## Loading Historical Transactions Data into NoSQL Database

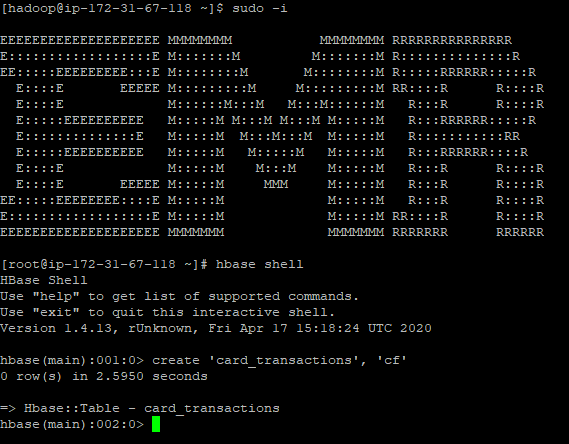
**Commands to load the past transactions data into NoSQL database**

1. Change to root user and launch hbase shell to create target table card\_transactions with column family as cf:

sudo -i

hbase shell

create ‘card\_transactions', 'cf'



1. Run the below command on the EMR cluster to install gcc and happybase

sudo yum update

yum install gcc

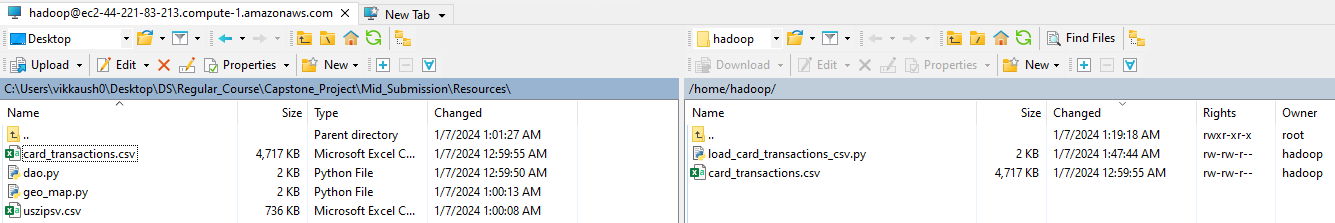
sudo yum install python3-devel

pip install happybase

jps

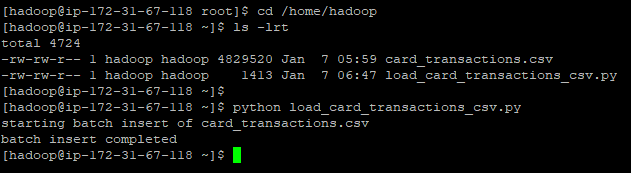
python -c "import happybase"

1. Copy the files - load\_card\_transactions\_csv.py and card\_transactions.csv to /home/hadoop folder using WINSCP tool.



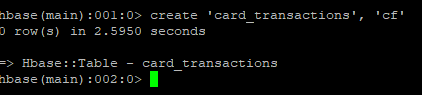
1. Run the below python script(load\_card\_transactions\_csv.py) to load data into HBase table from csv file:

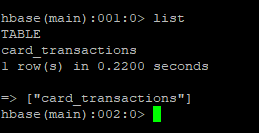
import happybase  
  
# create connection  
connection = happybase.Connection('localhost'**,** port=**9090,** autoconnect=False)  
  
# open connection to perform operations  
def open\_connection():  
 connection.open()  
  
# close opened connection   
def close\_connection():  
 connection.close()  
  
# get the pointer to a table  
def get\_table(tablename):  
 open\_connection()  
 table = connection.table(tablename)  
 close\_connection()  
 return table  
  
# batch insert data for transactions  
def batch\_insert\_data(filename**,** tablename):  
 print("starting batch insert of "+filename)  
 file = open(filename**,** 'r')  
 table = get\_table(tablename)  
 col\_names = ['card\_id'**,**'member\_id'**,**'amount'**,**'postcode'**,**'pos\_id'**,**'transaction\_dt'**,**'status']   
 open\_connection()  
 i = **0** with table.batch(batch\_size=**1000**) as b:  
 for line in file:  
 if i!=**0**:  
 temp = line.strip().split(",")  
 # taking card\_id, amount, transaction\_dt as rowkey  
 row\_key = f"{temp[**0**]}\_{temp[**2**]}\_{temp[**5**]}".encode() # Encode row key to bytes  
 for j in range(len(col\_names)) :  
 if j not in [**0,2,5**]:  
 b.put(row\_key**,** {b'cf:' + col\_names[j].encode(): temp[j].encode()})  
   
 i+=**1** file.close()  
 print("batch insert completed")  
 close\_connection()   
  
  
batch\_insert\_data('card\_transactions.csv'**,** 'card\_transactions')



**<Command to list the table in which the data is loaded and the command to get the count of the rows of the table>**

1. list command is used to list the hbase tables.

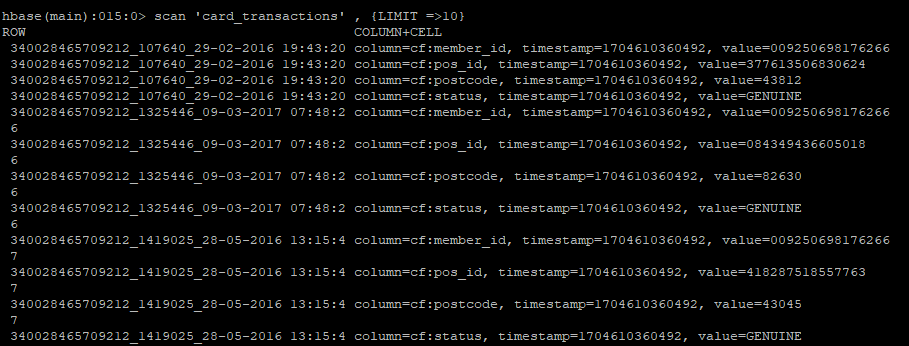




**Screenshot of the table created**

1. scan 'card\_transactions' , {LIMIT =>10}

The command will display the first 10 rows in the table



1. count of rows in card\_transactions table

count 'card\_transactions', INTERVAL => 10000

