INSTITUTE FOR ADVANCED
COMPUTING AND
SOFTWARE
DEVELOPME
NT AKURDI,
PUNE

Documentation On

" US Federal Campaign Finance Data Analysis Using BigData Technologies "

PG-DBDA MARCH 2022

Submitted By:

Group No: 06

UMESH BHARUDE 223313

Mr. Prashant Karhale
Center Coordinator

Mr. Akshay Tilekar
Project Guide

Contents

1.Introduction

1.1 PROBLEM STATEMENT

To analyze the campaign finance & contributions (individuals & PACs) for all federal elections over the last 2 decades & a half.

1.2 Abstract

US FEDERAL CAMPAIGN FINANCE DATA ANALYSIS using AWS services like S3 Bucket, EMR, CFT, Spark, Hive and PowerBI. To analyze the historical data of US Federal Campaign's finance and contributions made individually and through PACs. I have determined the relationship between various features and their impact on contribution patterns for candidates. Performed ETL in Apache Spark and displayed respective visualizations in Power BI.

The dataset we have taken from kaggle.

Below is the following steps how we imported the data and store the shell script file in s3 bucket

2. TOOLS & TECHNOLOGIES

1. S3 → Storing raw data
2. Extract data from Kaggle and storing in Amazon S3 Bucket
3. CFT : Creating EMR

4.EMR :

Apache Hadoop Ecosystem:

a. HDFS → Using Hadoop distributed file system for data storage

b. Apache PySpark → Combine various tables for proper analysis and performing ETL

c. Apache Hive → To process structured Data

5. Languages to use:

a. Python → ETL and Data Processing

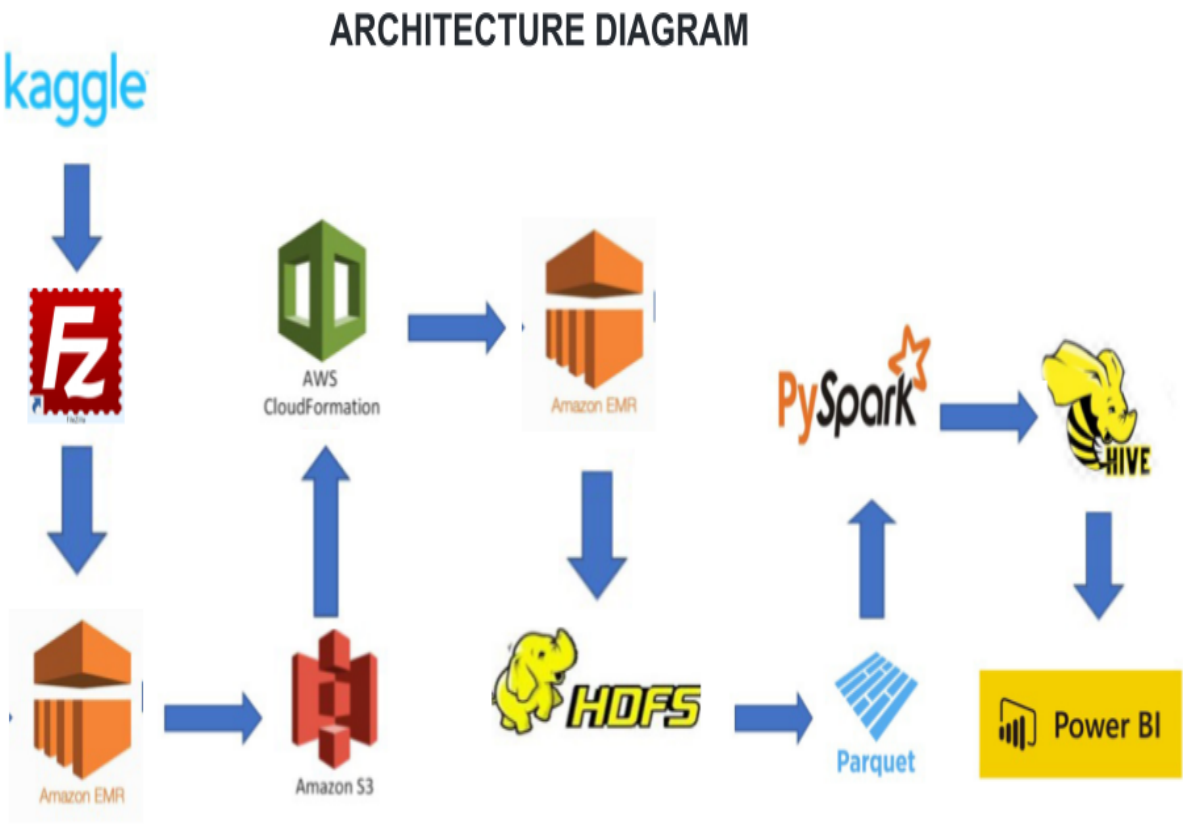
b. HQL → Creating Hive tables and loading data

c. Shell Script → Automation

6. File Formats to be used:

a. CSV b. Parquet

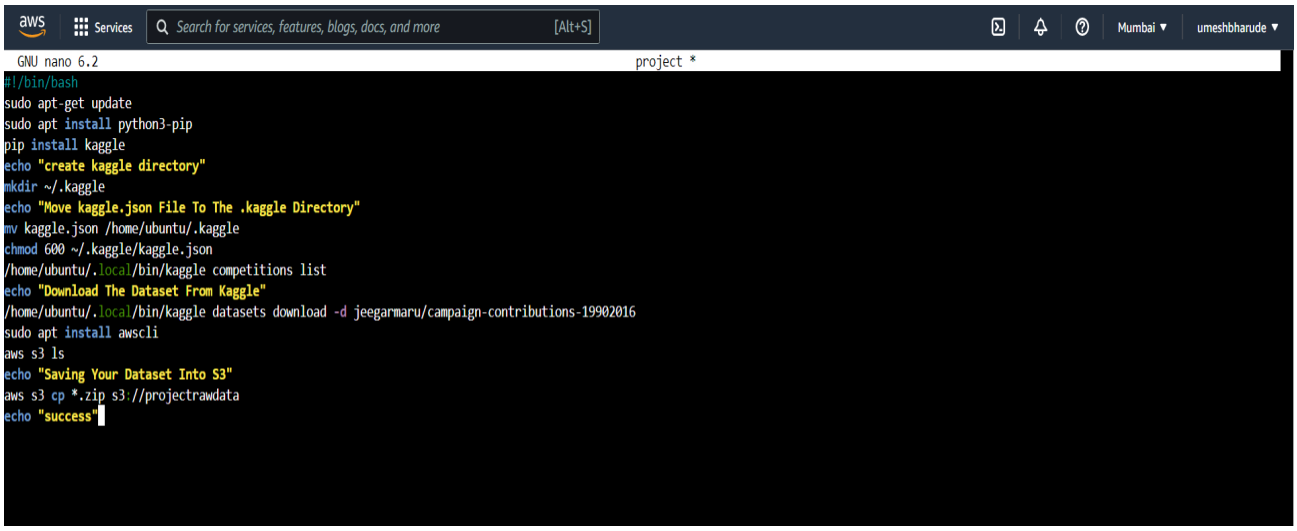
7.Power BI: Creating Dashboard



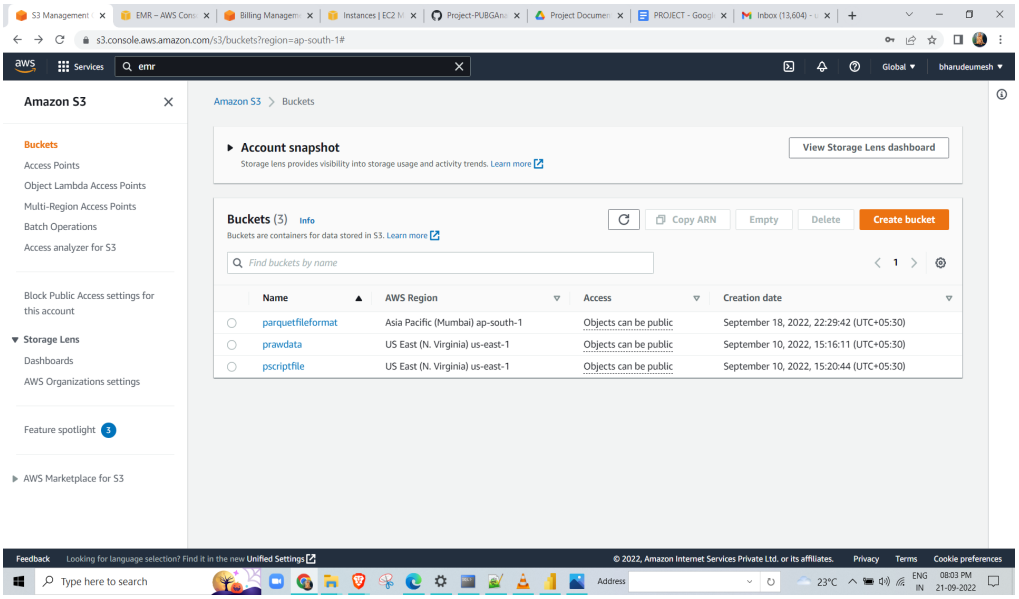
1) Upload dataset from kaggle to EMR:

- Go to the kaggle website.
- Click on Your profile button on the top right and then select My Account.
- Scroll down to the API section and click on the Create New API Token button.
- It will download the file kaggle.json & Save the file at a known location on your machine.
- Move the downloaded file to a location: `~/.kaggle/kaggle.json`.
- Create .kaggle folder in your home directory, you can create one using the command: `mkdir ~/.kaggle`
- Now move the downloaded file to this location using:
- `mv <location>/kaggle.json ~/.kaggle/kaggle.json`

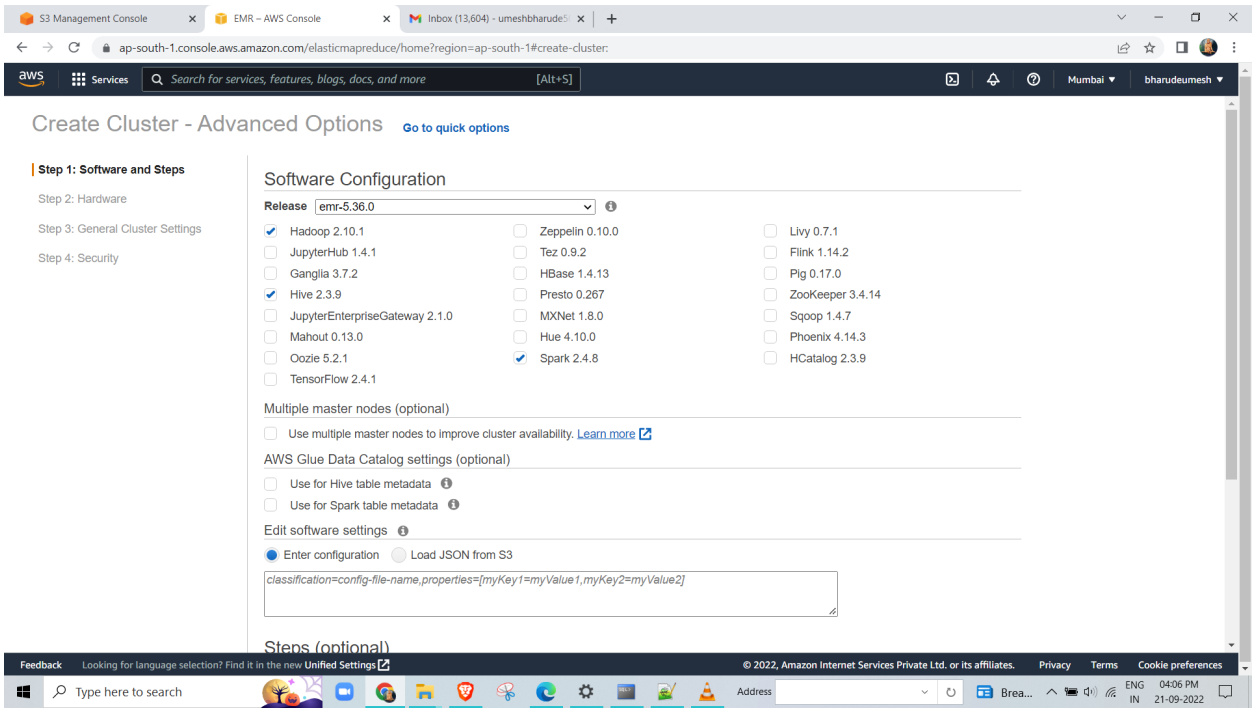
- You need to give proper permissions to the file:
- `chmod 600 ~/.kaggle/kaggle.json`
- The run the command as ...> *kaggle competitions list*

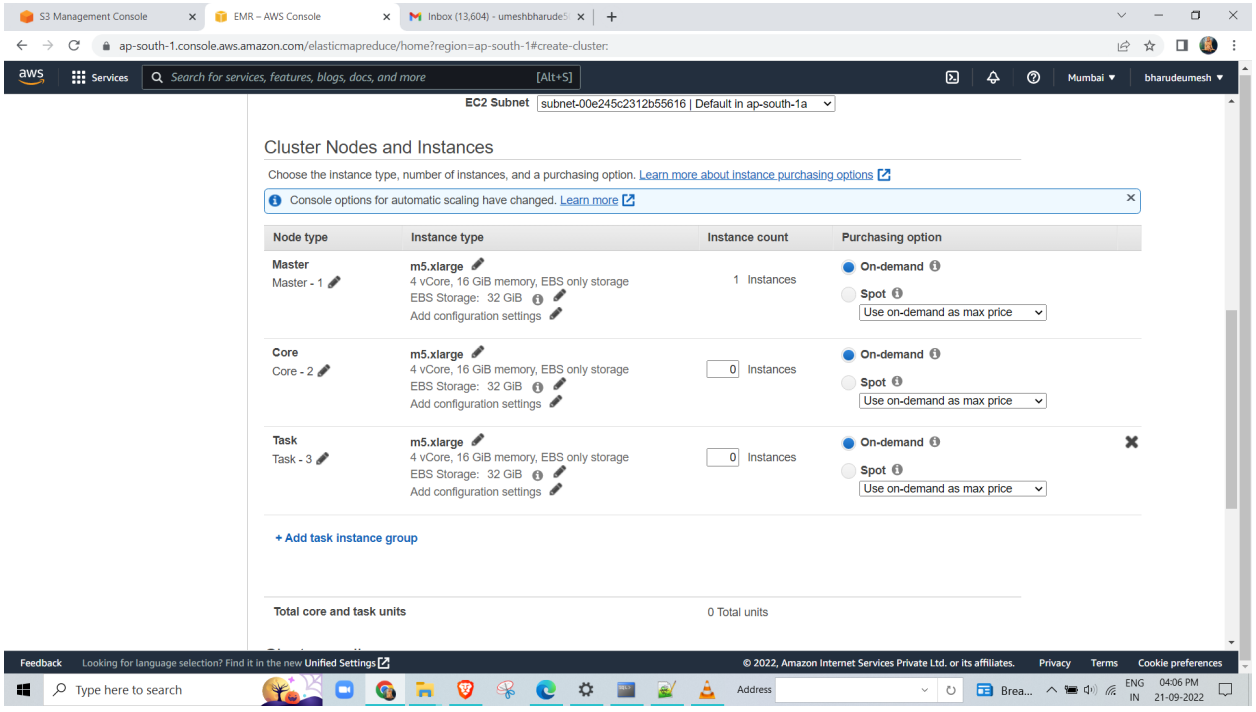


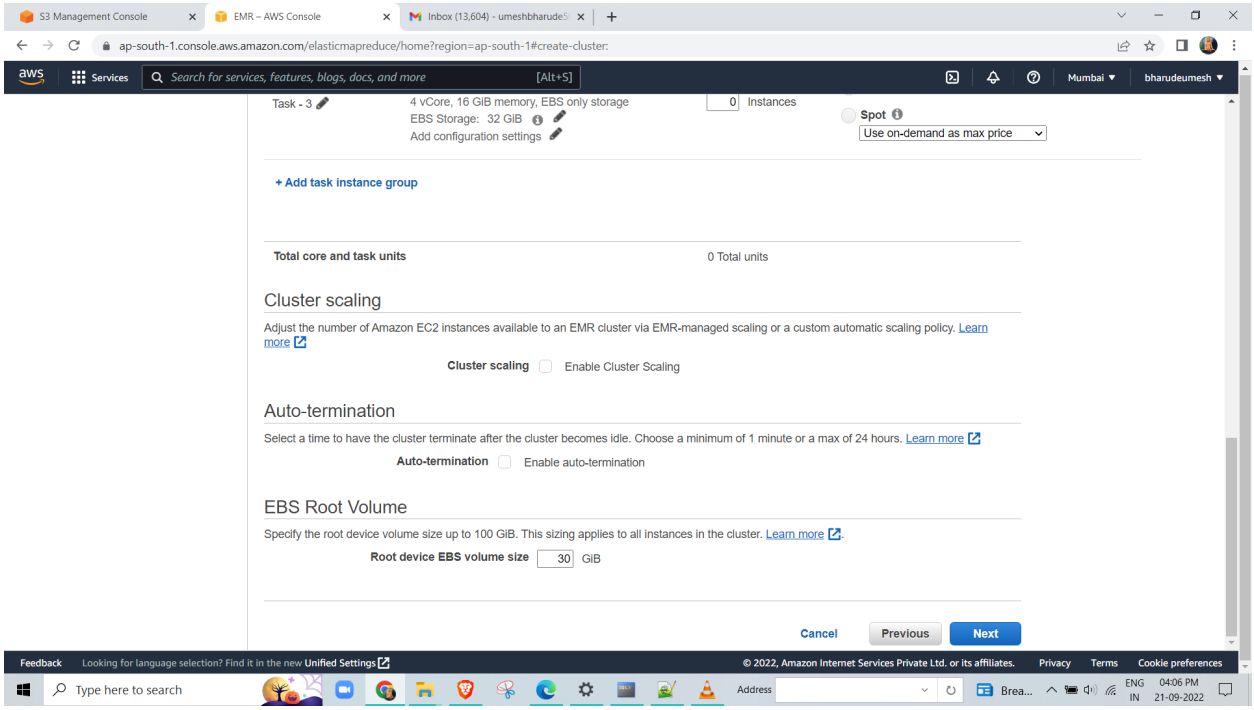
- We Run the above shell script
- The raw data is download and
- Then the raw data is store in s3 bucket
- We also store the shell script file in s3 bucket

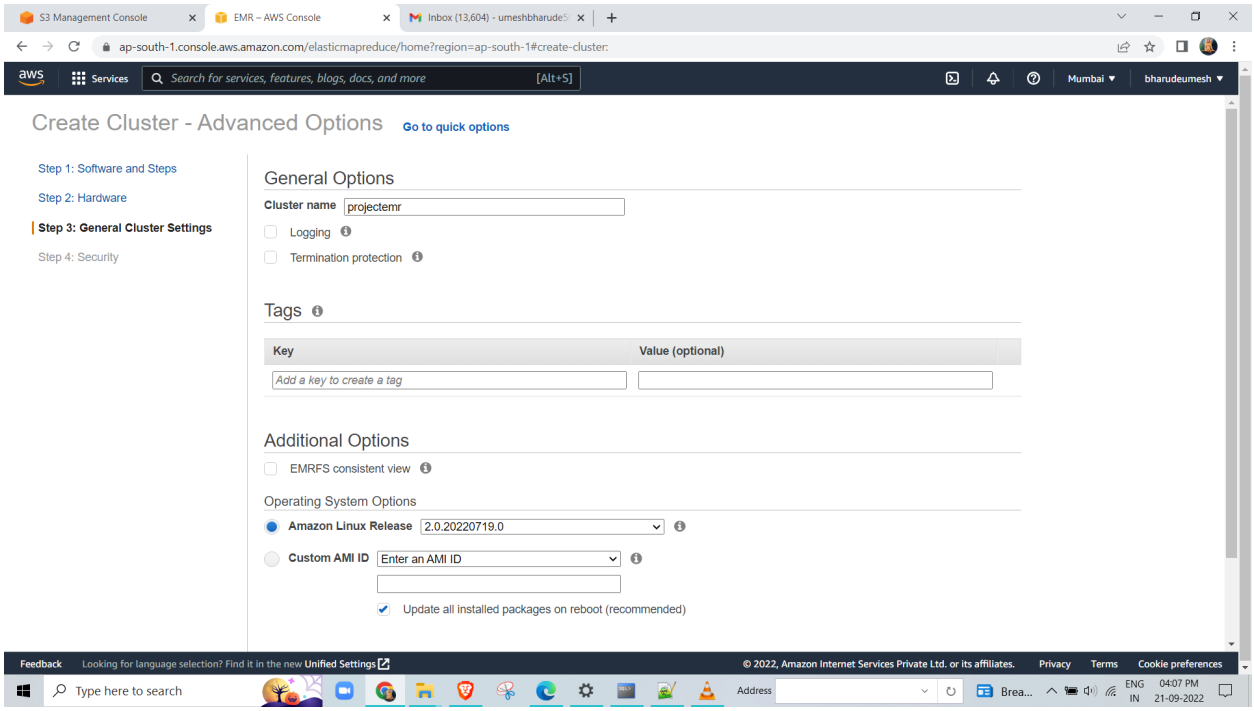


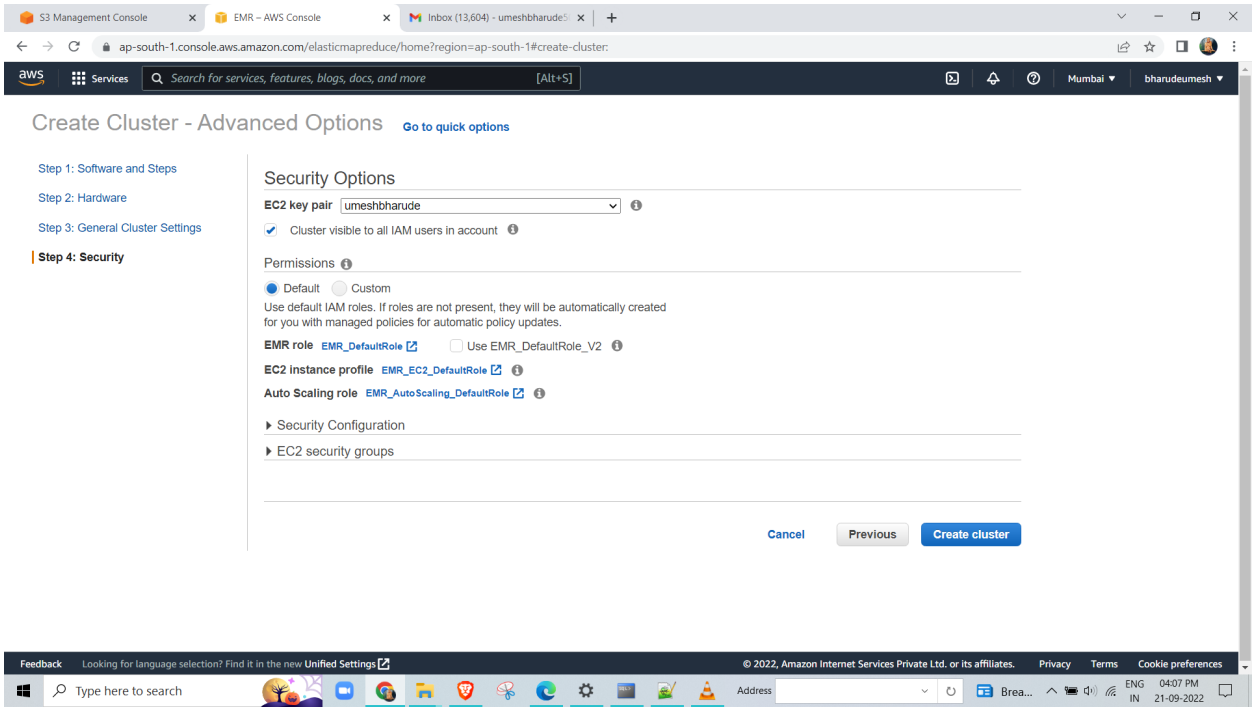
2)create the emr cluster and we choose services hadoop, spark and hive :

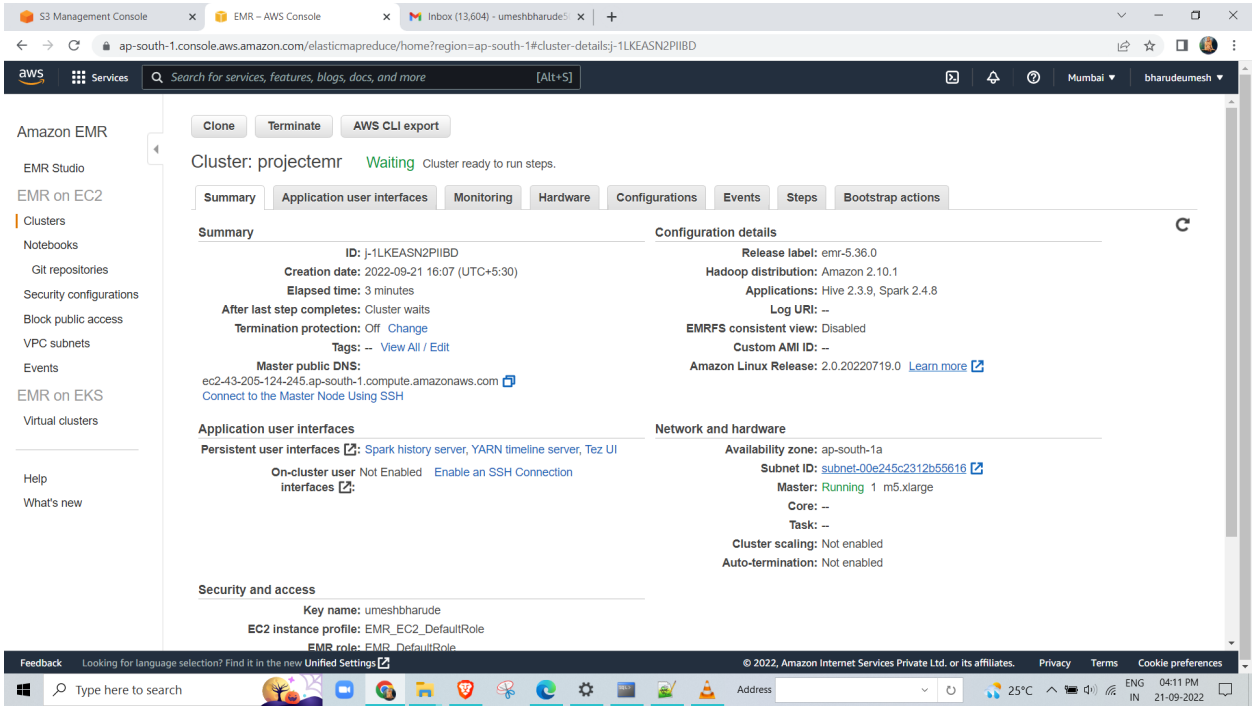


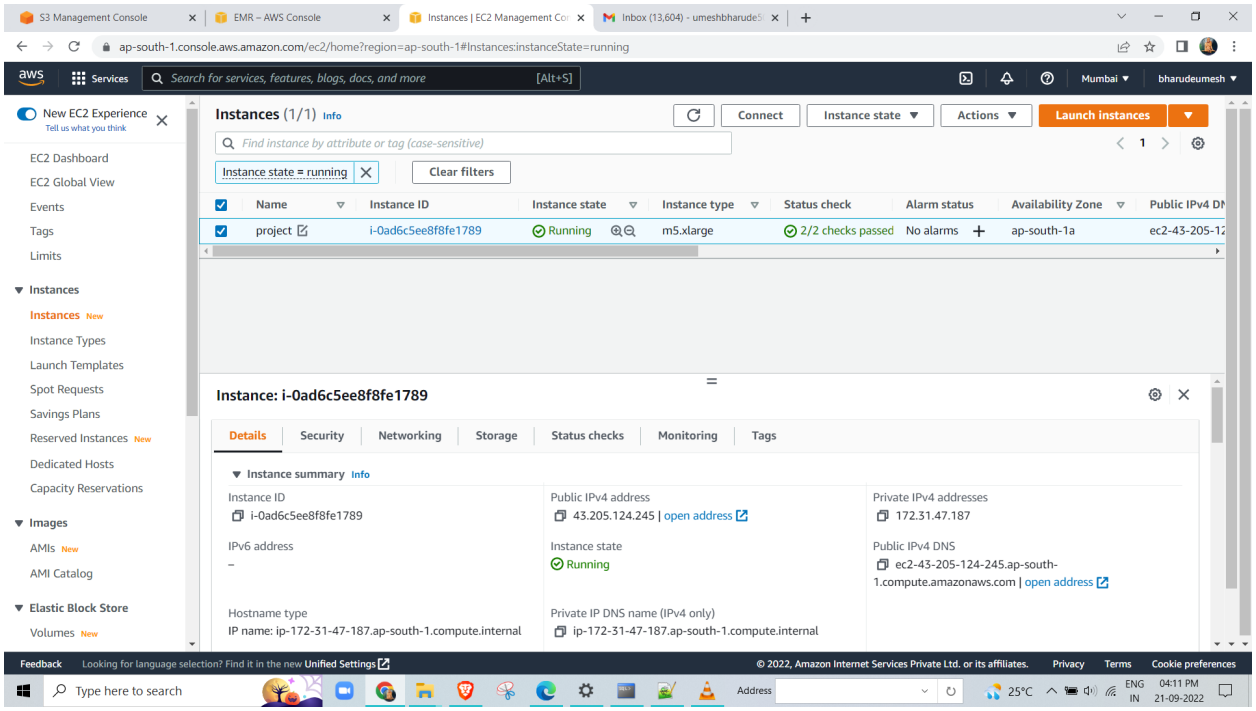


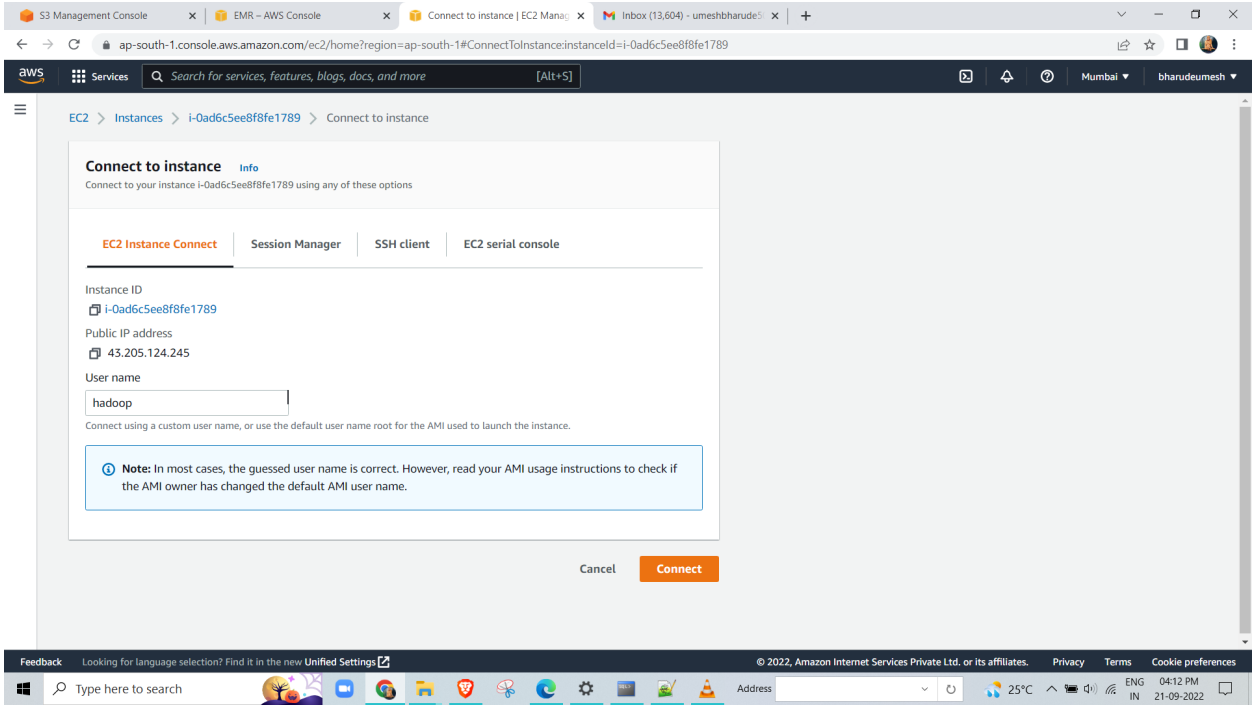

S3 Management Console
EMR – AWS Console
ap-south-1.console.aws.amazon.com/elasticmapreduce/home?region=ap-south-1#create-cluster:
Search for services, features, blogs, docs, and more [Alt+S]
Mumbai
bharudeumesh
Task - 3
4 vCore, 16 GiB memory, EBS only storage
EBS Storage: 32 GiB
Add configuration settings
0 Instances
Spot
Use on-demand as max price
+ Add task instance group
Total core and task units
0 Total units
Cluster scaling
Adjust the number of Amazon EC2 instances available to an EMR cluster via EMR-managed scaling or a custom automatic scaling policy. Learn more
Cluster scaling
Enable Cluster Scaling
Auto-termination
Select a time to have the cluster terminate after the cluster becomes idle. Choose a minimum of 1 minute or a max of 24 hours. Learn more
Auto-termination
Enable auto-termination
EBS Root Volume
Specify the root device volume size up to 100 GiB. This sizing applies to all instances in the cluster. Learn more.
Root device EBS volume size
30
GiB
Cancel
Previous
Next
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
Address
ENG
IN
04:06 PM
21-09-2022

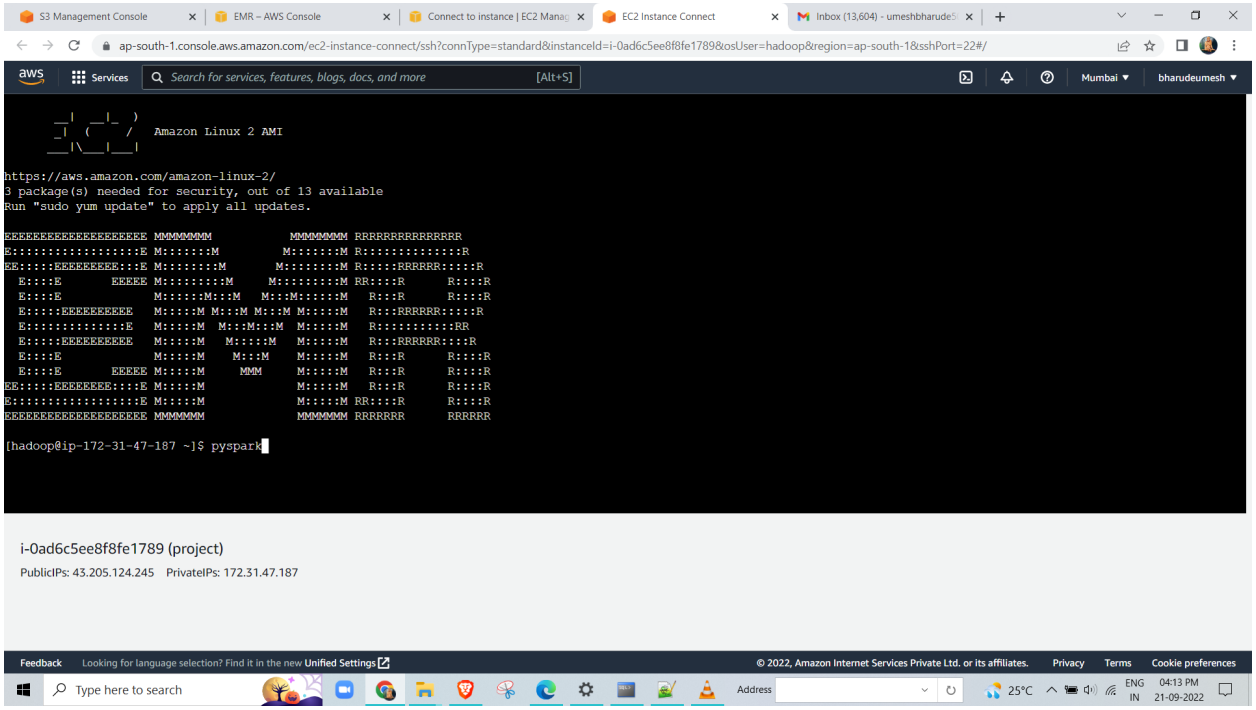

S3 Management Console
EMR – AWS Console
ap-south-1.console.aws.amazon.com/elasticmapreduce/home?region=ap-south-1#create-cluster:
Search for services, features, blogs, docs, and more [Alt+S]
Mumbai
bharudeumesh
Create Cluster - Advanced Options
Go to quick options
Step 1: Software and Steps
Step 2: Hardware
Step 3: General Cluster Settings
Step 4: Security
General Options
Cluster name
projectemr
Logging
Termination protection
Tags
Key
Value (optional)
Add a key to create a tag
Additional Options
EMRFS consistent view
Operating System Options
Amazon Linux Release
2.0.20220719.0
Custom AMI ID
Enter an AMI ID
Update all installed packages on reboot (recommended)
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
Address
ENG
IN
04:07 PM
21-09-2022


S3 Management Console
EMR – AWS Console
Inbox (13,604) - umeshbharude5
ap-south-1.console.aws.amazon.com/elasticmapreduce/home?region=ap-south-1#create-cluster:
Search for services, features, blogs, docs, and more [Alt+S]
Mumbai
bharudeumesh
Create Cluster - Advanced Options
Go to quick options
Step 1: Software and Steps
Step 2: Hardware
Step 3: General Cluster Settings
Step 4: Security
Security Options
EC2 key pair umeshbharude
Cluster visible to all IAM users in account
Permissions
Default
Custom
Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.
EMR role EMR_DefaultRole
Use EMR_DefaultRole_V2
EC2 instance profile EMR_EC2_DefaultRole
Auto Scaling role EMR_AutoScaling_DefaultRole
Security Configuration
EC2 security groups
Cancel
Previous
Create cluster
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
Address
ENG IN 04:07 PM 21-09-2022


S3 Management Console
EMR – AWS Console
Inbox (13,604) - umeshbharude5
ap-south-1.console.aws.amazon.com/elasticmapreduce/home?region=ap-south-1#cluster-details:j-1LKEASN2PIIBD
Search for services, features, blogs, docs, and more [Alt+S]
Mumbai
bharudeumesh
Amazon EMR
EMR Studio
EMR on EC2
Clusters
Notebooks
Git repositories
Security configurations
Block public access
VPC subnets
Events
EMR on EKS
Virtual clusters
Help
What's new
Clone
Terminate
AWS CLI export
Cluster: projectemr
Waiting
Cluster ready to run steps.
Summary
Application user interfaces
Monitoring
Hardware
Configurations
Events
Steps
Bootstrap actions
Summary
ID: j-1LKEASN2PIIBD
Creation date: 2022-09-21 16:07 (UTC+5:30)
Elapsed time: 3 minutes
After last step completes: Cluster waits
Termination protection: Off Change
Tags: -- View All / Edit
Master public DNS:
ec2-43-205-124-245.ap-south-1.compute.amazonaws.com
Connect to the Master Node Using SSH
Configuration details
Release label: emr-5.36.0
Hadoop distribution: Amazon 2.10.1
Applications: Hive 2.3.9, Spark 2.4.8
Log URI: --
EMRFS consistent view: Disabled
Custom AMI ID: --
Amazon Linux Release: 2.0.20220719.0 Learn more
Application user interfaces
Persistent user interfaces: Spark history server, YARN timeline server, Tez UI
On-cluster user interfaces: Not Enabled Enable an SSH Connection
Network and hardware
Availability zone: ap-south-1a
Subnet ID: subnet-00e245c2312b55616
Master: Running 1 m5.xlarge
Core: --
Task: --
Cluster scaling: Not enabled
Auto-termination: Not enabled
Security and access
Key name: umeshbharude
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
Address
25°C
ENG IN 04:11 PM 21-09-2022


S3 Management Console
EMR – AWS Console
Instances | EC2 Management Cor
Inbox (13,604) - umeshbharude5
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Instances:instanceState=running
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Mumbai
bharudeumesh
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Capacity Reservations
Images
AMIs New
AMI Catalog
Elastic Block Store
Volumes New
Instances (1/1) Info
Connect
Instance state
Actions
Launch instances
Find instance by attribute or tag (case-sensitive)
Instance state = running
Clear filters
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
project
i-0ad6c5ee8f8fe1789
Running
m5.xlarge
2/2 checks passed
No alarms
ap-south-1a
Instance: i-0ad6c5ee8f8fe1789
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance ID
i-0ad6c5ee8f8fe1789
Public IPv4 address
43.205.124.245 | open address
Private IPv4 addresses
172.31.47.187
IPv6 address
–
Instance state
Running
Public IPv4 DNS
ec2-43-205-124-245.ap-south-1.compute.amazonaws.com | open address
Hostname type
IP name: ip-172-31-47-187.ap-south-1.compute.internal
Private IP DNS name (IPv4 only)
ip-172-31-47-187.ap-south-1.compute.internal
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
Address
25°C
ENG
IN
04:11 PM
21-09-2022


S3 Management Console
EMR – AWS Console
Connect to instance | EC2 Manag
Inbox (13,604) - umeshbharude5
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#ConnectToInstance:instanceId=i-0ad6c5ee8f8fe1789
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Mumbai
bharudeumesh
EC2 > Instances > i-0ad6c5ee8f8fe1789 > Connect to instance
Connect to instance Info
Connect to your instance i-0ad6c5ee8f8fe1789 using any of these options
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Instance ID
i-0ad6c5ee8f8fe1789
Public IP address
43.205.124.245
User name
hadoop
Connect using a custom user name, or use the default user name root for the AMI used to launch the instance.
Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.
Cancel
Connect
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
Address
25°C
ENG
IN
04:12 PM
21-09-2022

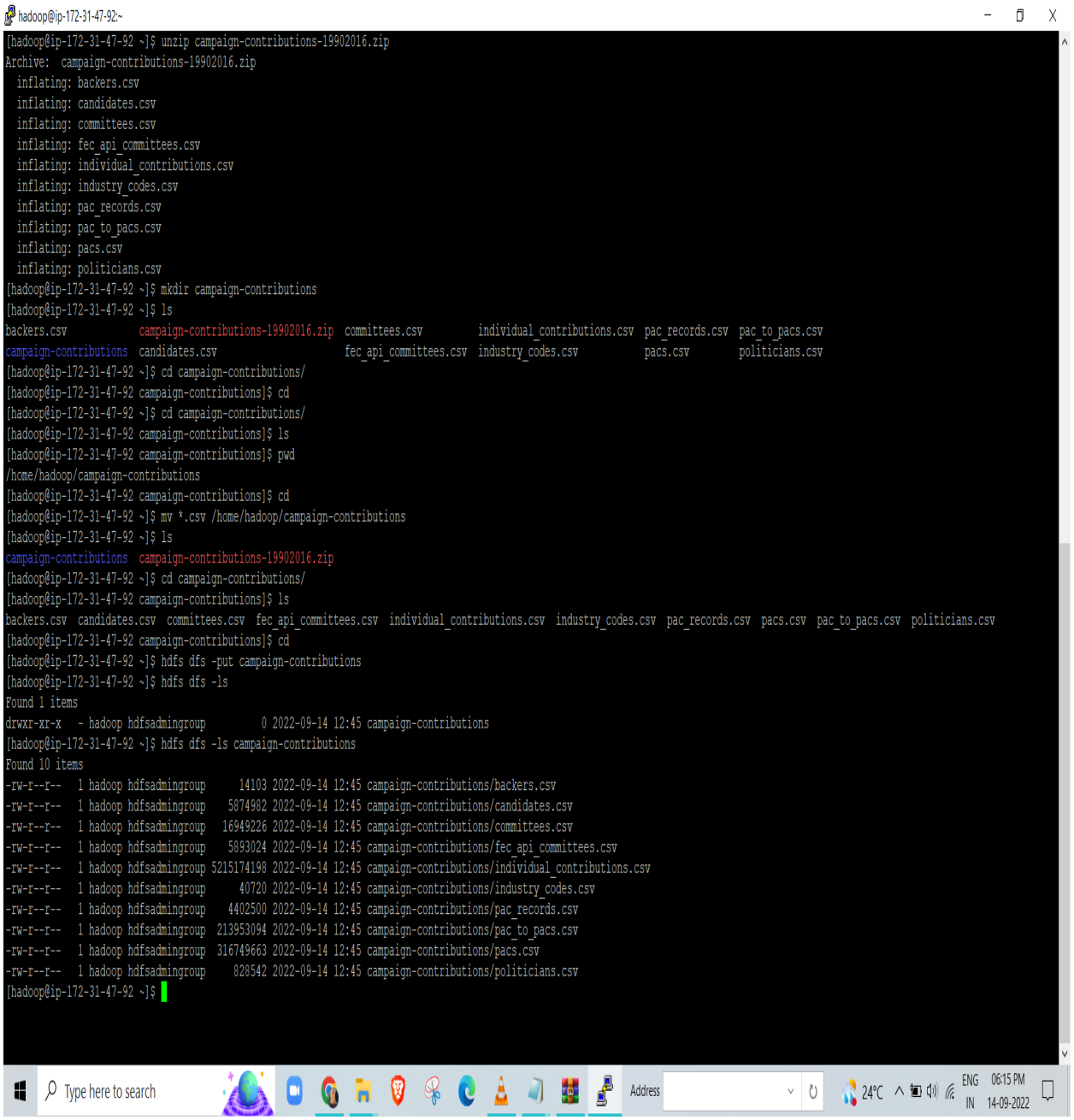

S3 Management Console
EMR – AWS Console
Connect to instance | EC2 Manag
EC2 Instance Connect
Inbox (13,604) - umeshbharude5
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0ad6c5ee8f8fe1789&osUser=hadoop&region=ap-south-1&sshPort=22#/
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Mumbai
bharudeumesh
Amazon Linux 2 AMI
https://aws.amazon.com/amazon-linux-2/
3 package(s) needed for security, out of 13 available
Run "sudo yum update" to apply all updates.
[hadoop@ip-172-31-47-187 ~]$ pyspark
i-0ad6c5ee8f8fe1789 (project)
PublicIPs: 43.205.124.245 PrivateIPs: 172.31.47.187
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
Address
25°C
ENG
IN
04:13 PM
21-09-2022

3)we copy the data from s3 bucket to emr:

By using the command

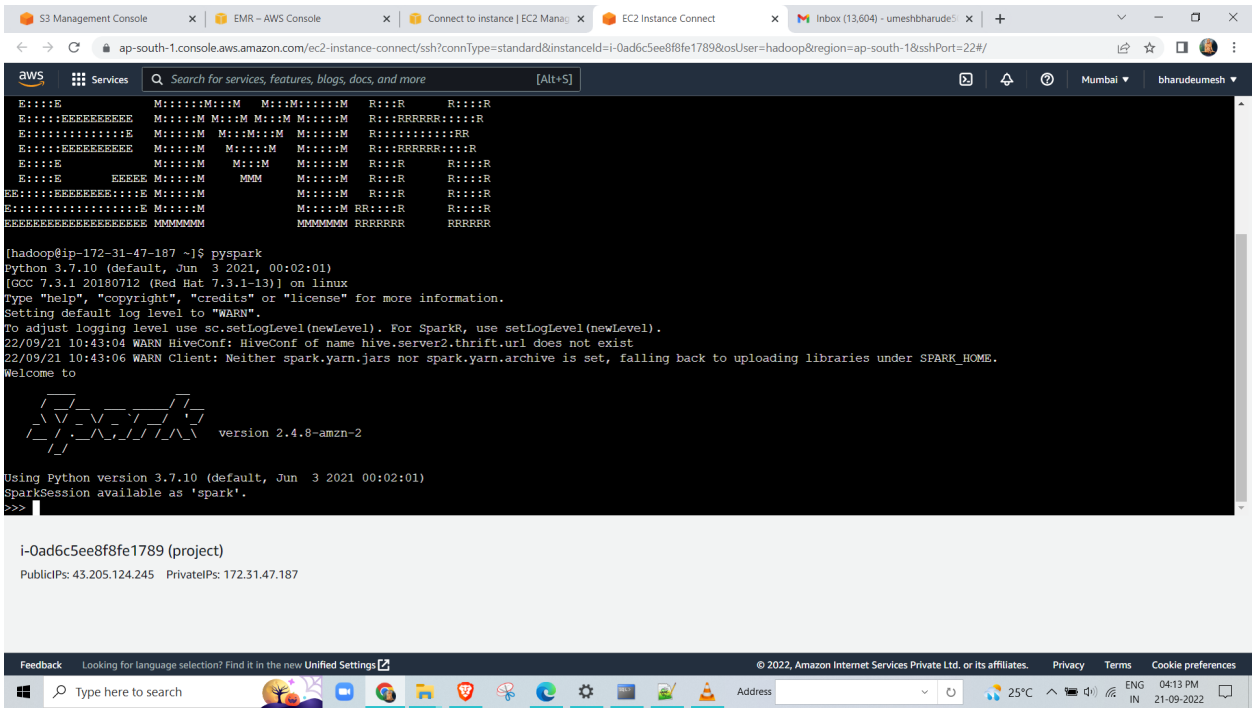
- aws s3 <path of bucket> ./
- Then we unzip the data and store data in hdfs by using command :

  $ hdfs dfs -put <directory name >

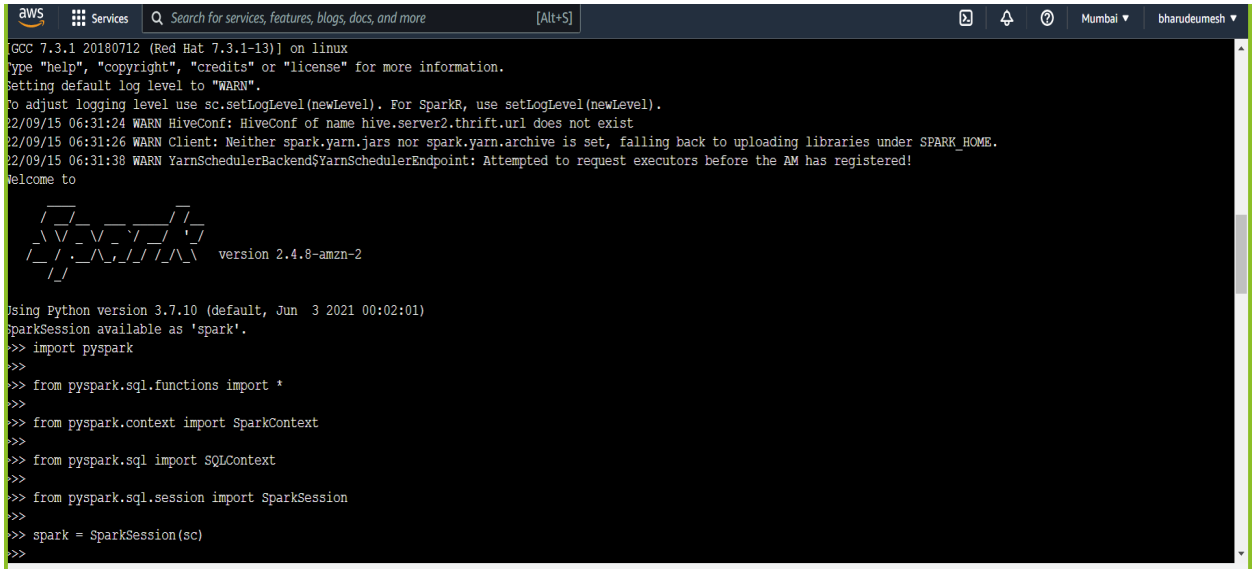


4)Go to the pyspark on HDFS

Command ===> pyspark
IACSD-PG-DBDA-MARCH-22



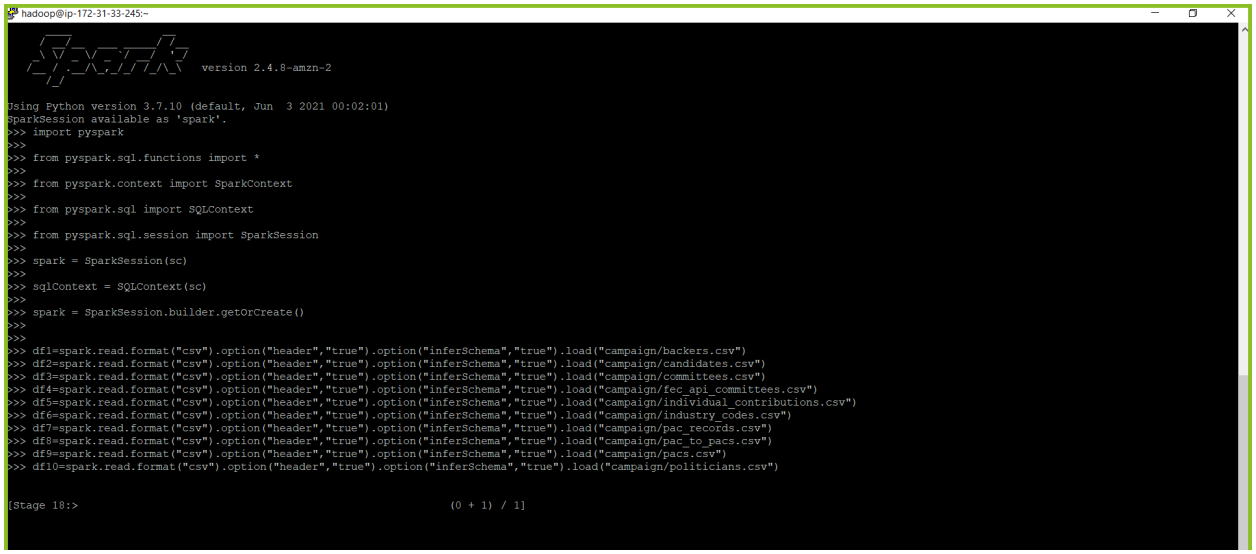
5) we install the libraries in pyspark:



6)Read multiple csv from hdfs to spark

##creating dataframe :

```
df1=spark.read.format("csv").option("header","true").option("inferSchema","true"
).load("campaign/backers.csv")
df2=spark.read.format("csv").option("header","true").option("inferSchema","true"
).load("campaign/candidates.csv")
df3=spark.read.format("csv").option("header","true").option("inferSchema","true"
).load("campaign/committees.csv")
df4=spark.read.format("csv").option("header","true").option("inferSchema","true"
).load("campaign/fec_api_committees.csv")
df5=spark.read.format("csv").option("header","true").option("inferSchema","true"
).load("campaign/individual_contributions.csv")
df6=spark.read.format("csv").option("header","true").option("inferSchema","true"
).load("campaign/industry_codes.csv")
df7=spark.read.format("csv").option("header","true").option("inferSchema","true"
).load("campaign/pac_records.csv")
df8=spark.read.format("csv").option("header","true").option("inferSchema","true"
).load("campaign/pac_to_pacs.csv")
df9=spark.read.format("csv").option("header","true").option("inferSchema","true"
).load("campaign/pacs.csv")
df10=spark.read.format("csv").option("header","true").option("inferSchema","true
").load("campaign/politicians.csv")
```

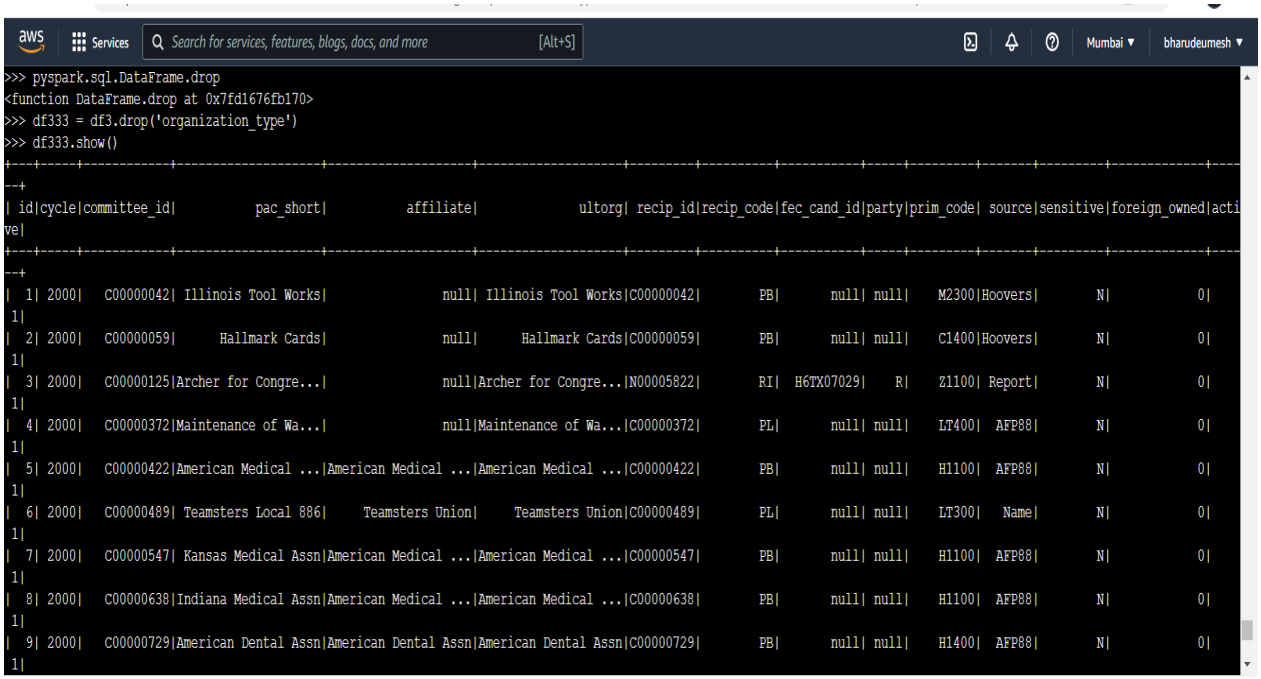


##creating temp view of tables:

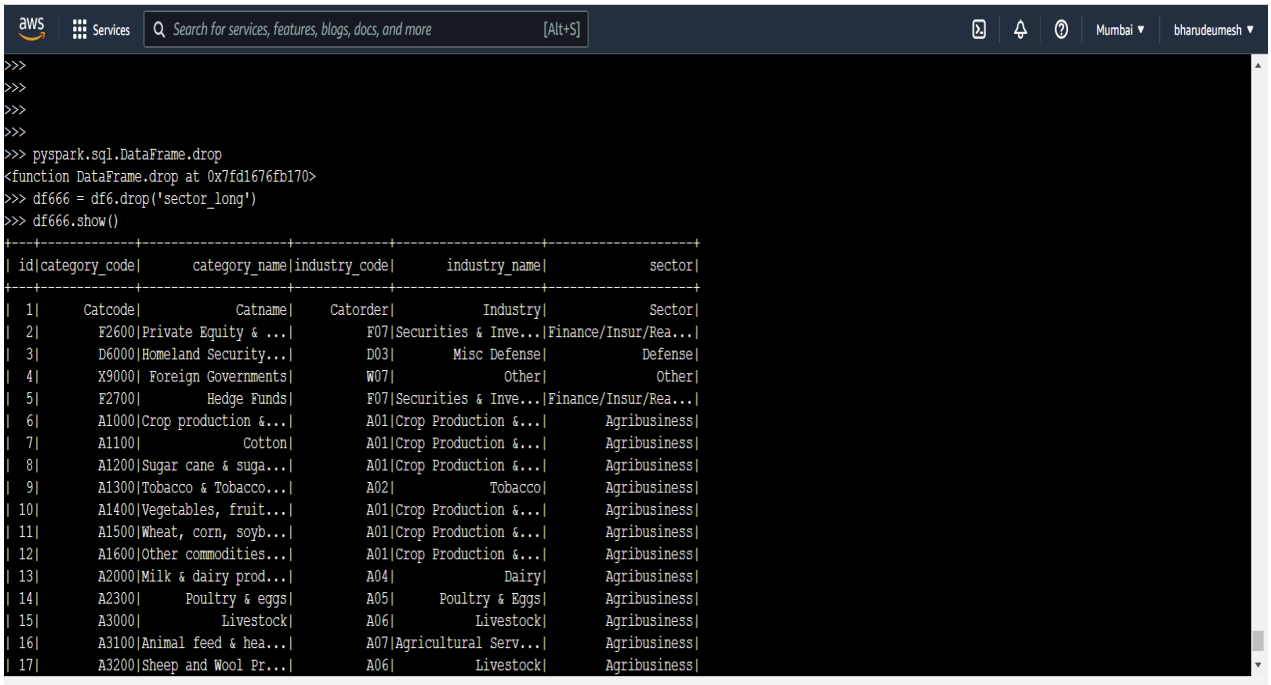
```
>>>
>>>
>>> df1.createOrReplaceTempView("backers")
>>> df2.createOrReplaceTempView("candidates")
>>> df3.createOrReplaceTempView("committees")
>>> df4.createOrReplaceTempView("fec_api_committees")
>>> df5.createOrReplaceTempView("individual_contributions")
>>> df6.createOrReplaceTempView("industry_codes")
>>> df7.createOrReplaceTempView("pac_records")
>>> df8.createOrReplaceTempView("pac_to_pacs")
>>> df9.createOrReplaceTempView("pacs")
>>> df10.createOrReplaceTempView("politicians")
>>>
```

```
>>>
>>> df1=spark.read.format("csv").option("header","true").option("inferSchema","true").load("campaign/backers.csv")
>>>
>>>
>>> df1.createOrReplaceTempView("backers")
>>>
>>> spark.sql("select * from backers").show(10)
+---+---------+--------------------+------------+-----------+
| id|      cid|                name|backer_level|kickstarter|
+---+---------+--------------------+------------+-----------+
|  1|N00003389|         Karan Dodia|  leadership|          t|
|  2|N00009922|          Neal Foard|  leadership|          t|
|  3|N00000528|          Kevin Pine|      senate|          t|
|  4|N00001093|        Adam Edwards|      senate|          t|
|  5|N00007364|"Michael ""Pinski...|      senate|          t|
|  6|N00007364|    Trevor von Stein|      senate|          t|
|  7|N00027605|         Susan Estep|      senate|          t|
|  8|N00002097|    Wesley W. Driver|      senate|          t|
|  9|N00029303|       Betsy Wentzel|      senate|          t|
| 10|N00034580|           Chuck Dye|      senate|          t|
+---+---------+--------------------+------------+-----------+
only showing top 10 rows
```

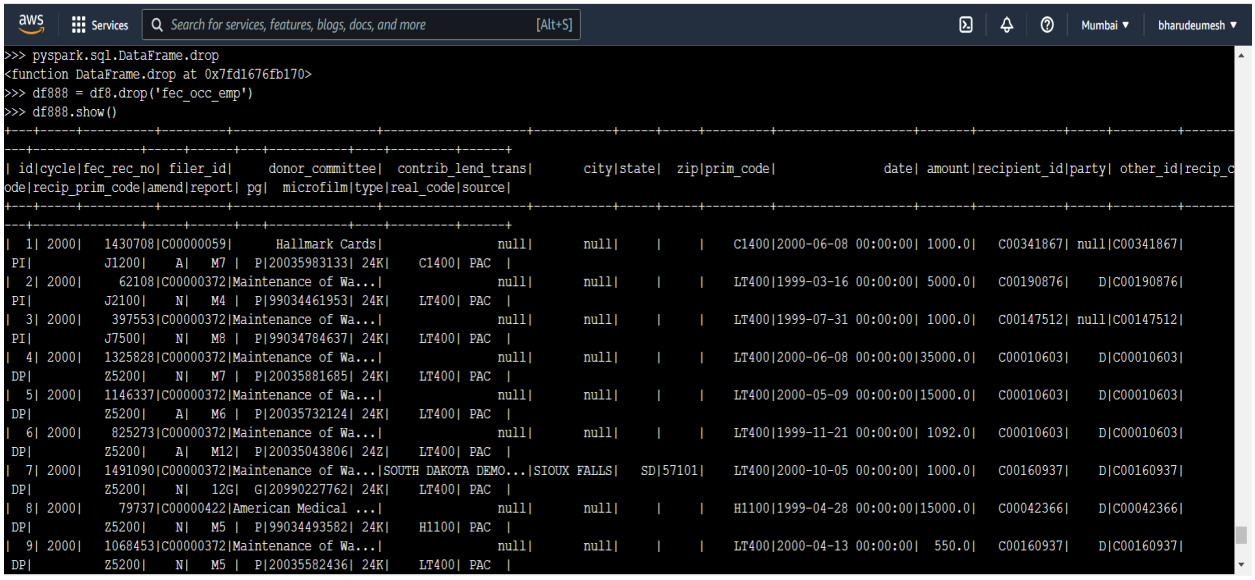
7) we clean the data ,we remove the null column:



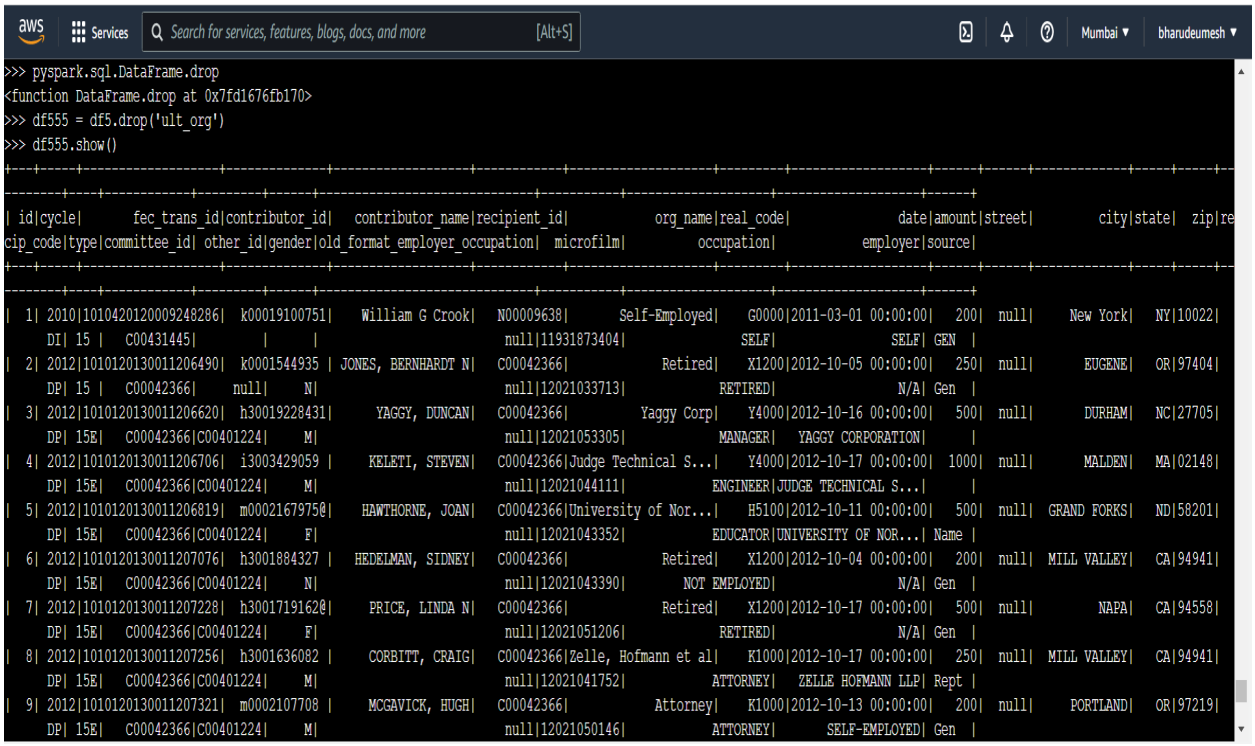
```
>>> pyspark.sql.DataFrame.drop
<function DataFrame.drop at 0x7fd1676fb170>
>>> df333 = df3.drop('organization_type')
>>> df333.show()
+---+-----+------------+--------------------+--------------------+--------------------+---------+----------+-----------+-----+---------+-------+---------+------------+----
--+
| id|cycle|committee_id|          pac_short|          affiliate|             ultorg| recip_id|recip_code|fec_cand_id|party|prim_code| source|sensitive|foreign_owned|acti
ve|
+---+-----+------------+--------------------+--------------------+--------------------+---------+----------+-----------+-----+---------+-------+---------+------------+----
--+
|  1| 2000|   C00000042| Illinois Tool Works|                null| Illinois Tool Works|C00000042|       PB|       null| null|    M2300|Hoovers|        N|           0|
 1|
|  2| 2000|   C00000059|      Hallmark Cards|                null|      Hallmark Cards|C00000059|       PB|       null| null|    C1400|Hoovers|        N|           0|
 1|
|  3| 2000|   C00000125|Archer for Congre...|                null|Archer for Congre...|N00005822|       RI|  H6TX07029|    R|    Z1100| Report|        N|           0|
 1|
|  4| 2000|   C00000372|Maintenance of Wa...|                null|Maintenance of Wa...|C00000372|       PL|       null| null|    LT400|  AFP88|        N|           0|
 1|
|  5| 2000|   C00000422|American Medical ...|American Medical ...|American Medical ...|C00000422|       PB|       null| null|    H1100|  AFP88|        N|           0|
 1|
|  6| 2000|   C00000489| Teamsters Local 886|     Teamsters Union|     Teamsters Union|C00000489|       PL|       null| null|    LT300|   Name|        N|           0|
 1|
|  7| 2000|   C00000547| Kansas Medical Assn|American Medical ...|American Medical ...|C00000547|       PB|       null| null|    H1100|  AFP88|        N|           0|
 1|
|  8| 2000|   C00000638|Indiana Medical Assn|American Medical ...|American Medical ...|C00000638|       PB|       null| null|    H1100|  AFP88|        N|           0|
 1|
|  9| 2000|   C00000729|American Dental Assn|American Dental Assn|American Dental Assn|C00000729|       PB|       null| null|    H1400|  AFP88|        N|           0|
 1|
```



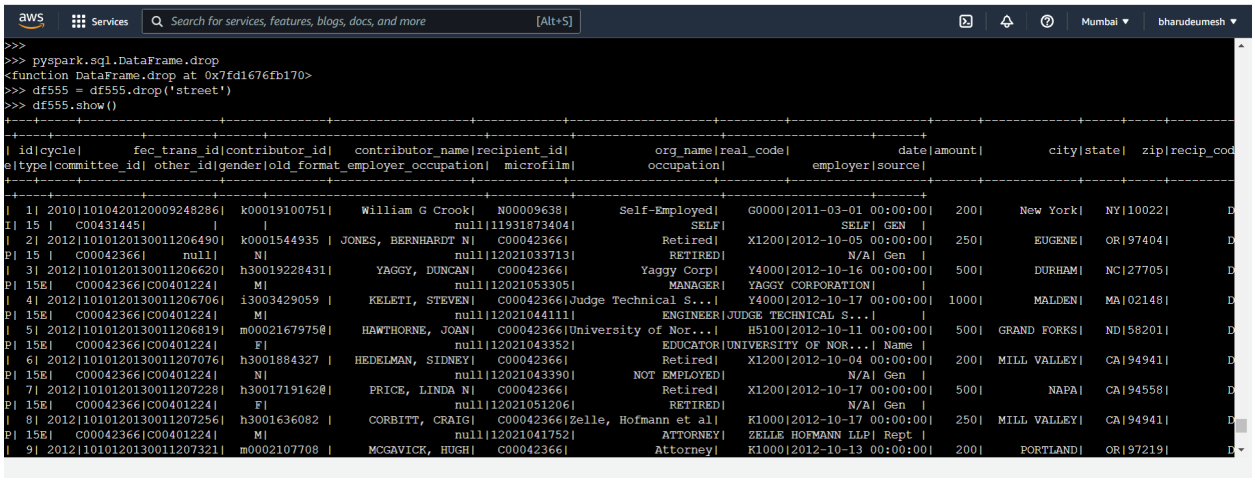
```
>>>
>>>
>>>
>>>
>>> pyspark.sql.DataFrame.drop
<function DataFrame.drop at 0x7fd1676fb170>
>>> df666 = df6.drop('sector_long')
>>> df666.show()
+---+------------+--------------------+------------+--------------------+--------------------+
| id|category_code|       category_name|industry_code|       industry_name|              sector|
+---+------------+--------------------+------------+--------------------+--------------------+
|  1|     Catcode|             Catname|    Catorder|            Industry|              Sector|
|  2|       F2600|Private Equity & ...|         F07|Securities & Inve...|Finance/Insur/Rea...|
|  3|       D6000|Homeland Security...|         D03|        Misc Defense|             Defense|
|  4|       X9000| Foreign Governments|         W07|               Other|               Other|
|  5|       F2700|         Hedge Funds|         F07|Securities & Inve...|Finance/Insur/Rea...|
|  6|       A1000|Crop production &...|         A01|Crop Production &...|        Agribusiness|
|  7|       A1100|              Cotton|         A01|Crop Production &...|        Agribusiness|
|  8|       A1200|Sugar cane & suga...|         A01|Crop Production &...|        Agribusiness|
|  9|       A1300|Tobacco & Tobacco...|         A02|             Tobacco|        Agribusiness|
| 10|       A1400|Vegetables, fruit...|         A01|Crop Production &...|        Agribusiness|
| 11|       A1500|Wheat, corn, soyb...|         A01|Crop Production &...|        Agribusiness|
| 12|       A1600|Other commodities...|         A01|Crop Production &...|        Agribusiness|
| 13|       A2000|Milk & dairy prod...|         A04|               Dairy|        Agribusiness|
| 14|       A2300|      Poultry & eggs|         A05|      Poultry & Eggs|        Agribusiness|
| 15|       A3000|           Livestock|         A06|           Livestock|        Agribusiness|
| 16|       A3100|Animal feed & hea...|         A07|Agricultural Serv...|        Agribusiness|
| 17|       A3200|Sheep and Wool Pr...|         A06|           Livestock|        Agribusiness|
```



```
>>> pyspark.sql.DataFrame.drop
<function DataFrame.drop at 0x7fd1676fb170>
>>> df888 = df8.drop('fec_occ_emp')
>>> df888.show()
+---+-----+----------+---------+--------------------+--------------------+-----------+-----+-----+---------+------------------+-------+------------+-----+---------+-------
---+--------------+-----+------+---+----------+----+---------+------+
| id|cycle|fec_rec_no| filer_id|     donor_committee|  contrib_lend_trans|       city|state|  zip|prim_code|              date| amount|recipient_id|party| other_id|recip_c
ode|recip_prim_code|amend|report| pg|  microfilm|type|real_code|source|
+---+-----+----------+---------+--------------------+--------------------+-----------+-----+-----+---------+------------------+-------+------------+-----+---------+-------
---+--------------+-----+------+---+----------+----+---------+------+
|  1| 2000|   1430708|C00000059|      Hallmark Cards|                null|       null|     |     |    C1400|2000-06-08 00:00:00| 1000.0|   C00341867| null|C00341867|
 PI|         J1200|    A|   M7 |  P|20035983133| 24K|    C1400| PAC  |
|  2| 2000|     62108|C00000372|Maintenance of Wa...|                null|       null|     |     |    LT400|1999-03-16 00:00:00| 5000.0|   C00190876|    D|C00190876|
 PI|         J2100|    N|   M4 |  P|99034461953| 24K|    LT400| PAC  |
|  3| 2000|    397553|C00000372|Maintenance of Wa...|                null|       null|     |     |    LT400|1999-07-31 00:00:00| 1000.0|   C00147512| null|C00147512|
 PI|         J7500|    N|   M8 |  P|99034784637| 24K|    LT400| PAC  |
|  4| 2000|   1325828|C00000372|Maintenance of Wa...|                null|       null|     |     |    LT400|2000-06-08 00:00:00|35000.0|   C00010603|    D|C00010603|
 DP|         Z5200|    N|   M7 |  P|20035881685| 24K|    LT400| PAC  |
|  5| 2000|   1146337|C00000372|Maintenance of Wa...|                null|       null|     |     |    LT400|2000-05-09 00:00:00|15000.0|   C00010603|    D|C00010603|
 DP|         Z5200|    A|   M6 |  P|20035732124| 24K|    LT400| PAC  |
|  6| 2000|    825273|C00000372|Maintenance of Wa...|                null|       null|     |     |    LT400|1999-11-21 00:00:00| 1092.0|   C00010603|    D|C00010603|
 DP|         Z5200|    A|   M12|  P|20035043806| 24Z|    LT400| PAC  |
|  7| 2000|   1491090|C00000372|Maintenance of Wa...|SOUTH DAKOTA DEMO...|SIOUX FALLS|   SD|57101|    LT400|2000-10-05 00:00:00| 1000.0|   C00160937|    D|C00160937|
 DP|         Z5200|    N|   12G|  G|20990227762| 24K|    LT400| PAC  |
|  8| 2000|     79737|C00000422|American Medical ...|                null|       null|     |     |    H1100|1999-04-28 00:00:00|15000.0|   C00042366|    D|C00042366|
 DP|         Z5200|    N|   M5 |  P|99034493582| 24K|    H1100| PAC  |
|  9| 2000|   1068453|C00000372|Maintenance of Wa...|                null|       null|     |     |    LT400|2000-04-13 00:00:00|  550.0|   C00160937|    D|C00160937|
 DP|         Z5200|    N|   M5 |  P|20035582436| 24K|    LT400| PAC  |
```

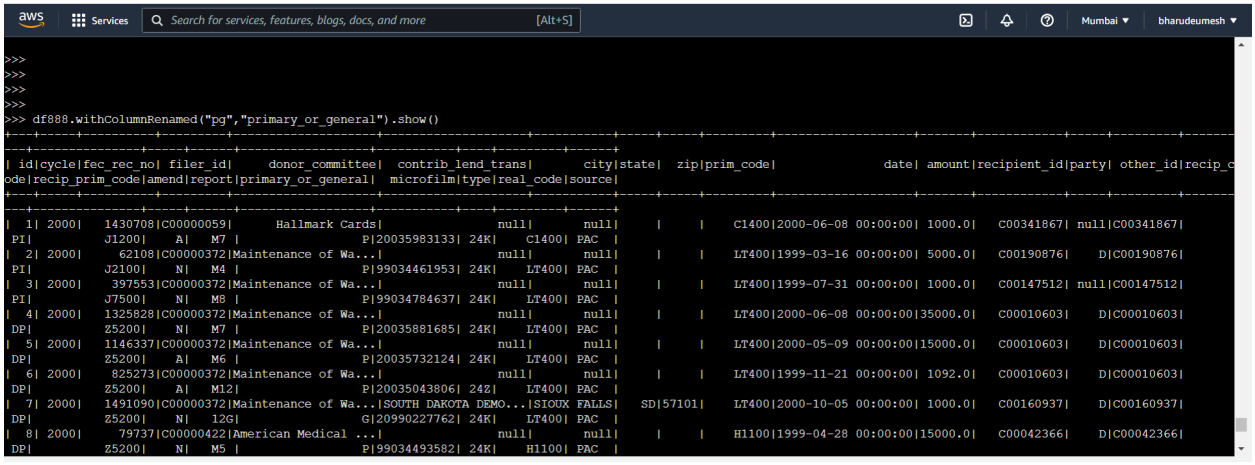


```
>>> pyspark.sql.DataFrame.drop
<function DataFrame.drop at 0x7fd1676fb170>
>>> df555 = df5.drop('ult_org')
>>> df555.show()
+---+-----+------------------+-------------+------------------+-----------+------------------+--------+------------------+------+-----+------------+-----+-----+--
--------+----+-----------+--------+------+--------------------------+----------+------------------+------------------+------+
| id|cycle|      fec_trans_id|contributor_id|  contributor_name|recipient_id|          org_name|real_code|              date|amount|street|        city|state|  zip|re
cip_code|type|committee_id| other_id|gender|old_format_employer_occupation| microfilm|        occupation|          employer|source|
+---+-----+------------------+-------------+------------------+-----------+------------------+--------+------------------+------+-----+------------+-----+-----+--
--------+----+-----------+--------+------+--------------------------+----------+------------------+------------------+------+
|  1| 2010|10104201200092482 86|  k00019100751|    William G Crook|  N00009638|      Self-Employed|    G0000|2011-03-01 00:00:00|   200|  null|    New York|   NY|10022|
     DI| 15 |   C00431445|         |      |                          |  null|11931873404|              SELF|              SELF| GEN  |
|  2| 2012|10101201300112064 90|  k0001544935 | JONES, BERNHARDT N|  C00042366|           Retired|    X1200|2012-10-05 00:00:00|   250|  null|      EUGENE|   OR|97404|
     DP| 15 |   C00042366|     null|     N|                          |  null|12021033713|           RETIRED|               N/A| Gen  |
|  3| 2012|10101201300112066 20|  h30019228431|      YAGGY, DUNCAN|  C00042366|         Yaggy Corp|    Y4000|2012-10-16 00:00:00|   500|  null|      DURHAM|   NC|27705|
     DP| 15E|   C00042366|C00401224|     M|                          |  null|12021053305|           MANAGER|   YAGGY CORPORATION|      |
|  4| 2012|10101201300112067 06|  i3003429059 |     KELETI, STEVEN|  C00042366|Judge Technical S...|    Y4000|2012-10-17 00:00:00|  1000|  null|      MALDEN|   MA|02148|
     DP| 15E|   C00042366|C00401224|     M|                          |  null|12021044111|          ENGINEER|JUDGE TECHNICAL S...|      |
|  5| 2012|10101201300112068 19|  m0002167975@|    HAWTHORNE, JOAN|  C00042366|University of Nor...|    H5100|2012-10-11 00:00:00|   500|  null|  GRAND FORKS|   ND|58201|
     DP| 15E|   C00042366|C00401224|     F|                          |  null|12021043352|          EDUCATOR|UNIVERSITY OF NOR...| Name |
|  6| 2012|10101201300112070 76|  h3001884327 |   HEDELMAN, SIDNEY|  C00042366|           Retired|    X1200|2012-10-04 00:00:00|   200|  null|  MILL VALLEY|   CA|94941|
     DP| 15E|   C00042366|C00401224|     N|                          |  null|12021043390|       NOT EMPLOYED|               N/A| Gen  |
|  7| 2012|10101201300112072 28|  h3001719162@|     PRICE, LINDA N|  C00042366|           Retired|    X1200|2012-10-17 00:00:00|   500|  null|        NAPA|   CA|94558|
     DP| 15E|   C00042366|C00401224|     F|                          |  null|12021051206|           RETIRED|               N/A| Gen  |
|  8| 2012|10101201300112072 56|  h3001636082 |     CORBITT, CRAIG|  C00042366|Zelle, Hofmann et al|    K1000|2012-10-17 00:00:00|   250|  null|  MILL VALLEY|   CA|94941|
     DP| 15E|   C00042366|C00401224|     M|                          |  null|12021041752|          ATTORNEY|   ZELLE HOFMANN LLP| Rept |
|  9| 2012|10101201300112073 21|  m0002107708 |     MCGAVICK, HUGH|  C00042366|          Attorney|    K1000|2012-10-13 00:00:00|   200|  null|     PORTLAND|   OR|97219|
     DP| 15E|   C00042366|C00401224|     M|                          |  null|12021050146|          ATTORNEY|      SELF-EMPLOYED| Gen  |
```



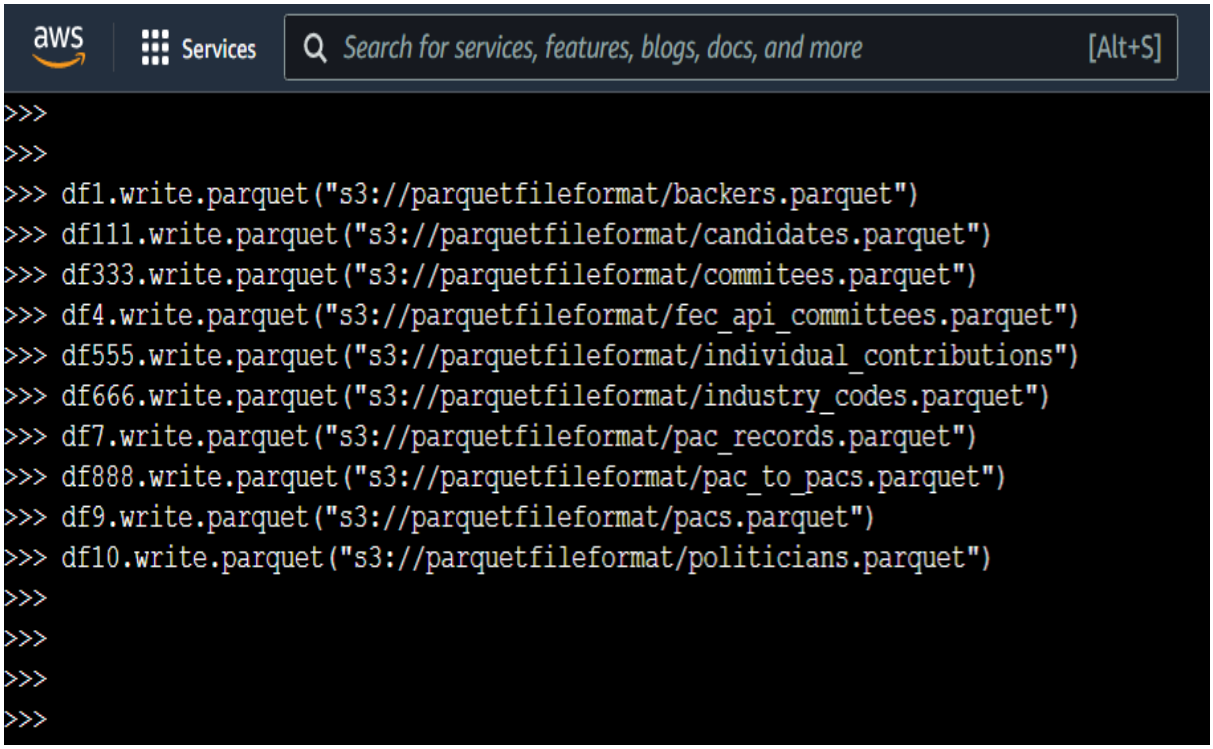

```
>>>
>>> pyspark.sql.DataFrame.drop
<function DataFrame.drop at 0x7fd1676fb170>
>>> df555 = df555.drop('street')
>>> df555.show()
+---+-----+-----------------+--------------+--------------------+------------+--------------------+---------+--------------------+------+-------------+-----+-----+---------
-+----+------------+---------+------+----------------------------+-----------+--------------------+--------------------+------+
| id|cycle|      fec_trans_id|contributor_id|   contributor_name|recipient_id|           org_name|real_code|              date|amount|         city|state|  zip|recip_cod
e|type|committee_id| other_id|gender|old_format_employer_occupation|  microfilm|          occupation|            employer|source|
+---+-----+-----------------+--------------+--------------------+------------+--------------------+---------+--------------------+------+-------------+-----+-----+---------
-+----+------------+---------+------+----------------------------+-----------+--------------------+--------------------+------+
|  1| 2010|10104201200092482861  k00019100751|    William G Crook|   N00009638|       Self-Employed|    G0000|2011-03-01 00:00:00|   200|     New York|   NY|10022|        D
I| 15 |   C00431445|         |      |                            |       null|11931873404|                SELF|                SELF| GEN  |
|  2| 2012|10101201300112064901  k0001544935 | JONES, BERNHARDT N|   C00042366|            Retired|    X1200|2012-10-05 00:00:00|   250|       EUGENE|   OR|97404|        D
P| 15 |   C00042366|     null|     N|                            |       null|12021033713|             RETIRED|                 N/A| Gen  |
|  3| 2012|10101201300112066201  h30019228431|      YAGGY, DUNCAN|   C00042366|          Yaggy Corp|    Y4000|2012-10-16 00:00:00|   500|       DURHAM|   NC|27705|        D
P| 15E|   C00042366|C00401224|     M|                            |       null|12021053305|            MANAGER|   YAGGY CORPORATION|      |
|  4| 2012|10101201300112067061  i3003429059 |     KELETI, STEVEN|   C00042366|Judge Technical S...|    Y4000|2012-10-17 00:00:00|  1000|       MALDEN|   MA|02148|        D
P| 15E|   C00042366|C00401224|     M|                            |       null|12021044111|          ENGINEER|JUDGE TECHNICAL S...|      |
|  5| 2012|10101201300112068191  m00021679750|    HAWTHORNE, JOAN|   C00042366|University of Nor...|    H5100|2012-10-11 00:00:00|   500|  GRAND FORKS|   ND|58201|        D
P| 15E|   C00042366|C00401224|     F|                            |       null|12021043352|          EDUCATOR|UNIVERSITY OF NOR...| Name |
|  6| 2012|10101201300112070761  h3001884327 |   HEDELMAN, SIDNEY|   C00042366|            Retired|    X1200|2012-10-04 00:00:00|   200|  MILL VALLEY|   CA|94941|        D
P| 15E|   C00042366|C00401224|     N|                            |       null|12021043390|       NOT EMPLOYED|                 N/A| Gen  |
|  7| 2012|10101201300112072281  h30017191620|     PRICE, LINDA N|   C00042366|            Retired|    X1200|2012-10-17 00:00:00|   500|         NAPA|   CA|94558|        D
P| 15E|   C00042366|C00401224|     F|                            |       null|12021051206|             RETIRED|                 N/A| Gen  |
|  8| 2012|10101201300112072561  h3001636082 |     CORBITT, CRAIG|   C00042366|Zelle, Hofmann et al|    K1000|2012-10-17 00:00:00|   250|  MILL VALLEY|   CA|94941|        D
P| 15E|   C00042366|C00401224|     M|                            |       null|12021041752|            ATTORNEY|   ZELLE HOFMANN LLP| Rept |
|  9| 2012|10101201300112073211  m00021077080 |      MCGAVICK, HUGH|   C00042366|           Attorney|    K1000|2012-10-13 00:00:00|   200|     PORTLAND|   OR|97219|        D
```

We renamed the column:




```
>>>
>>>
>>>
>>>
>>> df888.withColumnRenamed("pg","primary_or_general").show()
+---+-----+---------+--------+------------------+-----------+----+---------+------+
| id|cycle|fec_rec_no| filer_id|     donor_committee|  contrib_lend_trans|      city|state|  zip|prim_code|             date| amount|recipient_id|party| other_id|recip_c
ode|recip_prim_code|amend|report|primary_or_general|  microfilm|type|real_code|source|
+---+-----+---------+--------+------------------+-----------+----+---------+------+
---+---------------+-----+------+------------------+-----------+----+---------+------+
|  1| 2000|  1430708|C00000059|      Hallmark Cards|           null|       null|     |     |    C1400|2000-06-08 00:00:00| 1000.0|   C00341867| null|C00341867|
 PI|         J1200|    A|   M7 |                P|20035983133| 24K|    C1400| PAC  |
|  2| 2000|    62108|C00000372|Maintenance of Wa...|           null|       null|     |     |   LT400|1999-03-16 00:00:00| 5000.0|    C00190876|    D|C00190876|
 PI|         J2100|    N|   M4 |                P|99034461953| 24K|   LT400| PAC  |
|  3| 2000|   397553|C00000372|Maintenance of Wa...|           null|       null|     |     |   LT400|1999-07-31 00:00:00| 1000.0|   C00147512| null|C00147512|
 PI|         J7500|    N|   M8 |                P|99034784637| 24K|   LT400| PAC  |
|  4| 2000|  1325828|C00000372|Maintenance of Wa...|           null|       null|     |     |   LT400|2000-06-08 00:00:00|35000.0|   C00010603|    D|C00010603|
 DP|         Z5200|    N|   M7 |                P|20035881685| 24K|   LT400| PAC  |
|  5| 2000|  1146337|C00000372|Maintenance of Wa...|           null|       null|     |     |   LT400|2000-05-09 00:00:00|15000.0|   C00010603|    D|C00010603|
 DP|         Z5200|    A|   M6 |                P|20035732124| 24K|   LT400| PAC  |
|  6| 2000|   825273|C00000372|Maintenance of Wa...|           null|       null|     |     |   LT400|1999-11-21 00:00:00| 1092.0|   C00010603|    D|C00010603|
 DP|         Z5200|    A|  M12|                P|20035043806| 24Z|   LT400| PAC  |
|  7| 2000|  1491090|C00000372|Maintenance of Wa...|SOUTH DAKOTA DEMO...|SIOUX FALLS|   SD|57101|    LT400|2000-10-05 00:00:00| 1000.0|   C00160937|    D|C00160937|
 DP|         Z5200|    N|  12G|                G|20990227762| 24K|   LT400| PAC  |
|  8| 2000|    79737|C00000422|American Medical ...|           null|       null|     |     |    H1100|1999-04-28 00:00:00|15000.0|   C00042366|    D|C00042366|
 DP|         Z5200|    N|   M5 |                P|99034493582| 24K|    H1100| PAC  |
```

8)Save the cleaned data in S3 Bucket into parquet format:




```
>>>
>>>
>>> df1.write.parquet("s3://parquetfileformat/backers.parquet")
>>> df111.write.parquet("s3://parquetfileformat/candidates.parquet")
>>> df333.write.parquet("s3://parquetfileformat/commitees.parquet")
>>> df4.write.parquet("s3://parquetfileformat/fec_api_committees.parquet")
>>> df555.write.parquet("s3://parquetfileformat/individual_contributions")
>>> df666.write.parquet("s3://parquetfileformat/industry_codes.parquet")
>>> df7.write.parquet("s3://parquetfileformat/pac_records.parquet")
>>> df888.write.parquet("s3://parquetfileformat/pac_to_pacs.parquet")
>>> df9.write.parquet("s3://parquetfileformat/pacs.parquet")
>>> df10.write.parquet("s3://parquetfileformat/politicians.parquet")
>>>
>>>
>>>
>>>
```

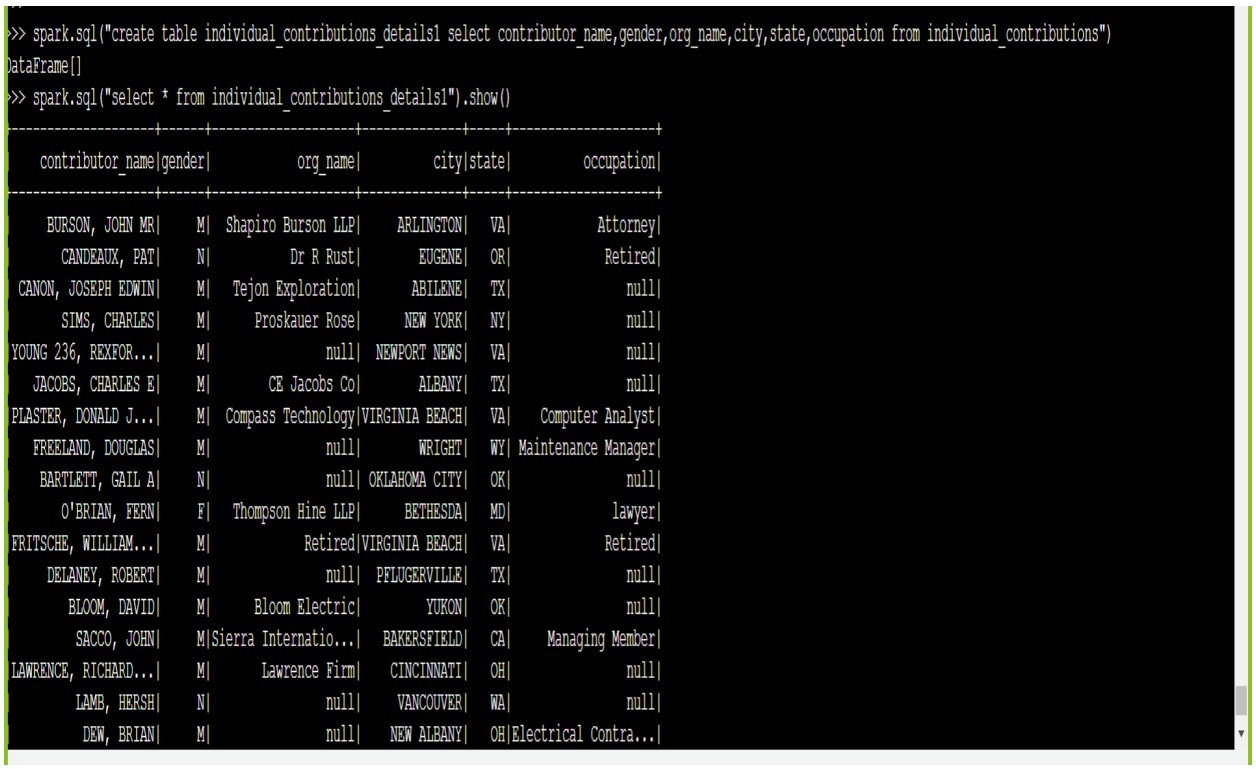
9)Load the data from S3 to pyspark:

```
>>>
>>> df1=spark.read.parquet("s3://parquetfileformat/backers.parquet")
>>> df2=spark.read.parquet("s3://parquetfileformat/candidates.parquet")
>>> df3=spark.read.parquet("s3://parquetfileformat/commitees.parquet")
>>> df4=spark.read.parquet("s3://parquetfileformat/fec_api_committees.parquet")
>>> df5=spark.read.parquet("s3://parquetfileformat/individual_contributions")
>>> df6=spark.read.parquet("s3://parquetfileformat/industry_codes.parquet")
>>> df7=spark.read.parquet("s3://parquetfileformat/pac_records.parquet")
>>> df8=spark.read.parquet("s3://parquetfileformat/pac_to_pacs.parquet")
>>> df9=spark.read.parquet("s3://parquetfileformat/pacs.parquet")
>>> df10=spark.read.parquet("s3://parquetfileformat/politicians.parquet")
>>>
>>>
>>>
```

```
>>> df1.count()
370
>>> df2.count()
67999
>>> df3.count()
157542
>>> df4.count()
45507
>>> df5.count()
24446421
>>> df6.count()
444
>>> df7.count()
44628
>>> df8.count()
1083525
>>> df9.count()
3539657
>>> df10.count()
26047
```

10)Create table and perform query in pyspark and store table in hive :

```
>>> sqlContext = SQLContext(sc)
>>>
>>> spark = SparkSession.builder.getOrCreate()
>>>
>>> spark.sql("create table senate select name,backer_level from backers where backer_level = 'senate'")
22/09/18 17:50:01 WARN HiveConf: HiveConf of name hive.server2.thrift.url does not exist
DataFrame[]
>>> spark.sql("select * from senate").show()
+--------------------+------------+
|                name|backer_level|
+--------------------+------------+
|          Kevin Pine|     senate|
|        Adam Edwards|     senate|
|"Michael ""Pinski...|     senate|
|     Trevor von Stein|     senate|
|         Susan Estep|     senate|
|    Wesley W. Driver|     senate|
|       Betsy Wentzel|     senate|
|           Chuck Dye|     senate|
|          David Cook|     senate|
|        David Powell|     senate|
|          Gene Mason|     senate|
|       Ken Ledenbach|     senate|
|         Micah Sifry|     senate|
|         Milind Shah|     senate|
|Please do not use...|     senate|
|        R.A.Plunkett|     senate|
|          Roy Clymer|     senate|
```



```
>>> spark.sql("create table individual_contributions_details1 select contributor_name,gender,org_name,city,state,occupation from individual_contributions")
DataFrame[]
>>> spark.sql("select * from individual_contributions_details1").show()
+--------------------+------+--------------------+-------------+-----+--------------------+
|    contributor_name|gender|            org_name|         city|state|          occupation|
+--------------------+------+--------------------+-------------+-----+--------------------+
|     BURSON, JOHN MR|     M|  Shapiro Burson LLP|    ARLINGTON|   VA|           Attorney|
|       CANDEAUX, PAT|     N|           Dr R Rust|      EUGENE|   OR|            Retired|
|  CANON, JOSEPH EDWIN|     M|   Tejon Exploration|      ABILENE|   TX|                null|
|       SIMS, CHARLES|     M|      Proskauer Rose|     NEW YORK|   NY|                null|
| YOUNG 236, REXFOR...|     M|                null| NEWPORT NEWS|   VA|                null|
|    JACOBS, CHARLES E|     M|       CE Jacobs Co|       ALBANY|   TX|                null|
| PLASTER, DONALD J...|     M| Compass Technology|VIRGINIA BEACH|   VA|    Computer Analyst|
|    FREELAND, DOUGLAS|     M|                null|      WRIGHT|   WY| Maintenance Manager|
|     BARTLETT, GAIL A|     N|                null| OKLAHOMA CITY|   OK|                null|
|       O'BRIAN, FERN|     F|   Thompson Hine LLP|     BETHESDA|   MD|              lawyer|
| FRITSCHE, WILLIAM...|     M|            Retired|VIRGINIA BEACH|   VA|            Retired|
|     DELANEY, ROBERT|     M|                null| PFLUGERVILLE|   TX|                null|
|        BLOOM, DAVID|     M|      Bloom Electric|        YUKON|   OK|                null|
|         SACCO, JOHN|     M|Sierra Internatio...|   BAKERSFIELD|   CA|     Managing Member|
| LAWRENCE, RICHARD...|     M|       Lawrence Firm|   CINCINNATI|   OH|                null|
|        LAMB, HERSH|     N|                null|    VANCOUVER|   WA|                null|
|         DEW, BRIAN|     M|                null|   NEW ALBANY|   OH|Electrical Contra...|
```

```
>>> spark.sql("create table countof_backerslevel select backer_level,count(backer_level) from backers group by backer_level")
DataFrame[]
>>> spark.sql("select * from countof_backerslevel").show()
+----------------+-----------------+
|     backer_level|count(backer_level)|
+----------------+-----------------+
|           house|               48|
|         upstart|               50|
|kickstarter_backer|              248|
|          senate|               22|
|      leadership|                2|
+----------------+-----------------+

>>>
```

spark.sql("create table cycleof_candidates select b.cid,c.cycle,b.name,party from backers b join candidates c where c.cycle=1996 and b.cid=c.cid")

spark.sql("select * from cycleof_candidates").show()

```
+---------+-----+-------------------+-----+
|      cid|cycle|               name|party|
+---------+-----+-------------------+-----+
|N00005165| 1996|             Warren|    R|
|N00002384| 1996|          Alex Cox|    D|
|N00008999| 1996|        Jorge Pinon|    R|
|N00000561| 1996|           Ainslie|    R|
|N00002247| 1996|           Nala R.|    R|
|N00007335| 1996|         Mark Menke|    D|
|N00004541| 1996|         lynn adler|    D|
|N00005195| 1996|         Gene Mason|    R|
|N00003333| 1996|   Andrew DiSabatino|    D|
|N00002097| 1996|    Wesley W. Driver|    D|
|N00000528| 1996|         Kevin Pine|    I|
|N00001743| 1996|         Steve King|    D|
|N00007063| 1996|       Nicole Lauber|    R|
|N00001750| 1996|       Steve Tuckner|    D|
|N00009082| 1996|Please do not use...|    R|
|N00009082| 1996|Please do not use...|    R|
|N00006863| 1996|    Happy Elder Geek|    R|
```

```
|N00001305| 1996|     Michael Maitlen|     D|
|N00007458| 1996|        Donald Simon|     D|
|N00001802| 1996|         Harper Reed|     R|
+---------+-----+--------------------+------+
only showing top 20 rows
```

---

spark.sql("select cycle,first_last_party,party from candidates where party = 'R'")

```
+-----+--------------------+-----+
|cycle|    first_last_party|party|
+-----+--------------------+-----+
| 2000|Richmond A Soluad...|    R|
| 1996|Mark Alan Behnke (R)|    R|
| 1996|Stephen Bonsal Yo...|    R|
| 1996|Raymond J Clatwor...|    R|
| 1996|        Jim Ford (R)|    R|
| 1996| Ernest J Istook (R)|    R|
| 1996|Stephen Wayne Hof...|    R|
| 1996|Olivia Coggin Eud...|    R|
| 1996|Michael Minoru Fa...|    R|
| 1996| Joseph Dr Smith (R)|    R|
| 1996| James A McClure (R)|    R|
| 2000|John A Holmes Jr (R)|    R|
| 1996|James T Robinson (R)|    R|
| 1996|William Jack Sext...|    R|
| 1996|       Gen Olson (R)|    R|
| 1996|Berwick (Barry) P...|    R|
| 1996|   Mark E Souder (R)|    R|
| 1996|Robert L Hammock (R)|    R|
| 1996|    Luis Acle Jr (R)|    R|
| 1996|    Jeffrey Bell (R)|    R|
+-----+--------------------+-----+
```

spark.sql("select party,count(party) from candidates where party<100 group by party").show()

```
+-----+------------+
|party|count(party)|
+-----+------------+
|    3|        5297|
+-----+------------+
```

spark.sql("select party,count(party) from candidates group by party")

```
+--------+------------+
|   party|count(party)|
+--------+------------+
| Jr (3)"|           4|
|       U|        1273|
|       D|       25778|
|       R|       27941|
|       L|        3053|
|       3|        5297|
|       I|        4653|
+--------+------------+
```

spark.sql("select party,count(party) from candidates group by party having party = 'R'").show()

```
+-----+------------+
|party|count(party)|
+-----+------------+
|    R|       27941|
+-----+------------+
```

Q.display the conributor_name whose state is CA, occupation is retired.

spark.sql(“select contributor_name, state , occupation from individual_contributions where state='CA' and occupation='retired'”)

Q.display the cycle,commitee type, contributor_name, city from fec_api_committees and individual_contributions.

spark.sql(“select i.contributor_name, f.cycle, f.committee_type, i.city from fec_api_committees f join individual_contributions i where f.cycle=i.cycle”)

Q:Display category_code,category_name from industry_codes by grouping the industry_code,industry_name,sector

spark.sql("select category_code,category_name from industry_codes group by industry_code,industry_name,sector")

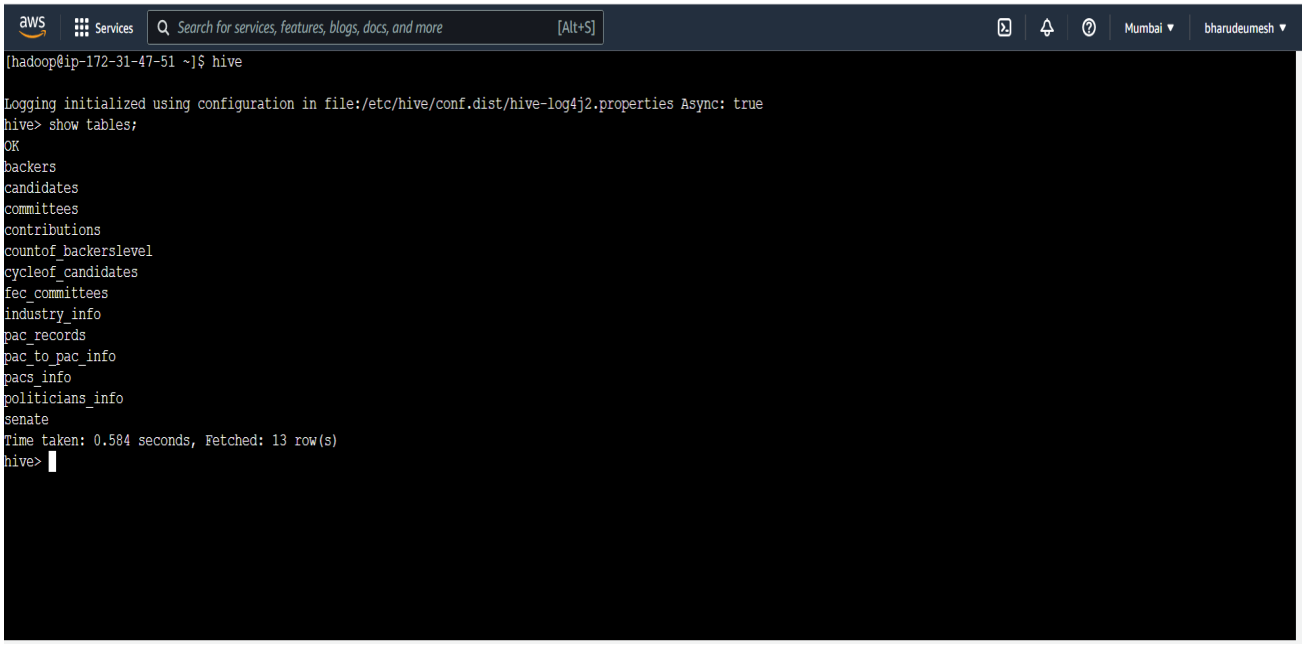
Q:which are the politicians contributed in different cycle

spark.sql("select p.cid,p.cycle,o.name as politician_name from pacs p join politicians o where p.cid=o.cid group by p.cycle")

```
>>> spark.sql("create table state_occupations select  contributor_name, state ,
occupation from individual_contributions")
DataFrame[]
>>> spark.sql("select * from state_occupations").show()
+--------------------+-----+--------------------+
|    contributor_name|state|          occupation|
+--------------------+-----+--------------------+
|     BURSON, JOHN MR|   VA|            Attorney|
|       CANDEAUX, PAT|   OR|             Retired|
| CANON, JOSEPH EDWIN|   TX|                null|
|       SIMS, CHARLES|   NY|                null|
|YOUNG 236, REXFOR...|   VA|                null|
|   JACOBS, CHARLES E|   TX|                null|
|PLASTER, DONALD J...|   VA|    Computer Analyst|
|   FREELAND, DOUGLAS|   WY| Maintenance Manager|
|    BARTLETT, GAIL A|   OK|                null|
|       O'BRIAN, FERN|   MD|              lawyer|
|FRITSCHE, WILLIAM...|   VA|             Retired|
|     DELANEY, ROBERT|   TX|                null|
|        BLOOM, DAVID|   OK|                null|
|         SACCO, JOHN|   CA|     Managing Member|
|LAWRENCE, RICHARD...|   OH|                null|
|         LAMB, HERSH|   WA|                null|
|          DEW, BRIAN|   OH|Electrical Contra...|
|  CHRISTENSEN, JAYNE|   OK|                null|
|  SANSARICQ, BERNARD|   FL|                null|
|    AYERS, HELEN MRS|   IL|                null|
+--------------------+-----+--------------------+
only showing top 20 rows
```

```
>>> spark.sql("select contributor_name, state , occupation from
individual_contributions").show()
+--------------------+-----+--------------------+
|    contributor_name|state|          occupation|
+--------------------+-----+--------------------+
|     BURSON, JOHN MR|   VA|            Attorney|
|       CANDEAUX, PAT|   OR|             Retired|
| CANON, JOSEPH EDWIN|   TX|                null|
|       SIMS, CHARLES|   NY|                null|
|YOUNG 236, REXFOR...|   VA|                null|
|   JACOBS, CHARLES E|   TX|                null|
|PLASTER, DONALD J...|   VA|    Computer Analyst|
|   FREELAND, DOUGLAS|   WY| Maintenance Manager|
|    BARTLETT, GAIL A|   OK|                null|
|       O'BRIAN, FERN|   MD|              lawyer|
|FRITSCHE, WILLIAM...|   VA|             Retired|
|     DELANEY, ROBERT|   TX|                null|
|        BLOOM, DAVID|   OK|                null|
|         SACCO, JOHN|   CA|     Managing Member|
|LAWRENCE, RICHARD...|   OH|                null|
|         LAMB, HERSH|   WA|                null|
|          DEW, BRIAN|   OH|Electrical Contra...|
|  CHRISTENSEN, JAYNE|   OK|                null|
|  SANSARICQ, BERNARD|   FL|                null|
|    AYERS, HELEN MRS|   IL|                null|
+--------------------+-----+--------------------+
only showing top 20 rows
```

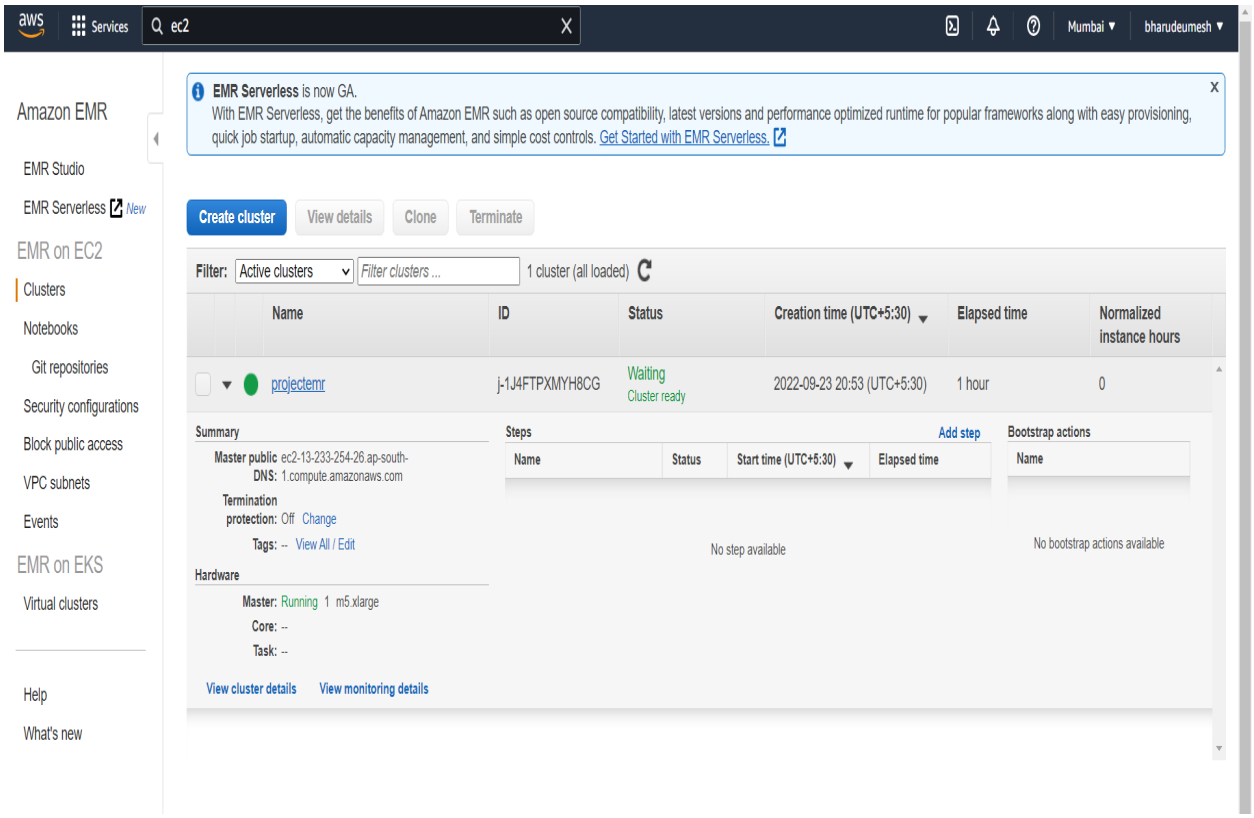
9.Tables which are stored in hive:

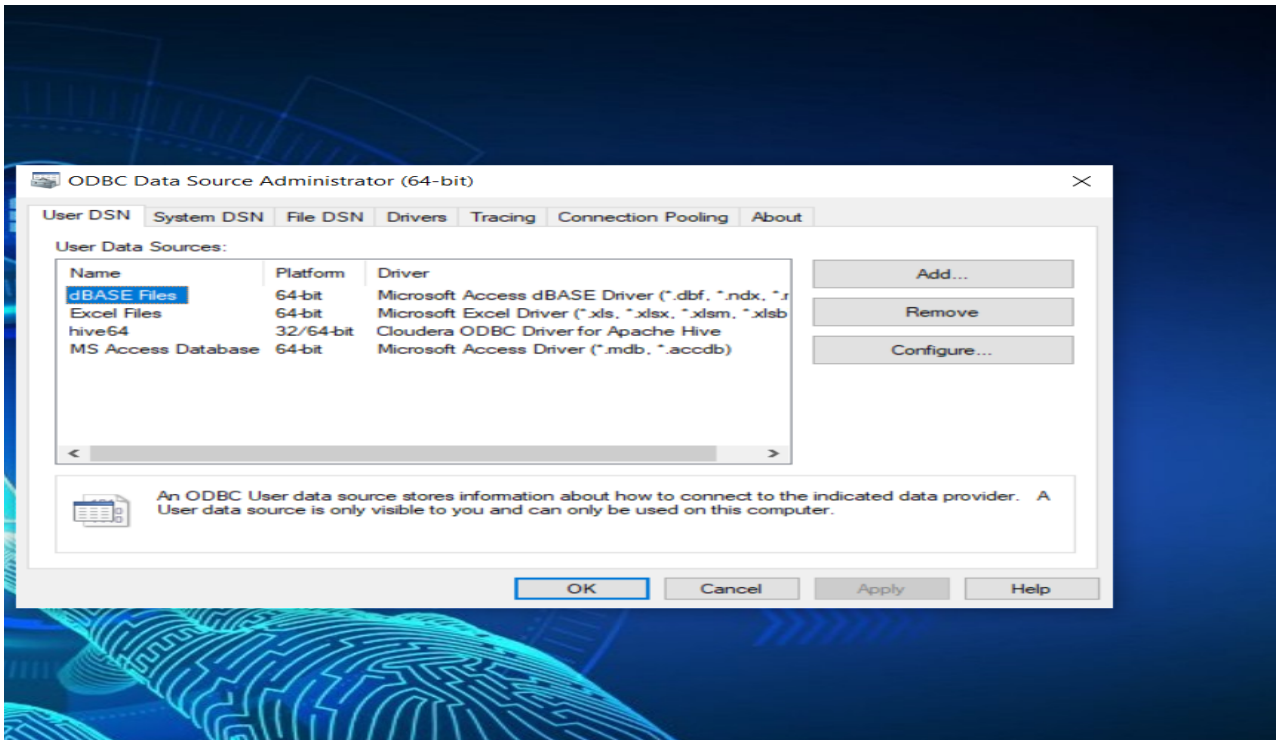


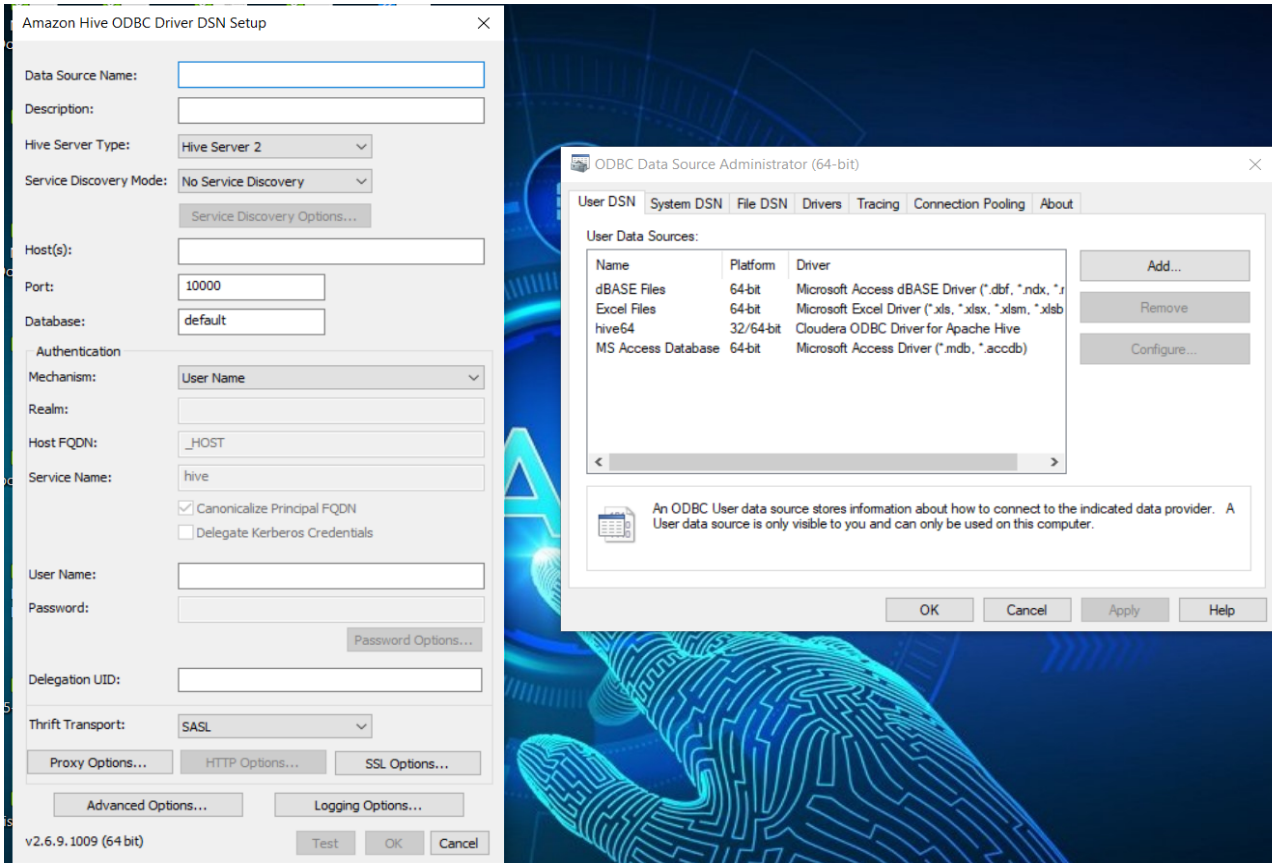
10.connectivity to power bi:

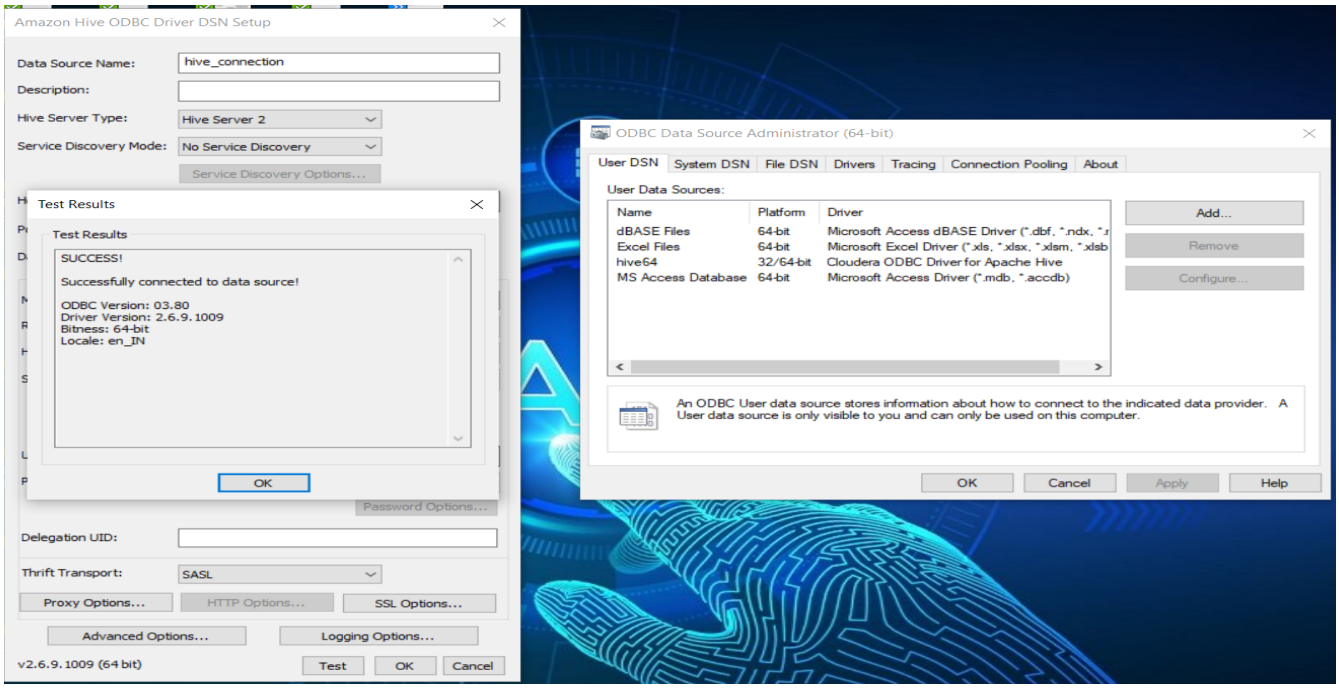
From the created tables we are now connecting to power bi for better Visualization and understanding below are the queries with their respective visualization.

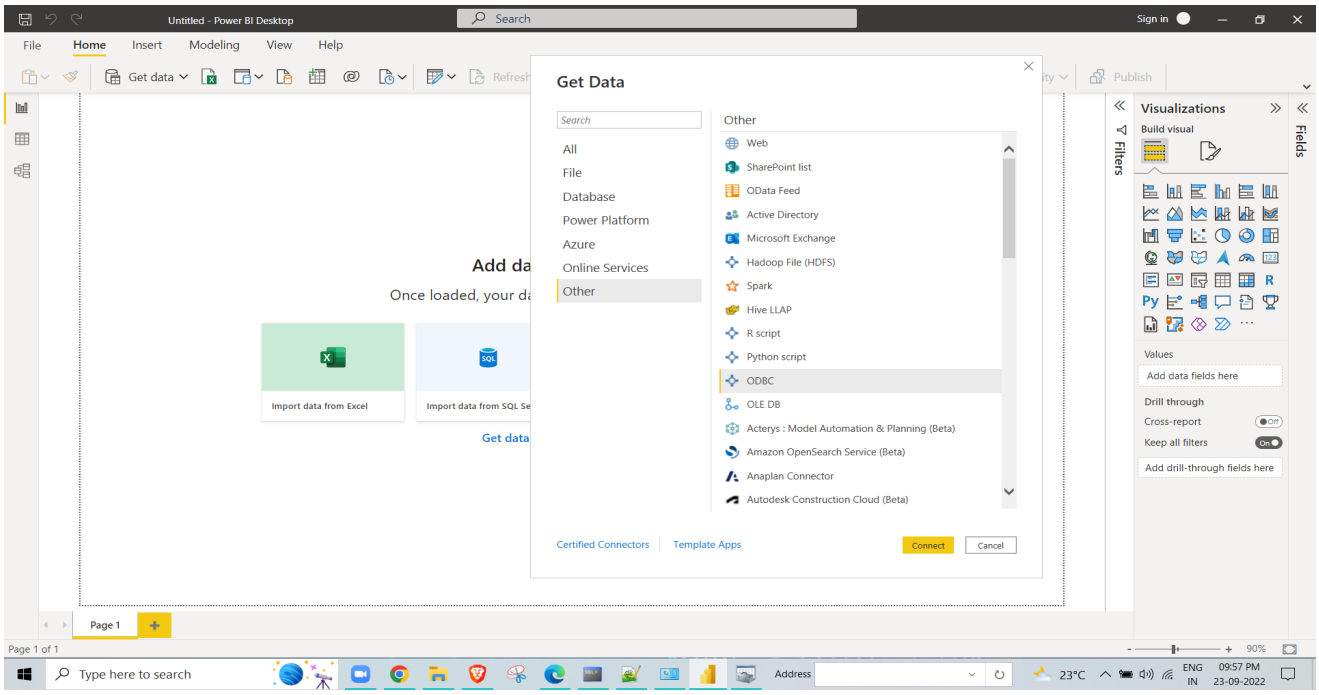
We connected power bi using ODBC DATA SOURCES by following steps

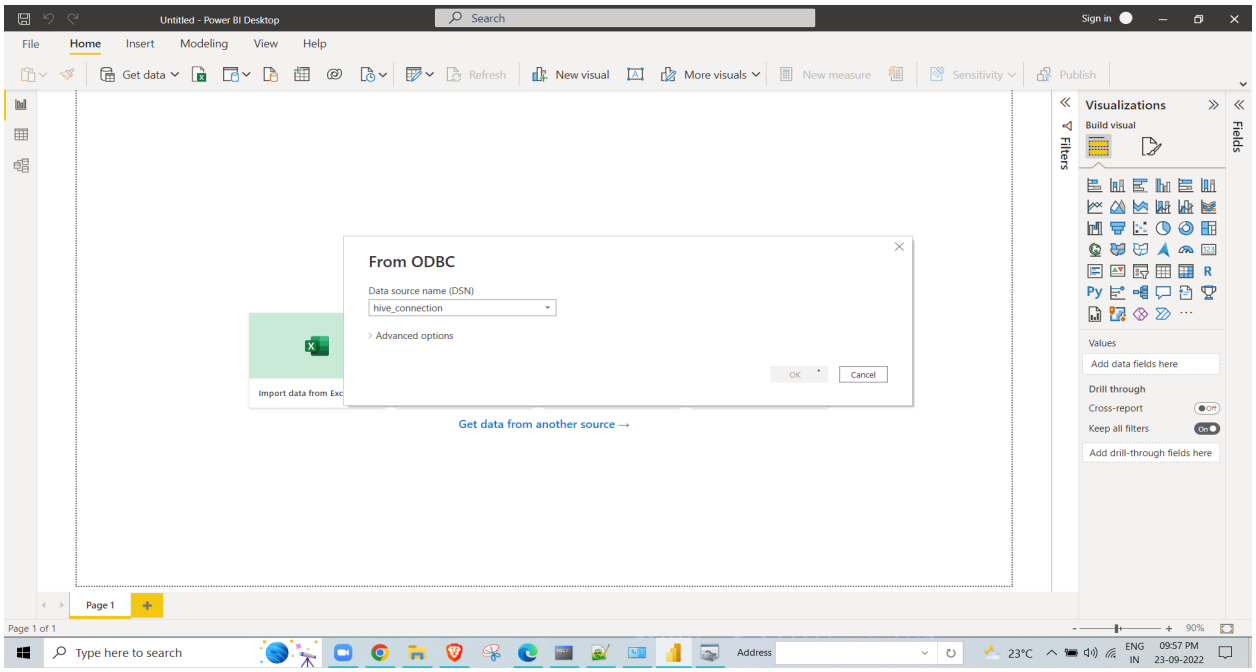


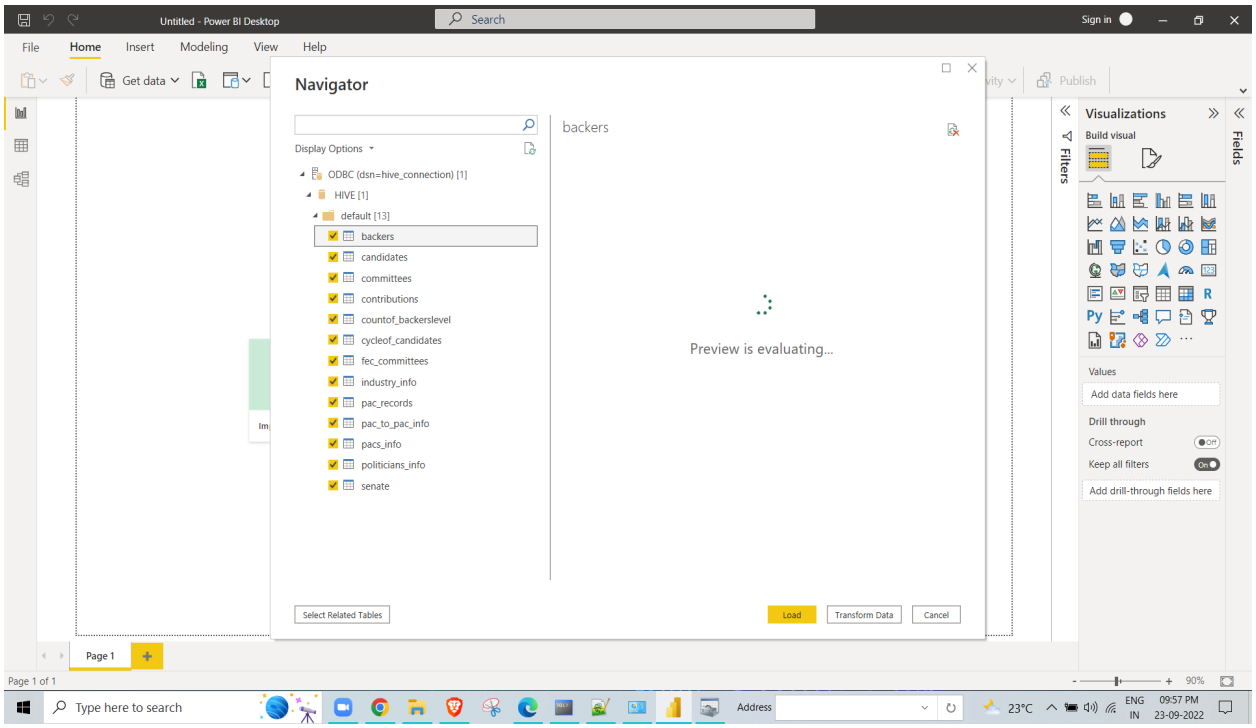

ODBC Data Source Administrator (64-bit)
User DSN
System DSN
File DSN
Drivers
Tracing
Connection Pooling
About
User Data Sources:
Name
Platform
Driver
dBASE Files
64-bit
Microsoft Access dBASE Driver (*.dbf, *.ndx, *.r
Excel Files
64-bit
Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb
hive64
32/64-bit
Cloudera ODBC Driver for Apache Hive
MS Access Database
64-bit
Microsoft Access Driver (*.mdb, *.accdb)
Add...
Remove
Configure...
An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you and can only be used on this computer.
OK
Cancel
Apply
Help


Amazon Hive ODBC Driver DSN Setup
Data Source Name:
Description:
Hive Server Type:
Hive Server 2
Service Discovery Mode:
No Service Discovery
Service Discovery Options...
Host(s):
Port:
10000
Database:
default
Authentication
Mechanism:
User Name
Realm:
Host FQDN:
_HOST
Service Name:
hive
Canonicalize Principal FQDN
Delegate Kerberos Credentials
User Name:
Password:
Password Options...
Delegation UID:
Thrift Transport:
SASL
Proxy Options...
HTTP Options...
SSL Options...
Advanced Options...
Logging Options...
v2.6.9.1009 (64 bit)
Test
OK
Cancel
ODBC Data Source Administrator (64-bit)
User DSN
System DSN
File DSN
Drivers
Tracing
Connection Pooling
About
User Data Sources:
Name
Platform
Driver
dBASE Files
64-bit
Microsoft Access dBASE Driver (*.dbf, *.ndx, *.r
Excel Files
64-bit
Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb
hive64
32/64-bit
Cloudera ODBC Driver for Apache Hive
MS Access Database
64-bit
Microsoft Access Driver (*.mdb, *.accdb)
Add...
Remove
Configure...
An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you and can only be used on this computer.
OK
Cancel
Apply
Help

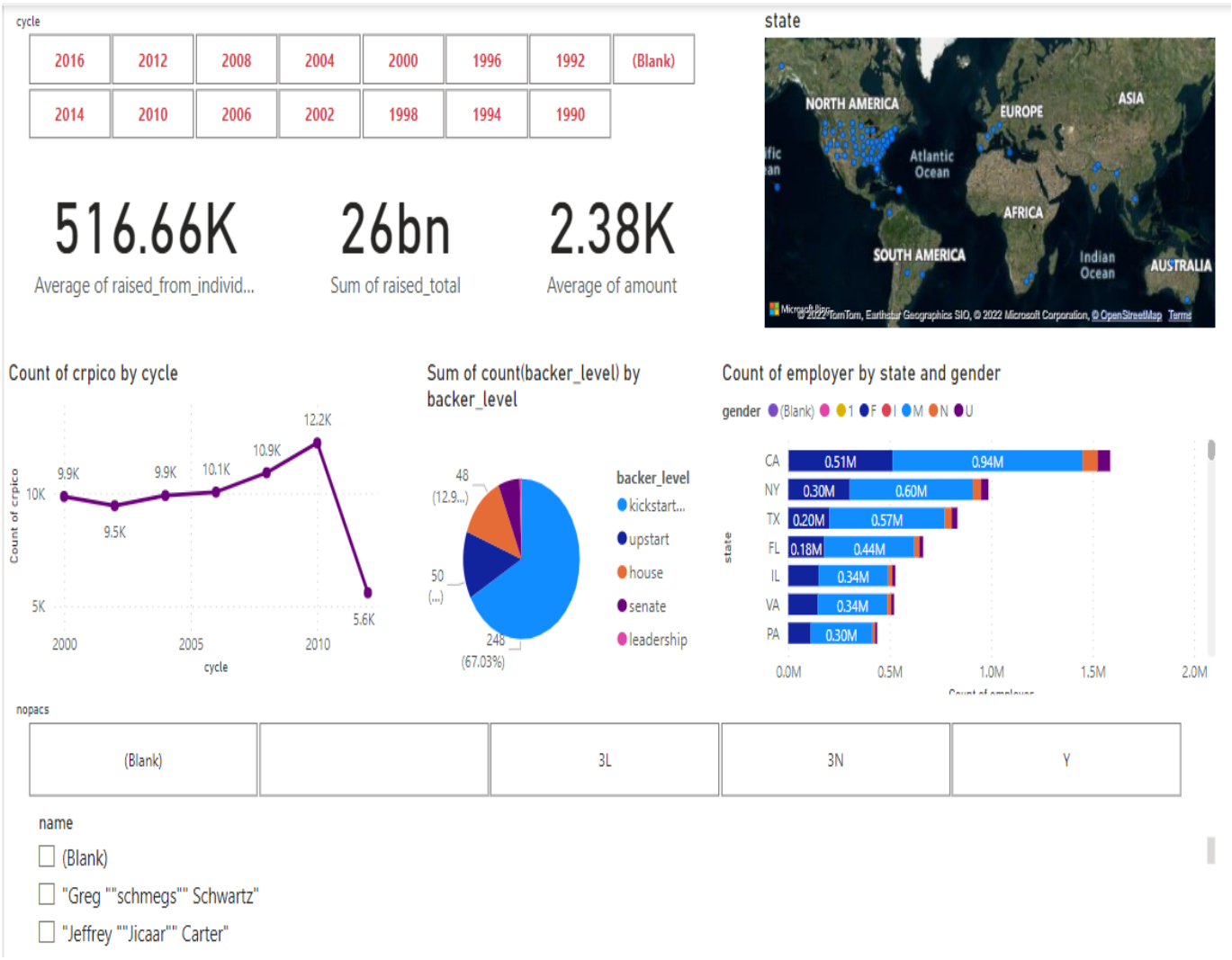

Amazon Hive ODBC Driver DSN Setup
Data Source Name:
hive_connection
Description:
Hive Server Type:
Hive Server 2
Service Discovery Mode:
No Service Discovery
Service Discovery Options...
Test Results
SUCCESS!
Successfully connected to data source!
ODBC Version: 03.80
Driver Version: 2.6.9.1009
Bitness: 64-bit
Locale: en_IN
OK
Password Options...
Delegation UID:
Thrift Transport:
SASL
Proxy Options...
HTTP Options...
SSL Options...
Advanced Options...
Logging Options...
v2.6.9.1009 (64 bit)
Test
OK
Cancel
ODBC Data Source Administrator (64-bit)
User DSN
System DSN
File DSN
Drivers
Tracing
Connection Pooling
About
User Data Sources:
Name
Platform
Driver
dBASE Files
64-bit
Microsoft Access dBASE Driver (*.dbf, *.ndx, *
Excel Files
64-bit
Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb
hive64
32/64-bit
Cloudera ODBC Driver for Apache Hive
MS Access Database
64-bit
Microsoft Access Driver (*.mdb, *.accdb)
Add...
Remove
Configure...
An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you and can only be used on this computer.
OK
Cancel
Apply
Help


Untitled - Power BI Desktop
Search
Sign in
File
Home
Insert
Modeling
View
Help
Get data
Publish
Get Data
Search
All
File
Database
Power Platform
Azure
Online Services
Other
Other
Web
SharePoint list
OData Feed
Active Directory
Microsoft Exchange
Hadoop File (HDFS)
Spark
Hive LLAP
R script
Python script
ODBC
OLE DB
Acterys : Model Automation & Planning (Beta)
Amazon OpenSearch Service (Beta)
Anaplan Connector
Autodesk Construction Cloud (Beta)
Certified Connectors
Template Apps
Connect
Cancel
Add da
Once loaded, your da
Import data from Excel
Import data from SQL Se
Get data
Visualizations
Build visual
Filters
Fields
Values
Add data fields here
Drill through
Cross-report
Keep all filters
Add drill-through fields here
Page 1
Page 1 of 1
90%
Type here to search
Address
23°C
ENG
IN
09:57 PM
23-09-2022


Untitled - Power BI Desktop
Search
Sign in
File
Home
Insert
Modeling
View
Help
Get data
Refresh
New visual
More visuals
New measure
Sensitivity
Publish
Visualizations
Build visual
Filters
Fields
From ODBC
Data source name (DSN)
hive_connection
Advanced options
OK
Cancel
Import data from Exc
Get data from another source →
Values
Add data fields here
Drill through
Cross-report
Keep all filters
Add drill-through fields here
Page 1
Page 1 of 1
90%
Type here to search
Address
23°C
ENG
IN
09:57 PM
23-09-2022


Untitled - Power BI Desktop
Search
Sign in
File
Home
Insert
Modeling
View
Help
Get data
Publish
Navigator
Display Options
ODBC (dsn=hive_connection) [1]
HIVE [1]
default [13]
backers
candidates
committees
contributions
countof_backerslevel
cycleof_candidates
fec_committees
industry_info
pac_records
pac_to_pac_info
pacs_info
politicians_info
senate
backers
Preview is evaluating...
Select Related Tables
Load
Transform Data
Cancel
Visualizations
Build visual
Filters
Fields
Values
Add data fields here
Drill through
Cross-report
Keep all filters
Add drill-through fields here
Page 1
Page 1 of 1
90%
Type here to search
Address
23°C
ENG
IN
09:57 PM
23-09-2022

11.PowerBI Dashboard:



12. Requirements Specification

Hardware Requirement

500 GB hard drive (Minimum requirement)

8 GB RAM (Minimum requirement)

PC x64-bit CPU

Software Requirement

Windows/Mac/Linux

Any Modern Web Browser like Google Chrome
To access aws account for creating emr etc
Power BI Desktop for creation dashboard

ODBC data sources for connecting hive to Power Bi

References

- US Federal Campaign Finance Data Analysis Using BigData Technologies Dataset
  https://www.kaggle.com/datasets/jeegarmaru/campaign-contributions-19902016

  https://github.com/Priyanka743/Project-PUBGAnalysis