INT, Date_Of_Update STRING, Latitude FLOAT, Longitude FLOAT, Location STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
tblproperties("skip.header.line.count"="1");

```
hive (daonemr)> create EXTERNAL table IF NOT EXISTS CrimeData (ID INT, CaseNo STRING, DateofCrime DATE, Block STR
ING, IUCR_Code STRING, Location_Desc STRING, Arrest STRING, Domestic STRING, Beat_Num INT, District_Code INT, War
d_No INT, Community_Code INT, FBI_Code STRING, X_Coord INT, Y_Coord INT, Year INT, Date_Of_Update STRING, Latitud
e FLOAT, Longitude FLOAT, Location STRING)
              > ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
              > tblproperties("skip.header.line.count"="1");
OK
Time taken: 0.427 seconds
hive (daonemr)> desc crimedata;
OK
id                      int
caseno                  string
dateofcrime             date
block                   string
iucr_code               string
location_desc           string
arrest                  string
domestic                string
beat_num                int
district_code           int
ward_no                 int
community_code          int
fbi_code                string
x_coord                 int
y_coord                 int
year                    int
date_of_update          string
latitude                float
longitude               float
location                string
```

Step5: Load the data from HDFS into the Hive Table **CrimeData** using the below command.

LOAD DATA INPATH '/Hivedemos/chicago_crime_dataset.csv' INTO TABLE CrimeData;

```
hive (daonemr)> select * from crimedata limit 10;
OK
10508693        HZ250496        2016-05-03      013XX S SAWYER AVE      486     APARTMENT       TRUE    TRUE    1
022     10      24      29      08B     1154907 1893681 2016    5/10/2016 15:56 41.864075       -87.70682       "
(41.864073157
10508695        HZ250409        2016-05-03      061XX S DREXEL AVE      486     RESIDENCE       FALSE   TRUE    3
13      3       20      42      08B     1183066 1864330 2016    5/10/2016 15:56 41.78292        -87.60436       "
(41.782921527
10508697        HZ250503        2016-05-03      053XX W CHICAGO AVE     470     STREET  FALSE   FALSE   1524    1
5       37      25      24      1140789 1904819 2016    5/10/2016 15:56 41.89491        -87.75837       "(41.8949
08283
10508698        HZ250424        2016-05-03      049XX W FULTON ST       460     SIDEWALK        FALSE   FALSE   1
532     15      28      25      08B     1143223 1901475 2016    5/10/2016 15:56 41.885685       -87.74952       "
(41.885686845
10508699        HZ250455        2016-05-03      003XX N LOTUS AVE       820     RESIDENCE       FALSE   TRUE    1
523     15      28      25      6       1139890 1901675 2016    5/10/2016 15:56 41.8863 -87.76175       "(41.8862
97242
10508702        HZ250447        2016-05-03      082XX S MARYLAND AVE    041A    STREET  FALSE   FALSE   631     6
8       44      04B     1183336 1850642 2016    5/10/2016 15:56 41.745354       -87.6038        "(41.745354023
10508703        HZ250489        2016-05-03      027XX S STATE ST        460     CHA HALLWAY/STAIRWELL/ELEVATOR  F
ALSE    FALSE   133     1       3       35      08B     1176730 1886544 2016    5/10/2016 15:56 41.844025       -
87.62692        "(41.844023772
10508704        HZ250514        2016-05-03      002XX E 46TH ST 460     RESIDENCE PORCH/HALLWAY FALSE   FALSE   2
15      2       3       38      08B     1178514 1874573 2016    5/10/2016 15:56 41.811134       -87.62074       "
(41.811133958
10508709        HZ250523        2016-05-03      014XX W DEVON AVE       460     SIDEWALK        FALSE   FALSE   2
432     24      40      1       08B     1165696 1942616 2016    5/10/2016 15:56 41.99813        -87.66582       "
(41.99813061
10508982        HZ250667        2016-05-03      069XX S ASHLAND AVE     486     STREET  FALSE   TRUE    735     7
17      67      08B     1166876 1858796 2016    5/10/2016 15:56 41.768097       -87.66388       "(41.768096835
```

Step6 : Execute the below Analytical Hive Query

SELECT dateofcrime, COUNT(caseno)
FROM crimedata
GROUP BY dateofcrime
SORT BY dateofcrime
LIMIT 20;

```
--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 5.39 s
--------------------------------------------------------------------------------
OK
2012-01-01      1035
2012-01-02      645
2012-01-03      765
2012-01-04      765
2012-01-05      823
2012-01-06      881
2012-01-07      816
2012-01-08      776
2012-01-09      834
2012-01-10      847
2012-01-11      847
2012-01-12      696
2012-01-13      721
2012-01-14      740
2012-01-15      708
2012-01-16      788
2012-01-17      789
2012-01-18      736
2012-01-19      693
2012-01-20      598
```

Redirecting the output to HDFS :

Run the below query to redirect the output to HDFS:

INSERT OVERWRITE DIRECTORY '/Hivedemos/Outputfiles'
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
SELECT dateofcrime, COUNT(caseno)
FROM crimedata
GROUP BY dateofcrime
SORT BY dateofcrime;

```
hive (daonemr)> INSERT OVERWRITE DIRECTORY '/Hivedemos/Outputfiles'
              > ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
              > STORED AS TEXTFILE
              > SELECT dateofcrime, COUNT(caseno)
              > FROM crimedata
              > GROUP BY dateofcrime
              > SORT BY dateofcrime;
Query ID = hadoop_20201020055410_08cf4e8b-0104-4367-9124-1632353ca132
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1603171549380_0005)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 5.22 s
--------------------------------------------------------------------------------
Moving data to directory /Hivedemos/Outputfiles
OK
Time taken: 6.799 seconds
hive (daonemr)>
```

Now Check the Output files on HDFS folder

```
hive (daonemr)> dfs -ls /Hivedemos;
Found 1 items
drwxr-xr-x   - hadoop hadoop          0 2020-10-20 05:54 /Hivedemos/Outputfiles
hive (daonemr)> dfs -ls /Hivedemos/Outputfiles;
Found 2 items
-rwxr-xr-x   1 hadoop hadoop       9874 2020-10-20 05:54 /Hivedemos/Outputfiles/000000_0
-rwxr-xr-x   1 hadoop hadoop       9016 2020-10-20 05:54 /Hivedemos/Outputfiles/000001_0
hive (daonemr)> dfs -cat /Hivedemos/Outputfiles/000000_0;
2012-01-01      1035
2012-01-03      765
2012-01-04      765
2012-01-06      881
2012-01-10      847
2012-01-13      721
2012-01-14      740
2012-01-15      708
2012-01-18      736
2012-01-20      598
2012-01-22      649
2012-01-24      820
2012-01-26      810
2012-01-27      787
2012-01-28      714
2012-01-30      560
2012-02-02      3
2012-02-03      1
2012-02-04      1
2012-02-05      2
2012-02-06      1
2012-02-08      1
2012-02-09      1
```

Use Case 2 : Create report on total number of crime cases on each day for each district from crimes dataset

Note : For this use case we can read the data directly from S3

Step1 : Create an Internal table with below schema

create table IF NOT EXISTS CrimeData_internal (ID INT, CaseNo STRING, DateofCrime DATE, Block STRING, IUCR_Code STRING, Location_Desc STRING, Arrest STRING, Domestic STRING, Beat_Num INT, District_Code INT, Ward_No INT, Community_Code INT, FBI_Code STRING, X_Coord INT, Y_Coord INT, Year INT, Date_Of_Update STRING, Latitude FLOAT, Longitude FLOAT, Location STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE
LOCATION 's3://demoemrbucket-10479255/Inputfiles'
tblproperties("skip.header.line.count"="1");

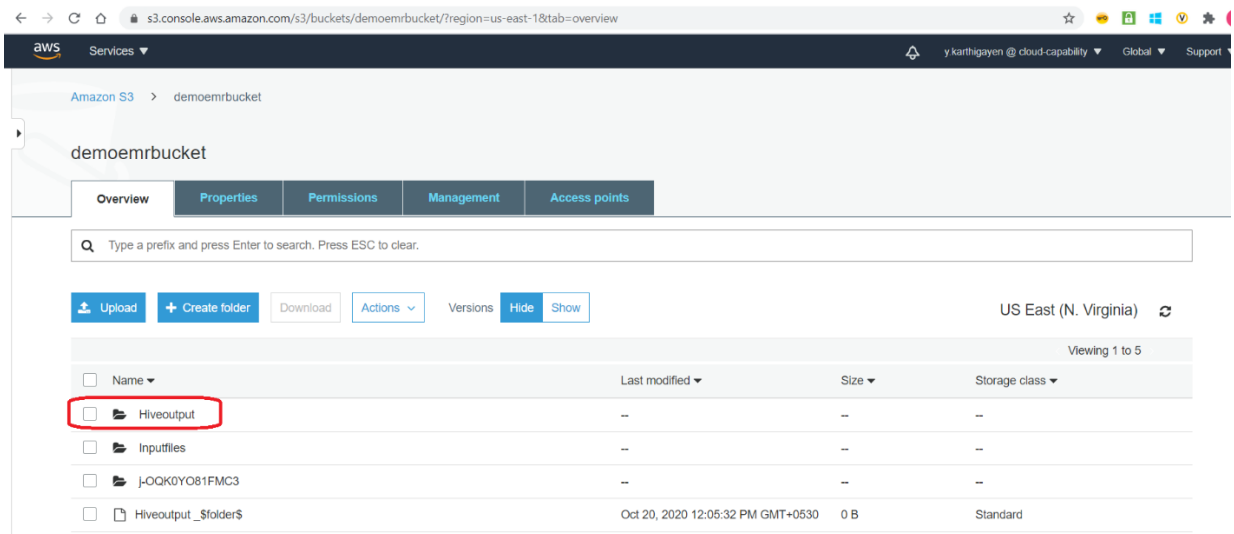
```
hive (daonemr)> create table IF NOT EXISTS CrimeData_internal (ID INT, CaseNo STRING, DateofCrime DATE, Bl
ock STRING, IUCR_Code STRING, Location_Desc STRING, Arrest STRING, Domestic STRING, Beat_Num INT, District
_Code INT, Ward_No INT, Community_Code INT, FBI_Code STRING, X_Coord INT, Y_Coord INT, Year INT, Date_Of_U
pdate STRING, Latitude FLOAT, Longitude FLOAT, Location STRING)
              > ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
              > LINES TERMINATED BY '\n'
              > STORED AS TEXTFILE
              > LOCATION 's3://demoemrbucket/Inputfiles'
              > tblproperties("skip.header.line.count"="1");
OK
Time taken: 4.4 seconds
hive (daonemr)> select * from crimedata_internal limit 10;
OK
10508693        HZ250496        2016-05-03      013XX S SAWYER AVE      486     APARTMENT       TRUE    TR
UE      1022    10      24      29      08B     1154907 1893681 2016    5/10/2016 15:56 41.864075       -8
7.70682 "(41.864073157
10508695        HZ250409        2016-05-03      061XX S DREXEL AVE      486     RESIDENCE       FALSE   TR
UE      313     3       20      42      08B     1183066 1864330 2016    5/10/2016 15:56 41.78292        -8
7.60436 "(41.782921527
10508697        HZ250503        2016-05-03      053XX W CHICAGO AVE     470     STREET  FALSE   FALSE   15
24      15      37      25      24      1140789 1904819 2016    5/10/2016 15:56 41.89491        -87.75837"
(41.894908283
10508698        HZ250424        2016-05-03      049XX W FULTON ST       460     SIDEWALK        FALSE   FA
LSE     1532    15      28      25      08B     1143223 1901475 2016    5/10/2016 15:56 41.885685       -8
7.74952 "(41.885686845
10508699        HZ250455        2016-05-03      003XX N LOTUS AVE       820     RESIDENCE       FALSE   TR
UE      1523    15      28      25      6       1139890 1901675 2016    5/10/2016 15:56 41.8863 -87.76175"
(41.886297242
```

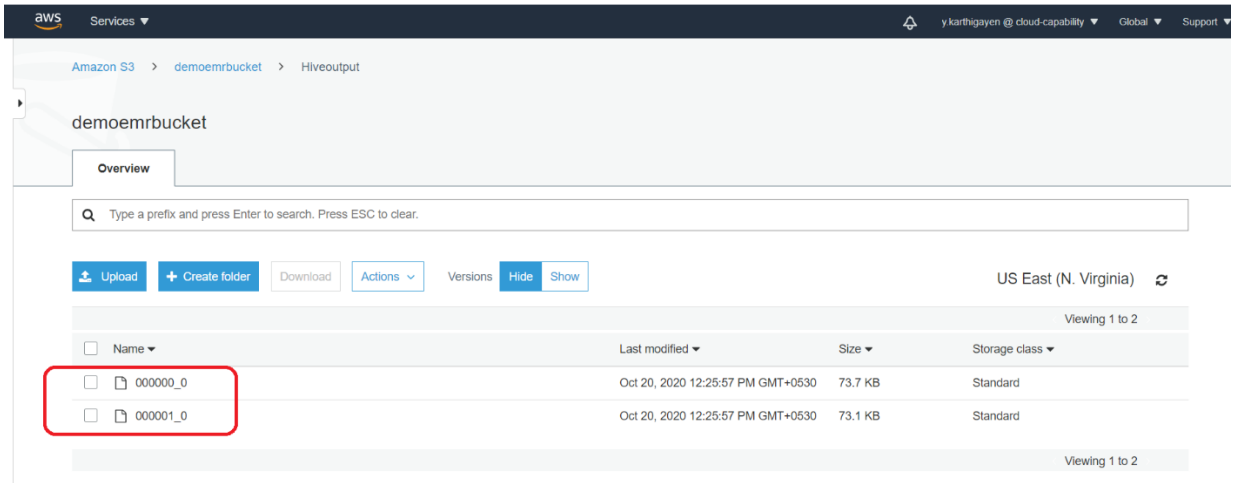
Step2 : Trigger the below Analytical Hive Query on this table and re-direct the output to a folder on S3

INSERT OVERWRITE DIRECTORY 's3://demoemrbucket-10479255/Hiveoutput '
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
SELECT dateofcrime, district_code, COUNT(caseno)
FROM crimedata_internal
GROUP BY dateofcrime , district_code
SORT BY dateofcrime , district_code ;

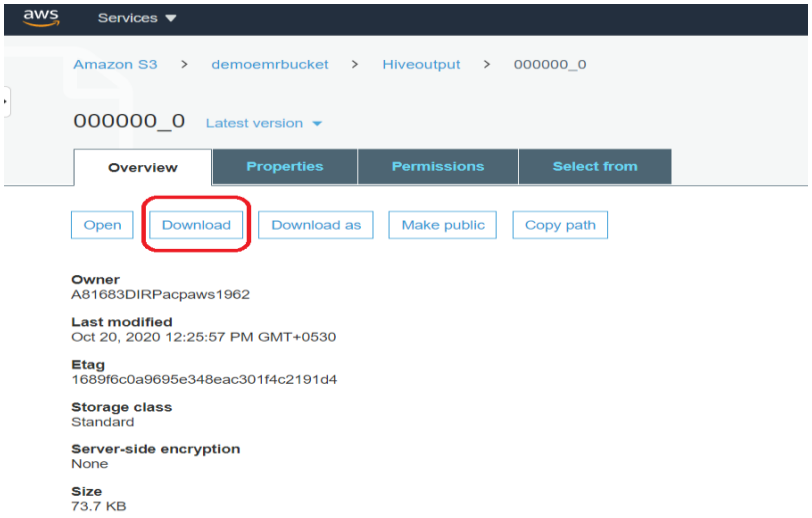
```
hive (daonemr)> INSERT OVERWRITE DIRECTORY 's3://demoemrbucket/Hiveoutput '
              > ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
              > STORED AS TEXTFILE
              > SELECT dateofcrime, district_code, COUNT(caseno)
              > FROM crimedata_internal
              > GROUP BY dateofcrime , district_code
              > SORT BY dateofcrime , district_code ;
Query ID = hadoop_20201020065539_e3fe0b1a-3bcf-41cf-9ed0-8062e13f8ca3
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1603171549380_0010)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 10.61 s
--------------------------------------------------------------------------------
Moving data to directory s3://demoemrbucket/Hiveoutput
OK
Time taken: 17.288 seconds
hive (daonemr)>
```

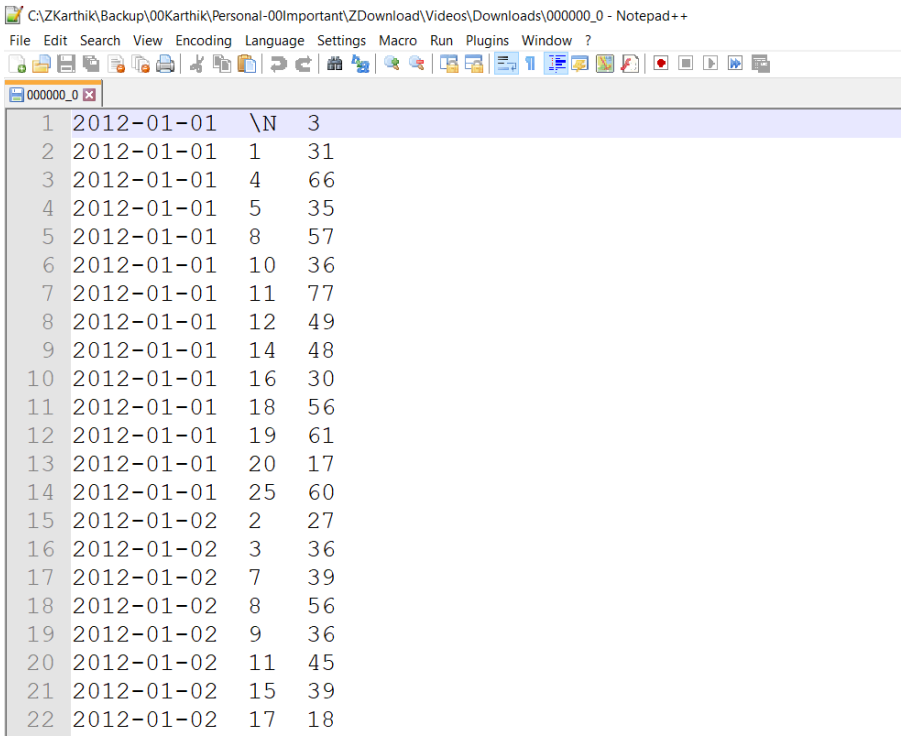
Step 3 : Now we can verify the Output files from the S3 folder





Step 4 : Download the part files and check the output





Presto : Demo

Note : Launch the EMR Cluster along with Presto

- First run a Analytical Query on HIVE and check the Time taken for fetch the results

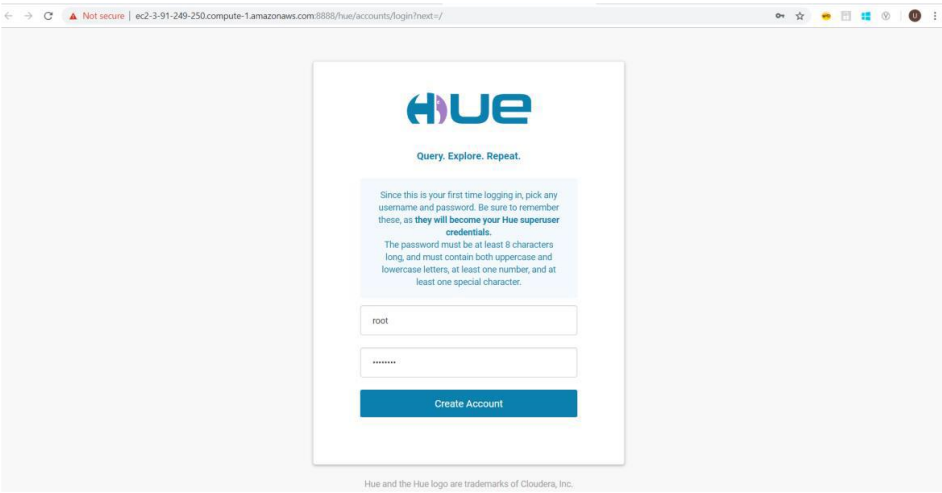
```
hive (daonemr)> SELECT dateofcrime, COUNT(caseno) FROM CrimeData_internal GROUP BY dateofcrime LIMIT 10;
Query ID = hadoop_20201118105131_4692d285-3401-40a7-920b-6733e402d418
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1605695419006_0002)

--------------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
--------------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 9.82 s
--------------------------------------------------------------------------------------------------
OK
2012-01-01      1035
2012-01-03      765
2012-01-04      765
2012-01-06      881
2012-01-10      847
2012-01-02      645
2012-01-05      823
2012-01-07      816
2012-01-08      776
2012-01-09      834
Time taken: 12.942 seconds, Fetched: 10 row(s)
hive (daonemr)>
```

- Now let us try the same Analytical Query on Presto, and the Time taken to Fetch the Results

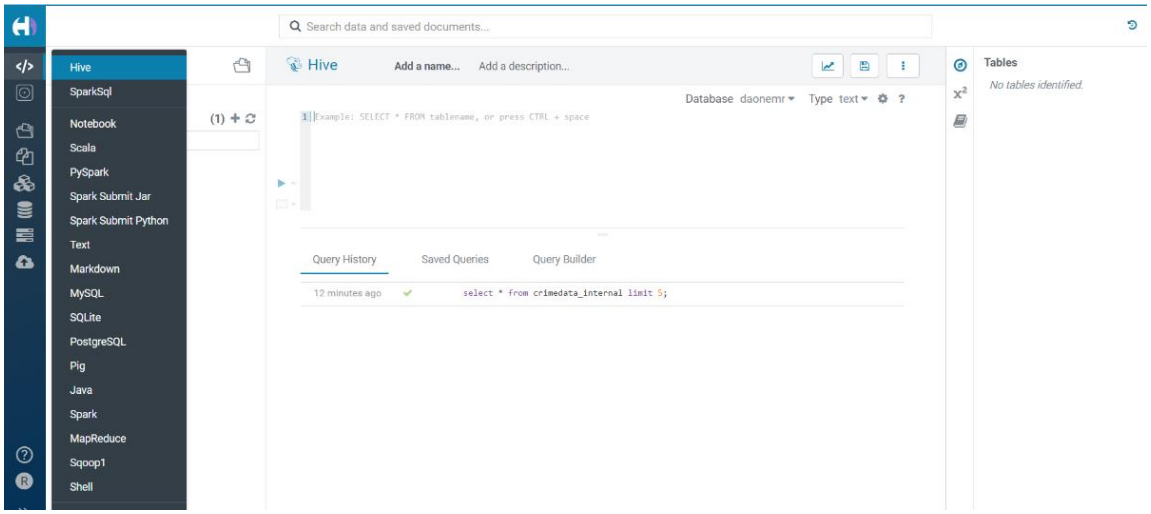
```
[hadoop@ip-172-31-34-73 ~]$ presto-cli
presto> use hive.daonemr;
USE
presto:daonemr> SELECT dateofcrime, COUNT(caseno) FROM CrimeData_internal GROUP BY dateofcrime LIMIT 10;
 dateofcrime | _col1
-------------+-------
 2016-01-01  |    25
 2016-04-28  |   107
 2016-03-31  |     6
 2016-05-05  |   660
 2015-07-01  |    12
 2015-08-13  |     3
 2015-04-17  |     5
 2012-02-13  |     1
 2016-03-27  |     2
 2012-05-28  |     1
(10 rows)

Query 20201118_105320_00005_8s6cp, FINISHED, 2 nodes
Splits: 82 total, 82 done (100.00%)
0:01 [100K rows, 16.8MB] [87.3K rows/s, 14.7MB/s]

presto:daonemr>
```

Working with HUE :

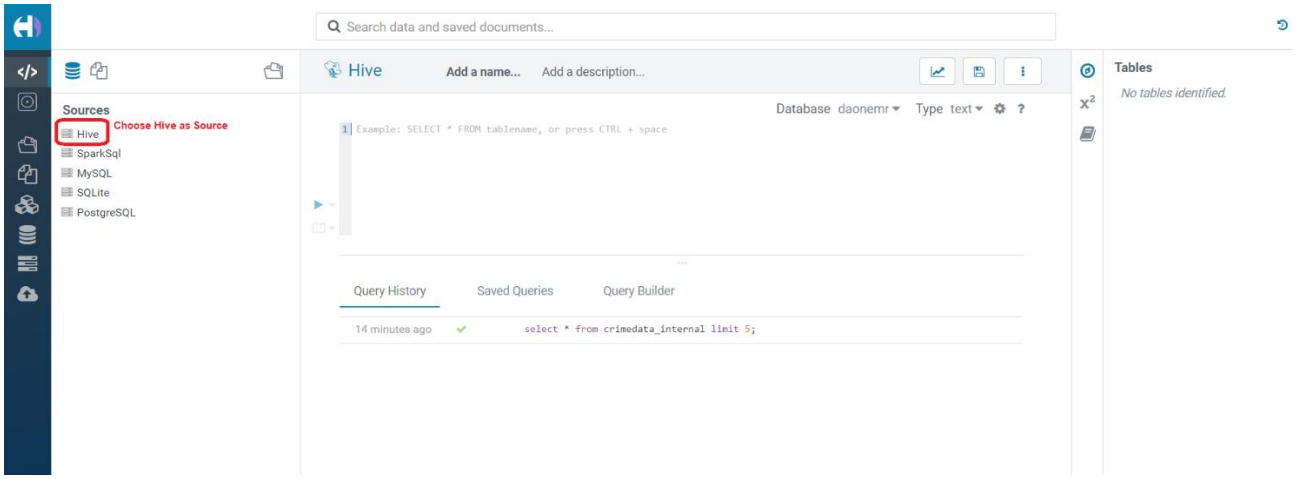
- After creating a EMR Cluster with HUE, Add **Custom TCP** with **port 8888** in the in-bound rules for Master Node Security Group
- Access HUE using a Browser as --- : http://< **Master public DNS**>:8888
- Since this is your first time log in, specify any username and password.
- These username and password will become your Hue superuser credentials.
- The password must be at least 8 characters long, and must contain both uppercase and lowercase letters, at least one number, and at least one special character.
- In our case we choose the **user name as "root: and password as "Root@1234$$".**



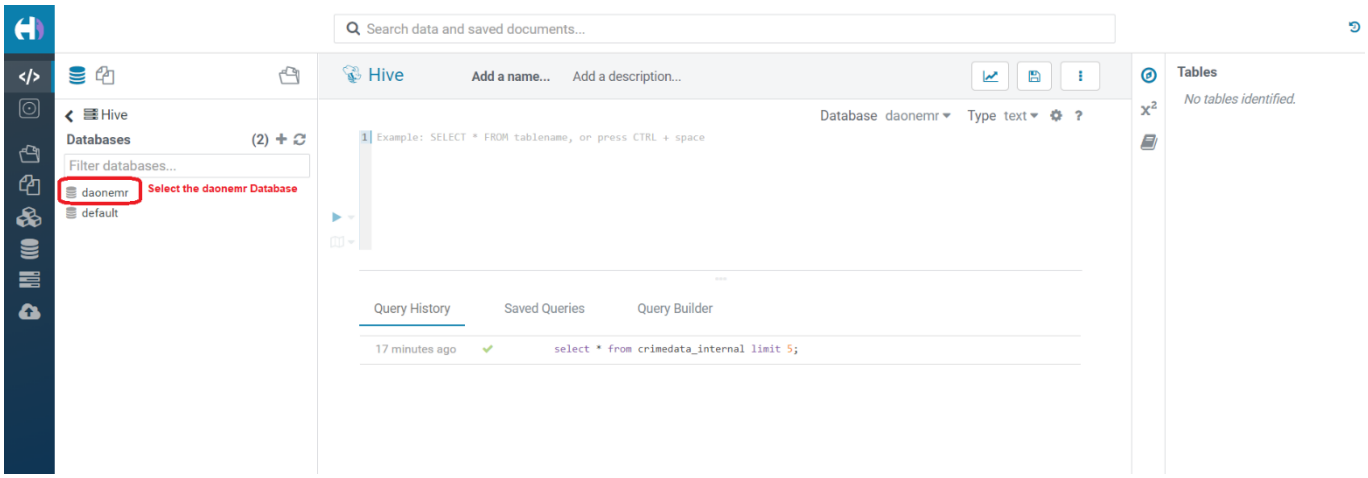
- By default, Hue will load with Hive Querying Editor.



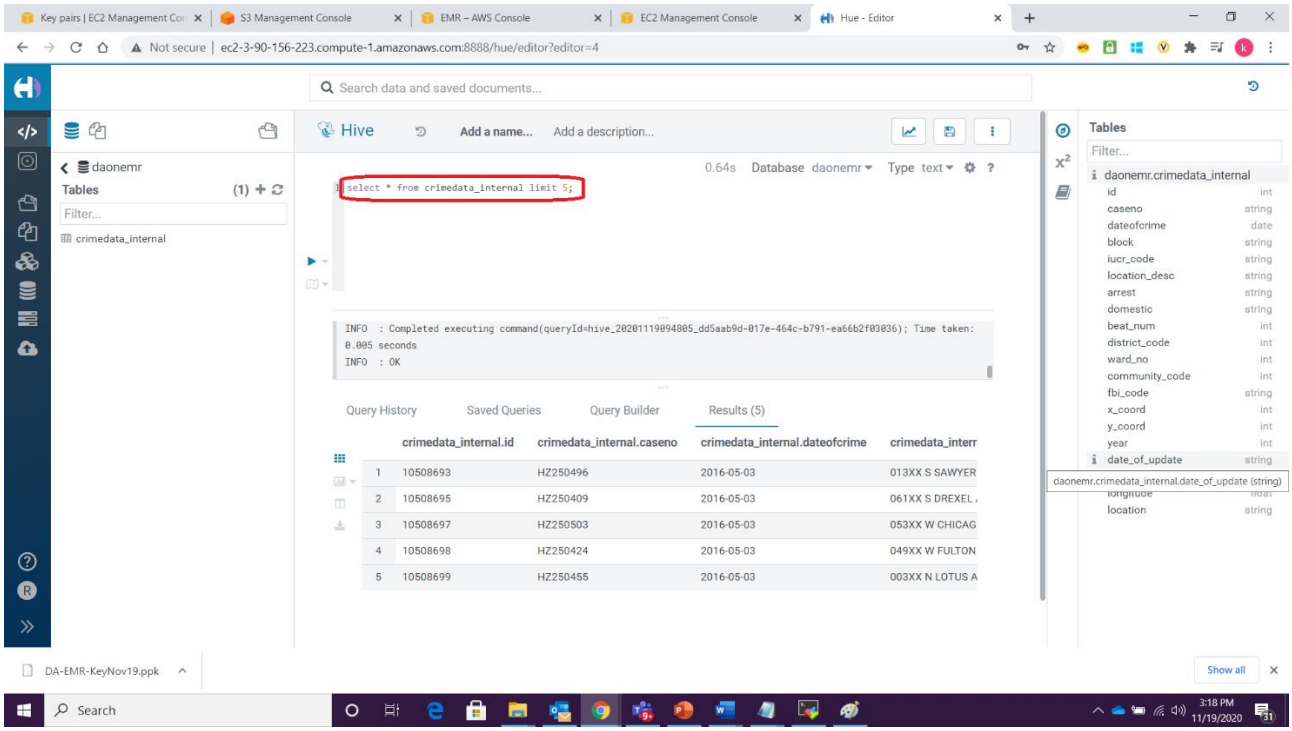
- We can select the source with which we want work.



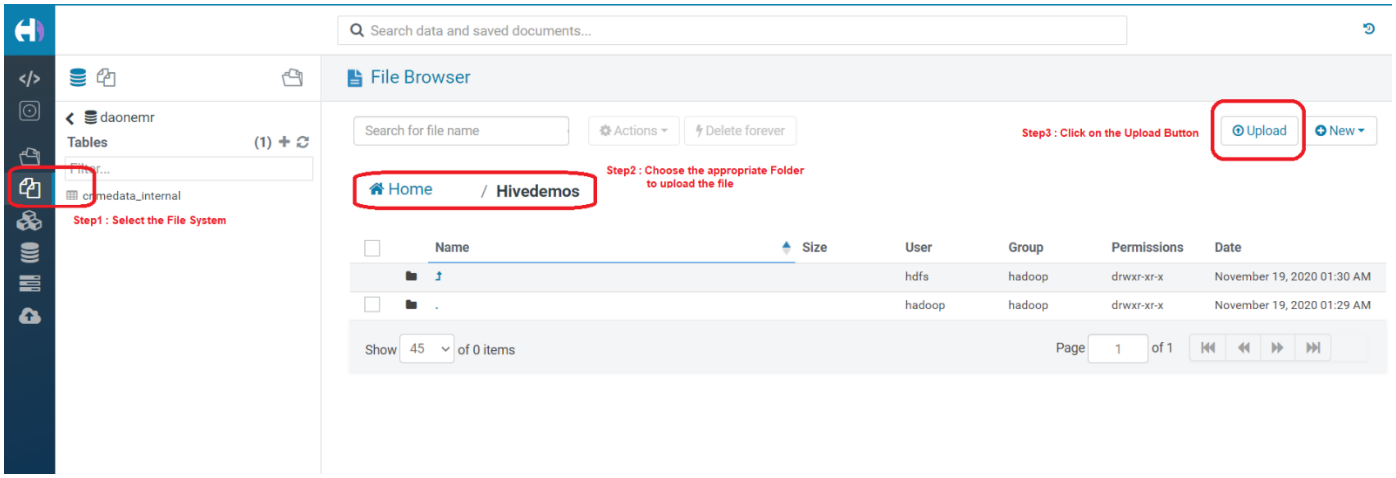
- Now choose the respective Database which we had created in Hive.

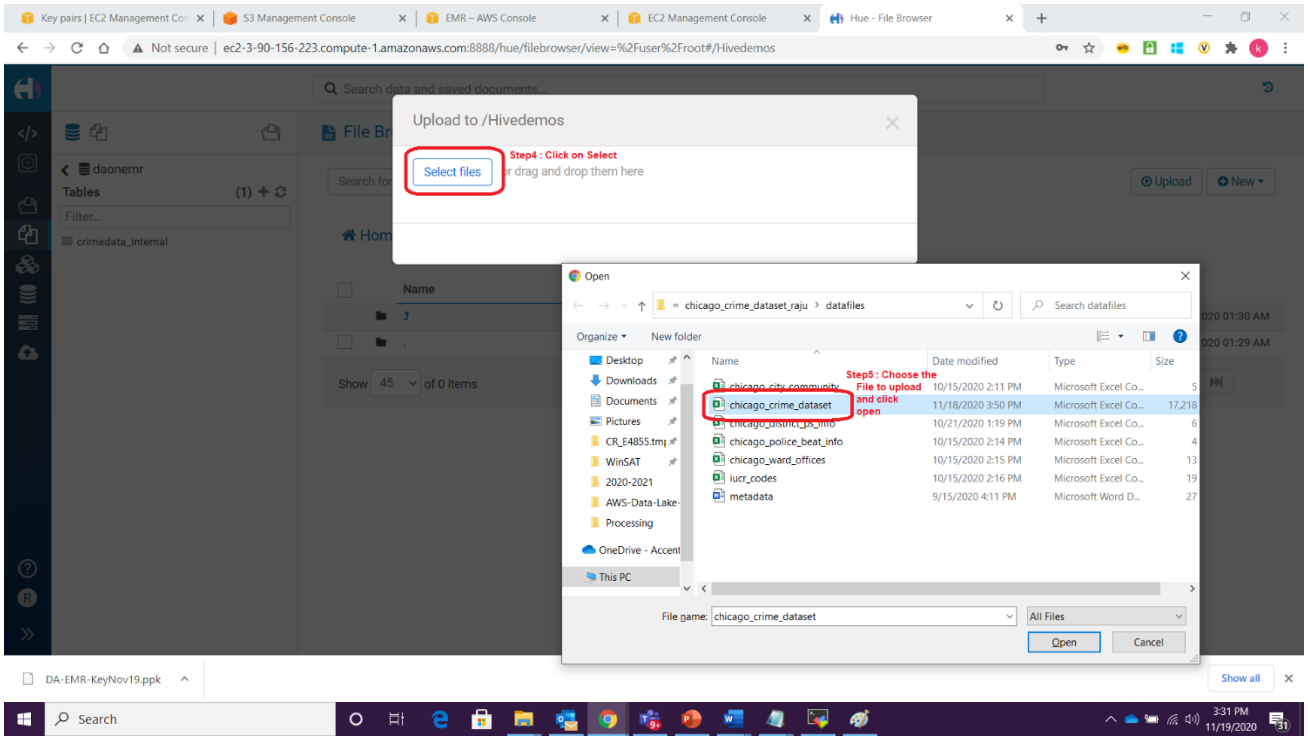


- We can now run the queries against the tables in that Database.



Uploading a File into HDFS Folder Using HUE :




Upload to /Hivedemos
Select files
Step4 : Click on Select
Step5 : Choose the File to upload and click open
chicago_crime_dataset
chicago_crime_dataset_raju
datafiles
Open
Cancel