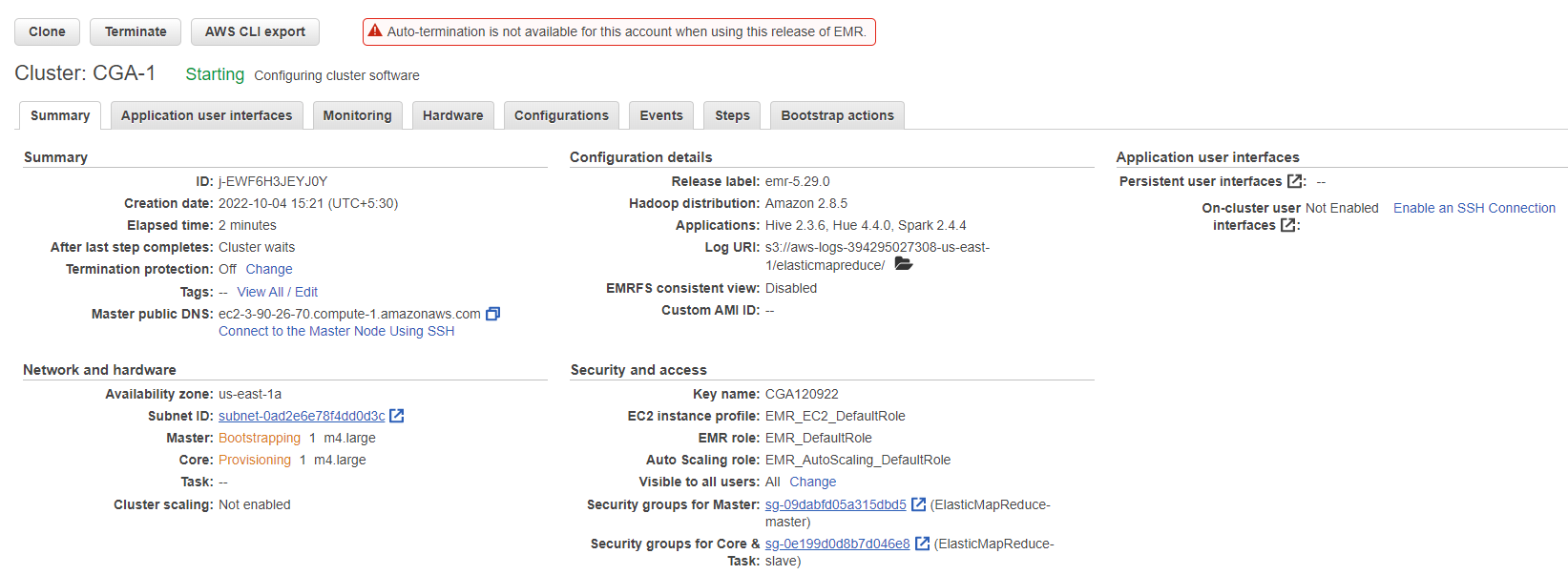
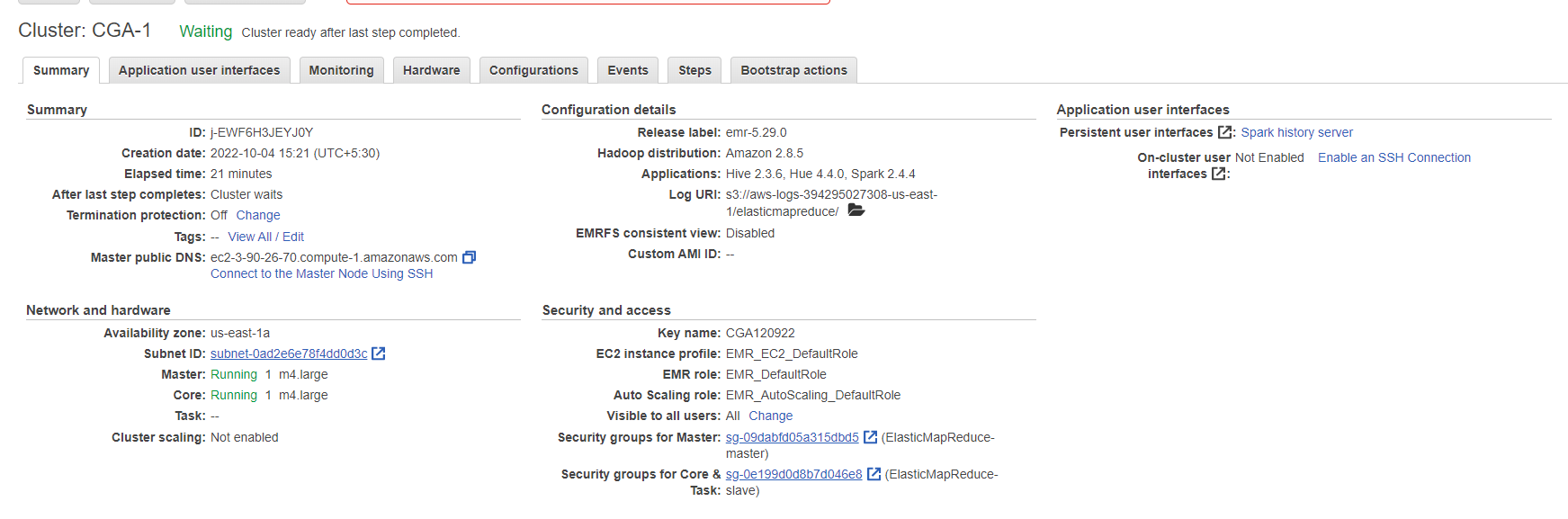
We login to Nuvepro dashboard, go to the console and then to EMR home page → Click on

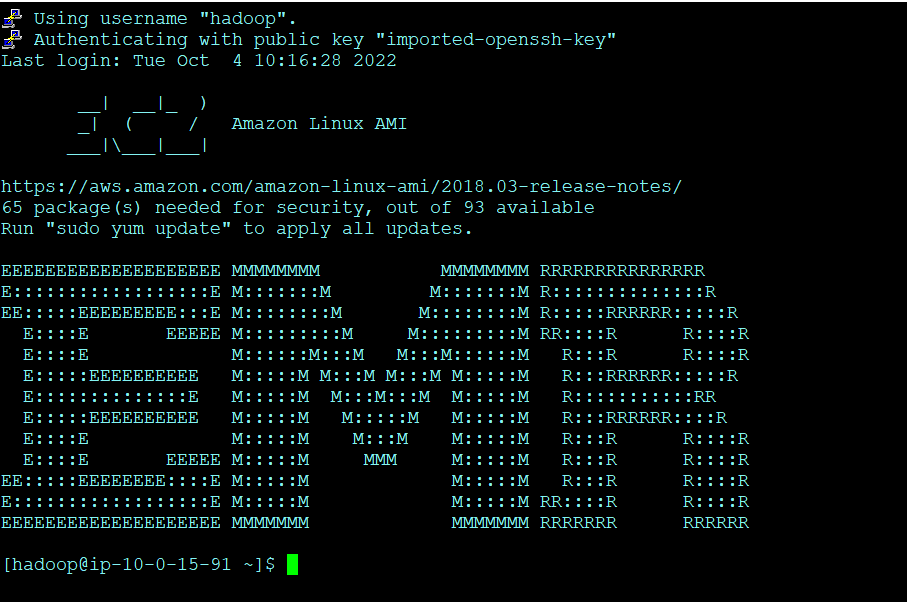
Create Cluster → select release EMR 5.29.0 and select required service for the case study.

Launching an EMR cluster that utilizes Hive services

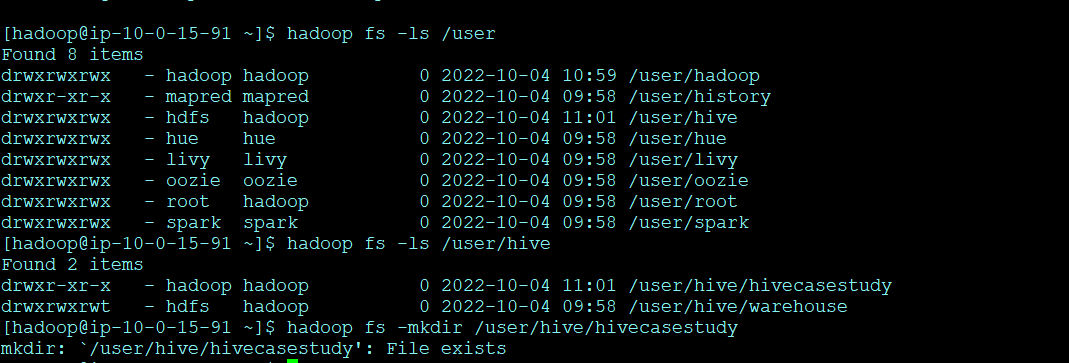




*Launch an EMR Cluster and connect to master node through SSH*

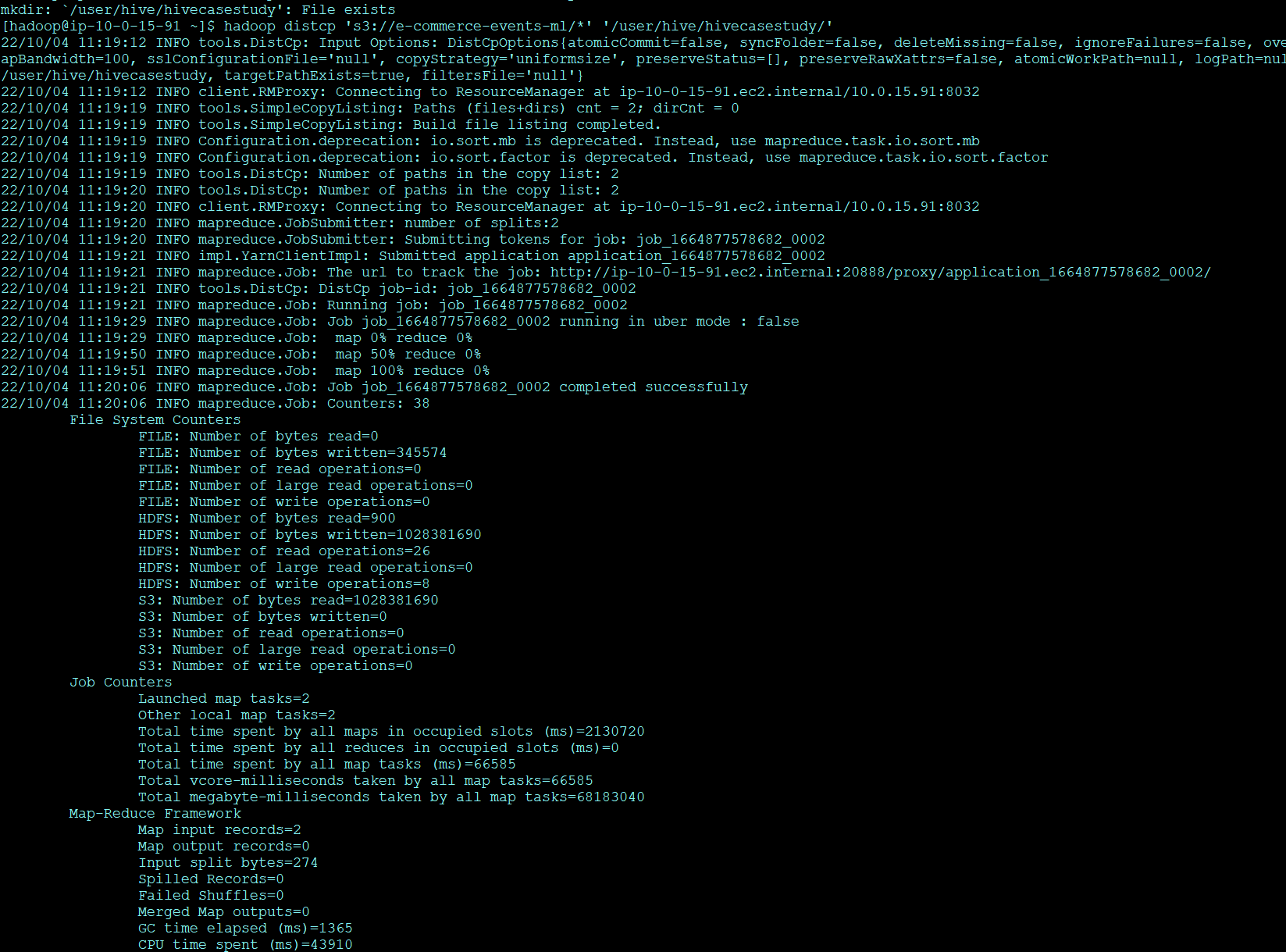


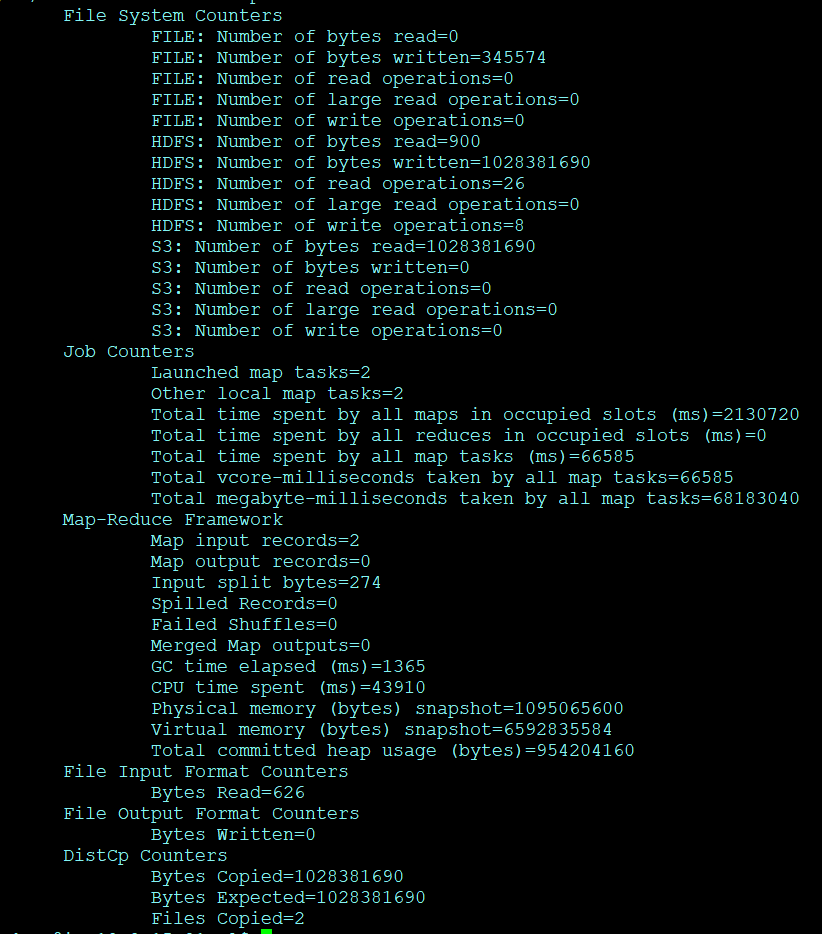
Creating a folder “hivecasestudy ”on the Hadoop file system



Moving the data from S3 bucket into the HDFS

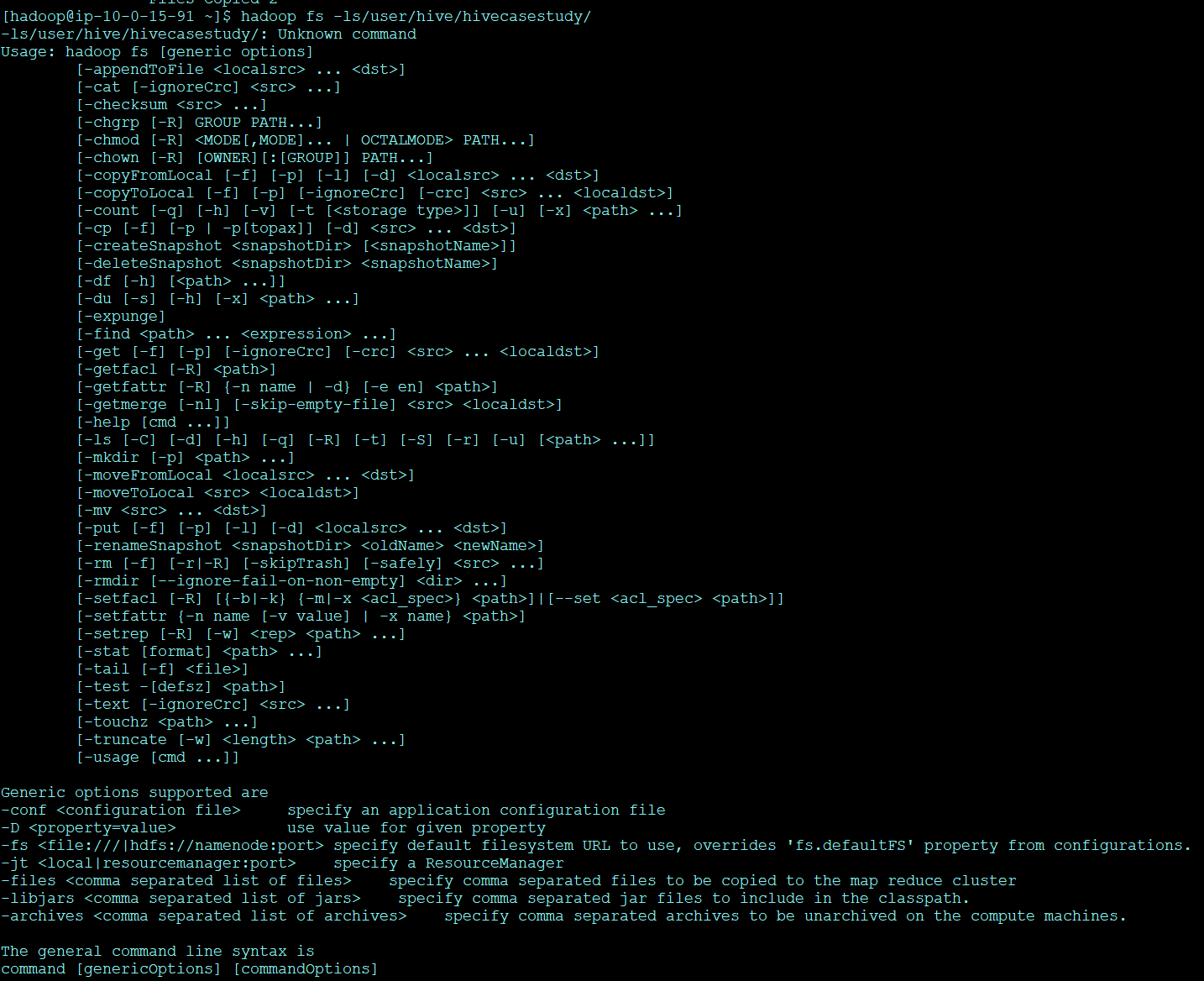
hadoop distcp 's3://e-commerce-events-ml/\*' '/user/hive/hivecasestudy/'





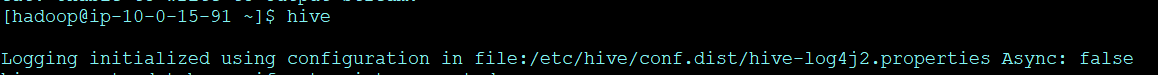
4 Checking if the files are correctly imported to HDFS

hadoop fs -ls/user/hive/hivecasestudy/

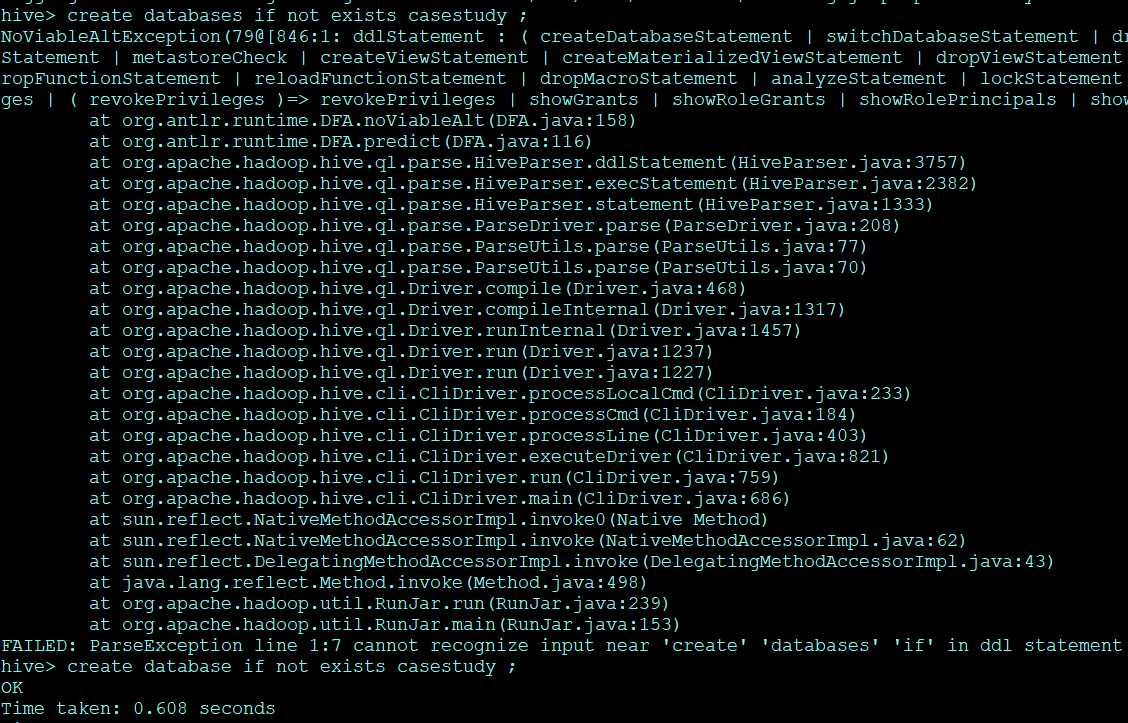


We can confirm the databases were loaded successfully.

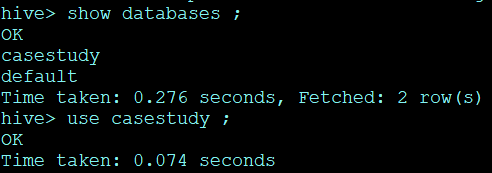
Changing cmd to the hive



Create database if not exists casestudy;

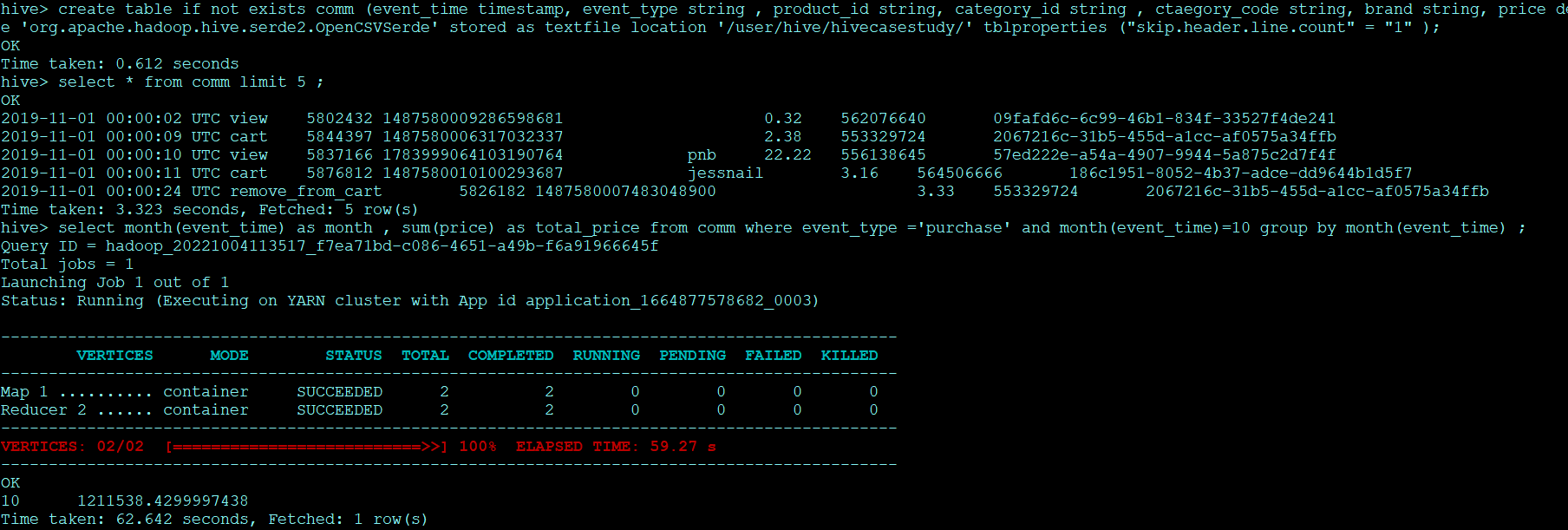


Checking data base

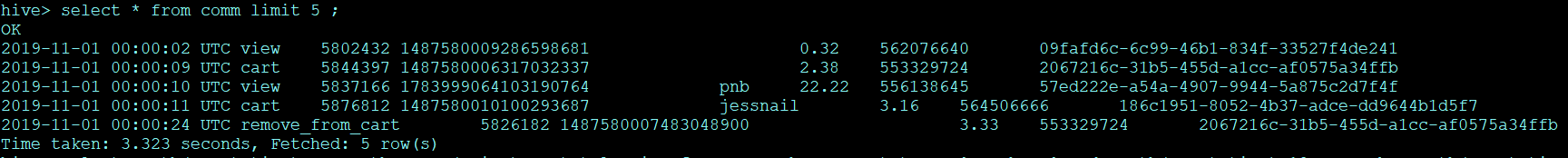


Creating an External Table, comm:

create table if not exists comm (event\_time timestamp, event\_type string , product\_id string, category\_id string , ctaegory\_code string, brand string, price decimal (10,2),user\_id bigint, user\_session string ) row format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde' stored as textfile location '/user/hive/hivecasestudy/' tblproperties ("skip.header.line.count" = "1" );



select \* from comm limit 5 ; query to view the dataset



Loading the Data into the table:

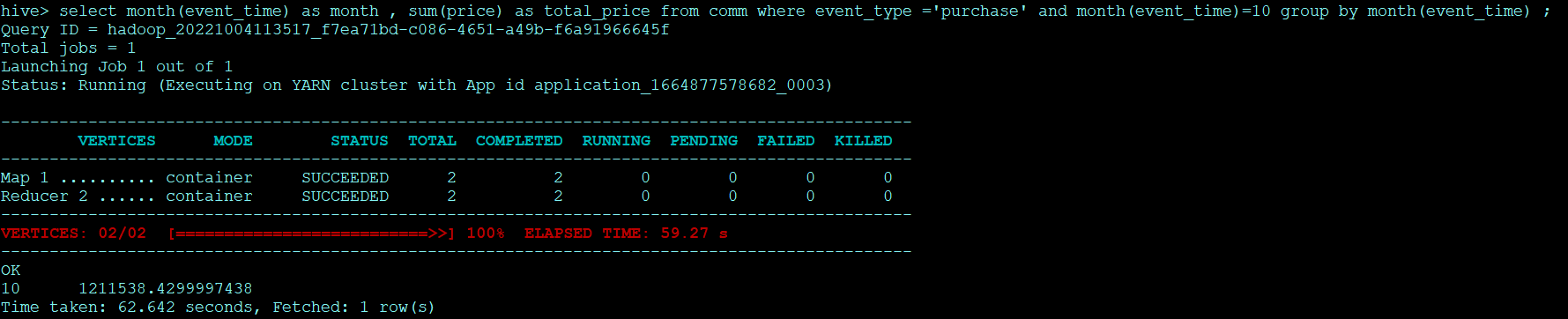
hive> load data inpath '/hiveassignment/2019-Oct.csv' into table sales;

hive> load data inpath '/hiveassignment/2019-Nov.csv' into table sales;

We are required to provide answers to the questions given below:

1 Find the total revenue generated due to purchases made in October.

select month(event\_time) as month , sum(price) as total\_price from comm where event\_type ='purchase' and month(event\_time)=10 group by month(event\_time) ;



1211538.4299997438

Here the query takes 69.373 seconds which can be optimized by creating dynamic partition and then compare the execution time.

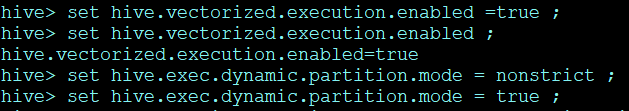
**Dynamic Partitioning and Bucketing:**

hive> set hive.vectorized.execution.enabled =true ;

hive> set hive.vectorized.execution.enabled ;

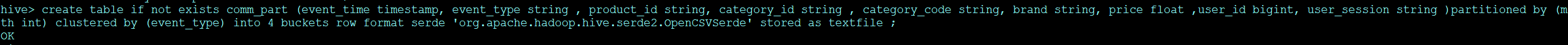
hive> set hive.exec.dynamic.partition.mode = nonstrict ;

hive> set hive.exec.dynamic.partition.mode = true ;



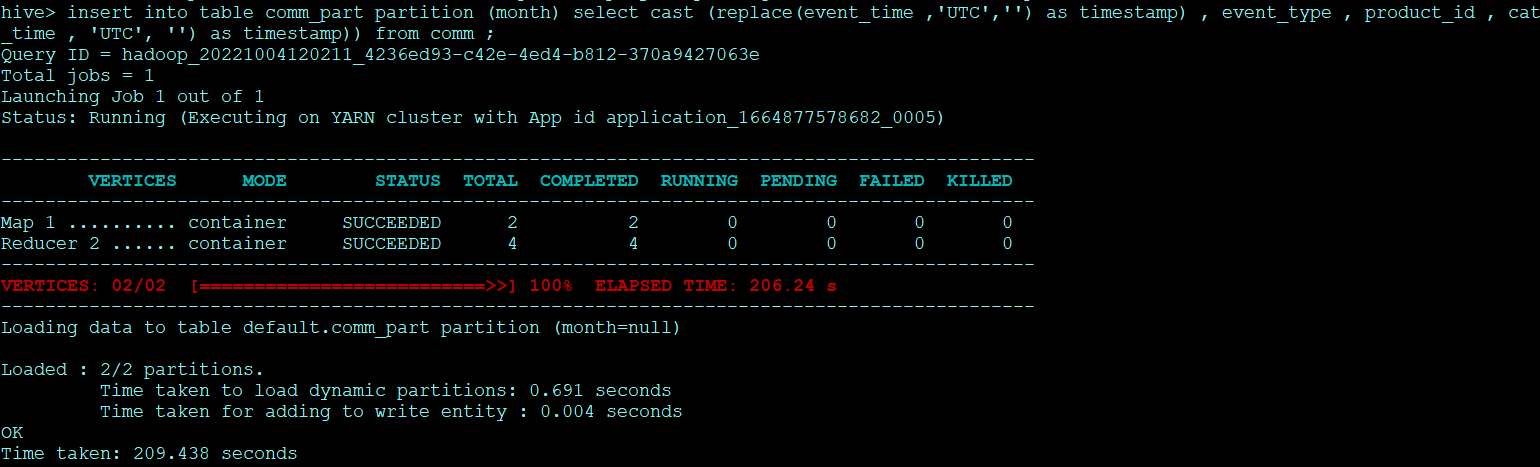
**Creating a table by name comm\_part to store the dataset which we partitioned by using ‘month int’and clustered by ‘event\_type’.**

create table if not exists comm\_part (event\_time timestamp, event\_type string , product\_id string, category\_id string , category\_code string, brand string, price float ,user\_id bigint, user\_session string )partitioned by (month int) clustered by (event\_type) into 4 buckets row format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde' stored as textfile ;



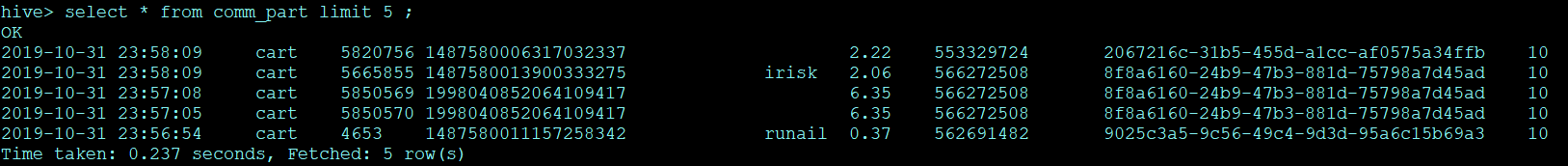
**Loading the data into the new table:**

hive> insert into table comm\_part partition (month) select cast (replace(event\_time ,'UTC','') as timestamp) , event\_type , product\_id , category\_id , ctaegory\_code , brand , price , user\_id ,user\_session , month(cast(replace(event\_time , 'UTC', '') as timestamp)) from comm ;



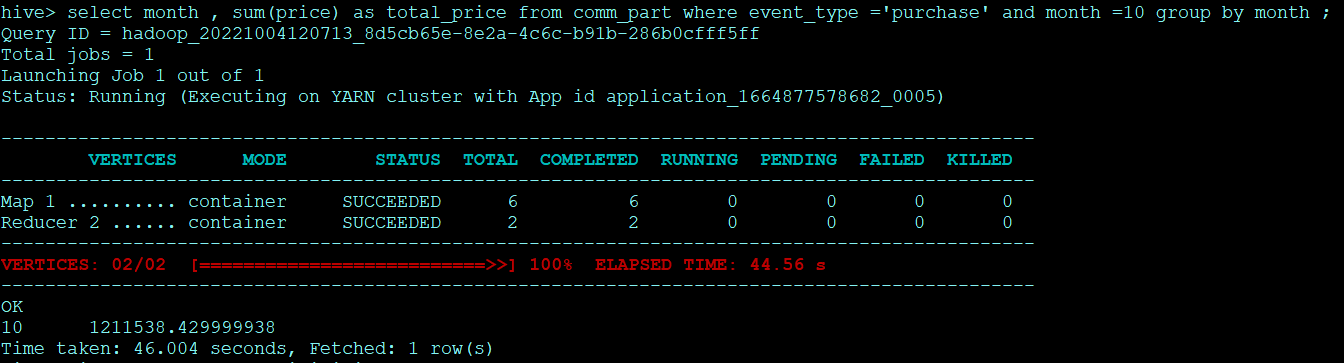
Checking partition database

select \* from comm\_part limit 5 ;



Now executing the same Q1 in the partition database:

hive> select month , sum(price) as total\_price from comm\_part where event\_type ='purchase' and month =10 group by month ;

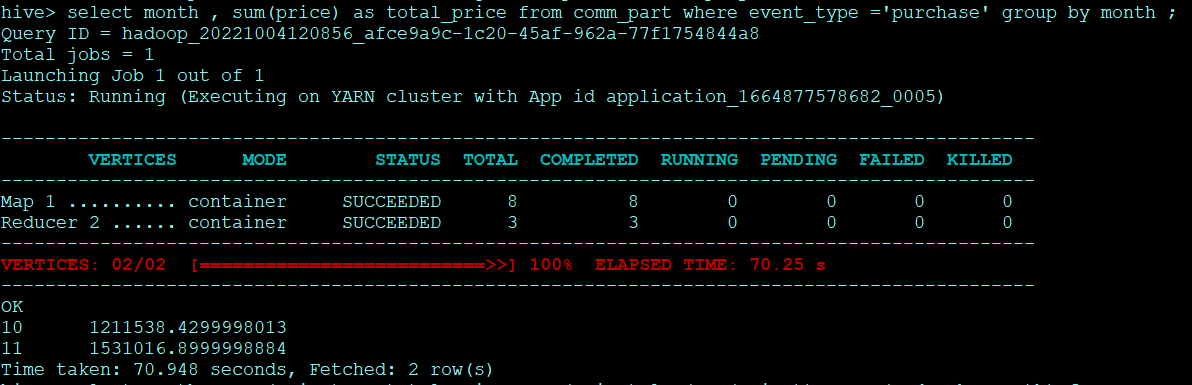


We can notice how the time taken reduced drastically due to partitioning and bucketing. Now it took only 46 sec.

The total sales in the month of October is 1211538.42

**2 Write a query to yield the total sum of purchases per month in a single output**.

hive> select month , sum(price) as total\_price from comm\_part where event\_type ='purchase' group by month ;

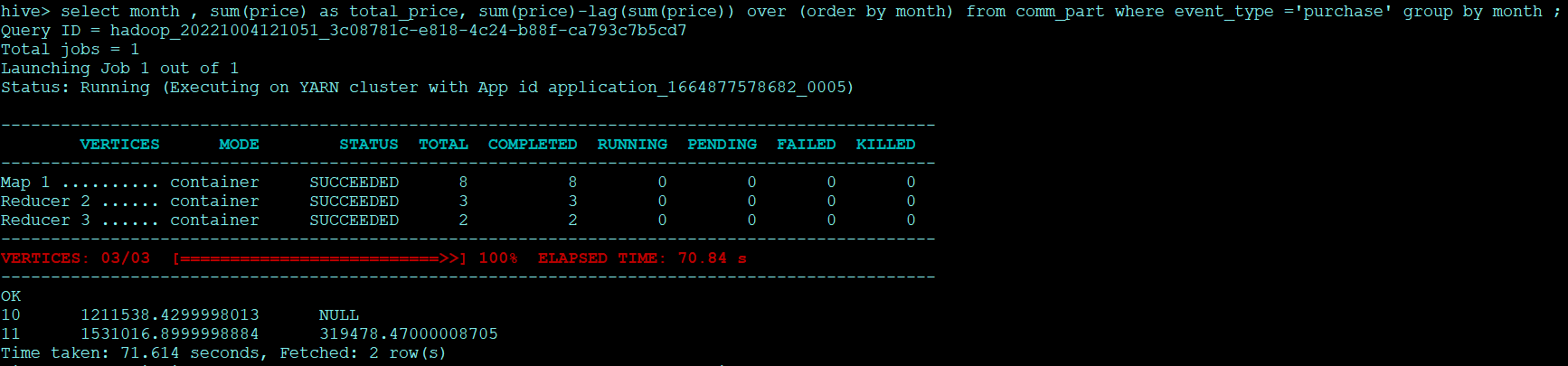


Total revenue generated due to purchases made in October is 1211538.429

Total revenue generated due to purchases made in November is 1531016.899

**3 Write a query to find the change in revenue generated due to purchases from October to November.**

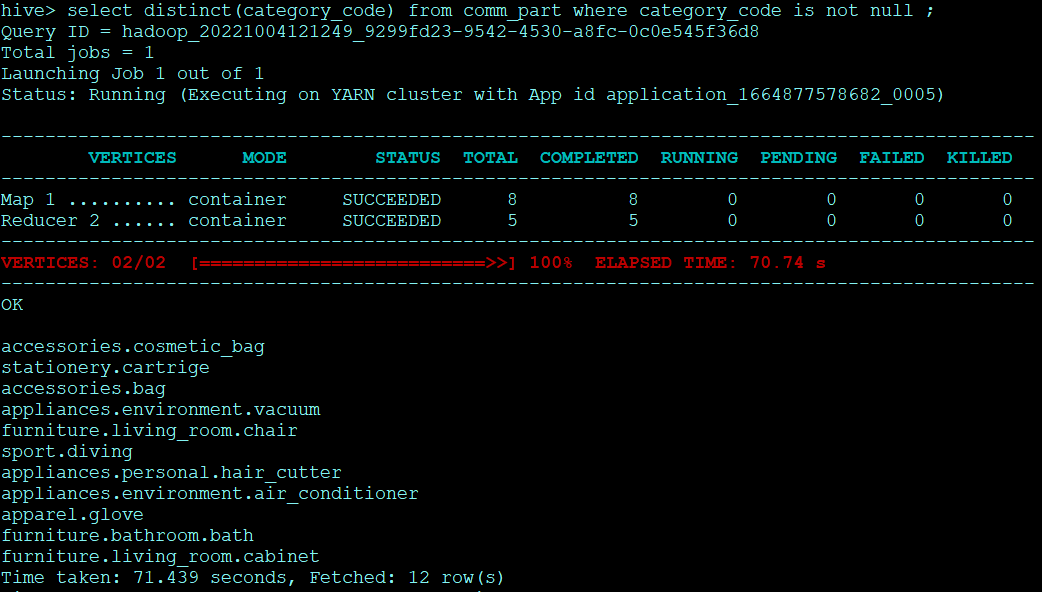
hive> select month , sum(price) as total\_price, sum(price)-lag(sum(price)) over (order by month) from comm\_part where event\_type ='purchase' group by month ;



We can see the difference in the revenue is 319478.47

**4 Find distinct categories of products. Categories with null category code can be ignored.**

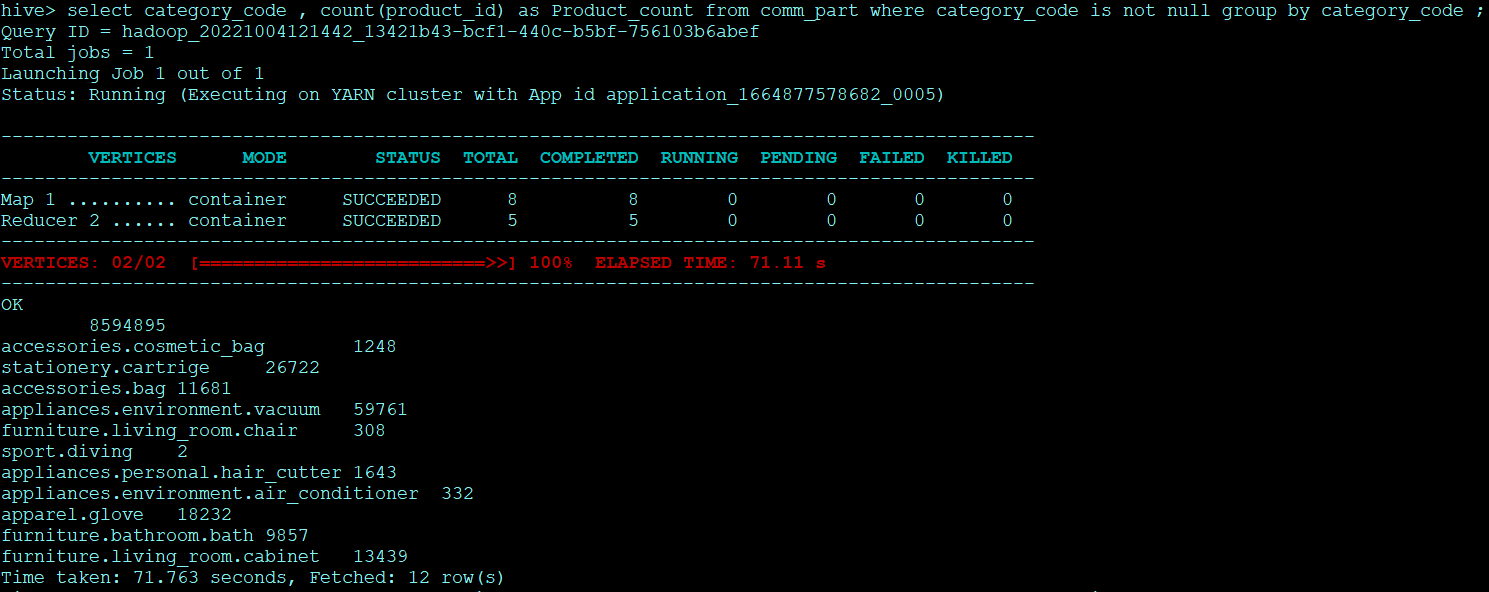
hive> select distinct(category\_code) from comm\_part where category\_code is not null ;



We can see the distinct categories are Furniture, Appliances, Accessories, Apparel, Sport, Stationery

**5 Find the total number of products available under each category.**

hive> select category\_code , count(product\_id) as Product\_count from comm\_part where category\_code is not null group by category\_code ;



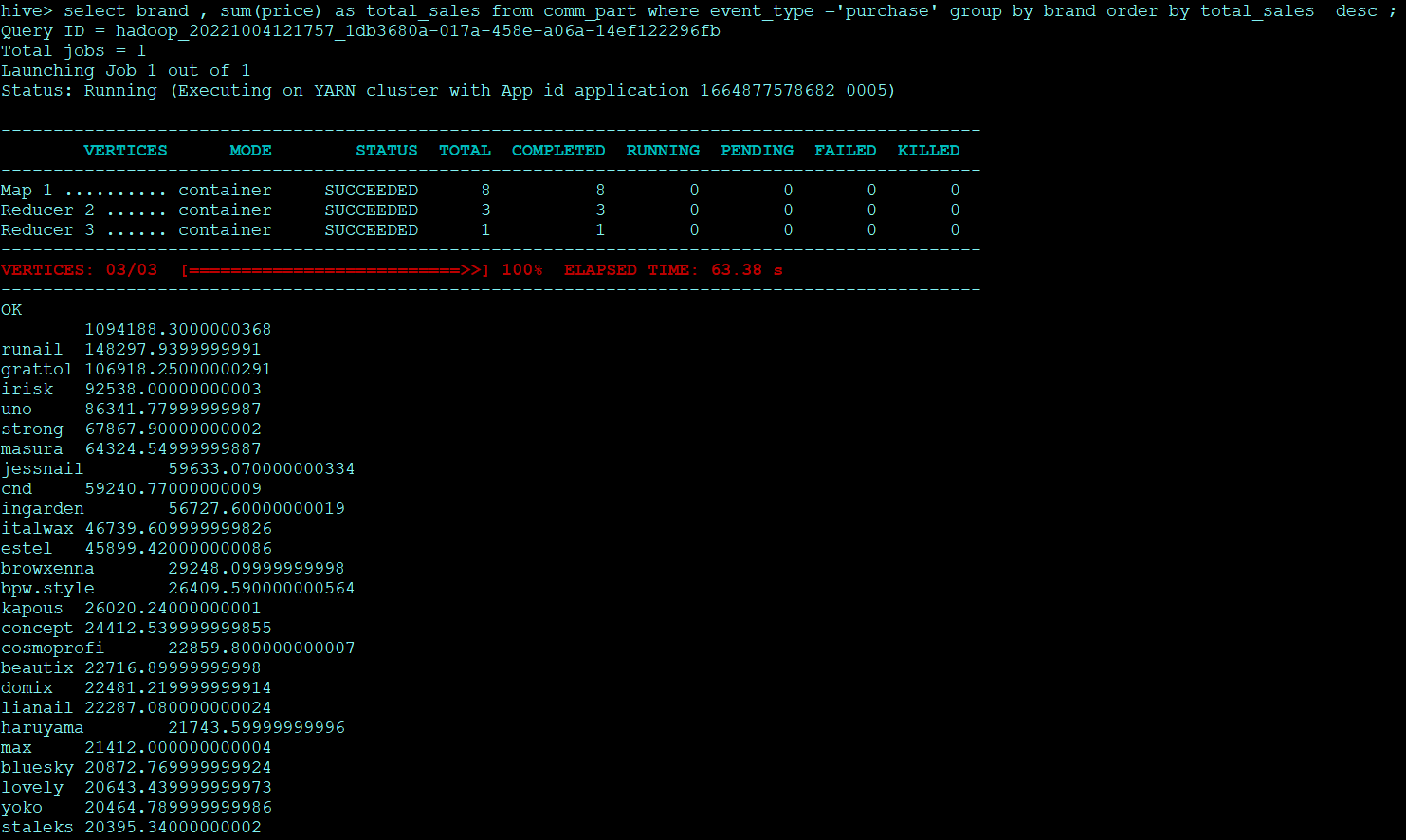
The total number of products under each category is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Category | Sub-category | Number of Sub-Category | Number of Category |
| Appliances | vacuum | 59761 | 61736 |
| hair\_cutter | 1643 |
| air\_conditioner | 332 |
| Stationery |  | 26722 | 26722 |
| Furniture | chair | 308 | 23604 |
| bath | 9857 |
| room.cabinet | 13439 |
| Apparel |  | 18232 | 18232 |
| Accessories | bag | 11681 | 12929 |
| cosmetic\_bag | 1248 |
| Sport | Sport | 2 | 2 |

Appliances 61736, Stationery 26722, Furniture 23604, Apparel 18232, Accessories 12929, Sport 2

6 Which brand had the maximum sales in October and November combined?

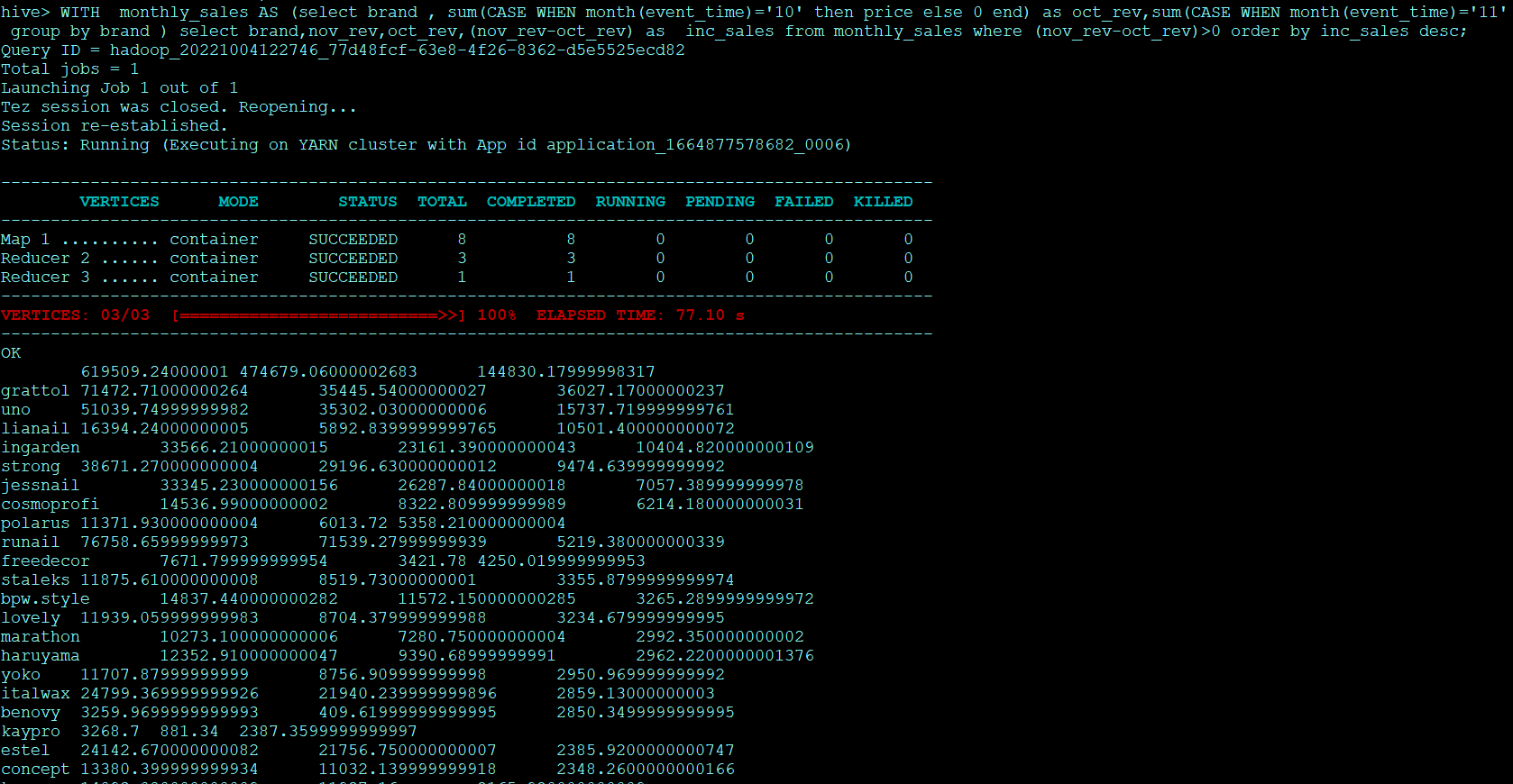
hive> select brand , sum(price) as total\_sales from comm\_part where event\_type ='purchase' group by brand order by total\_sales desc ;

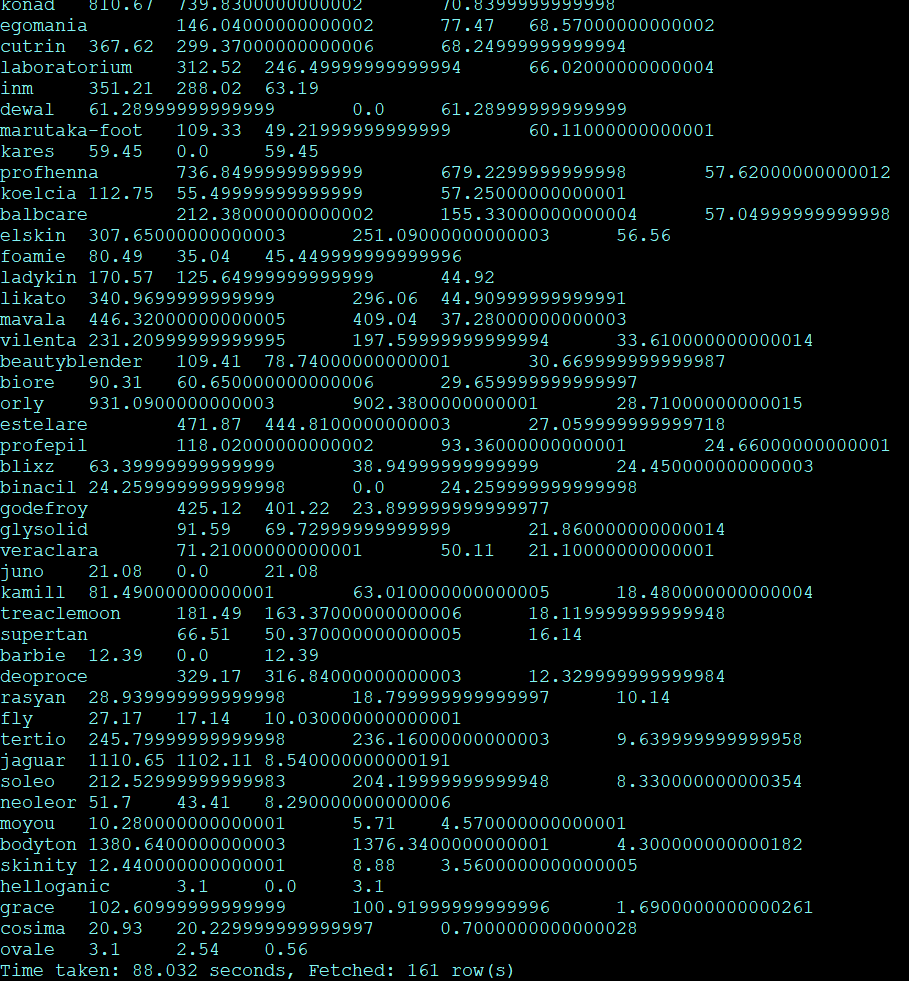


We can see that Runail is the brand with the maximum sales for oct and nov. Total sales is 148297.94

7 Which brands increased their sales from October to November?

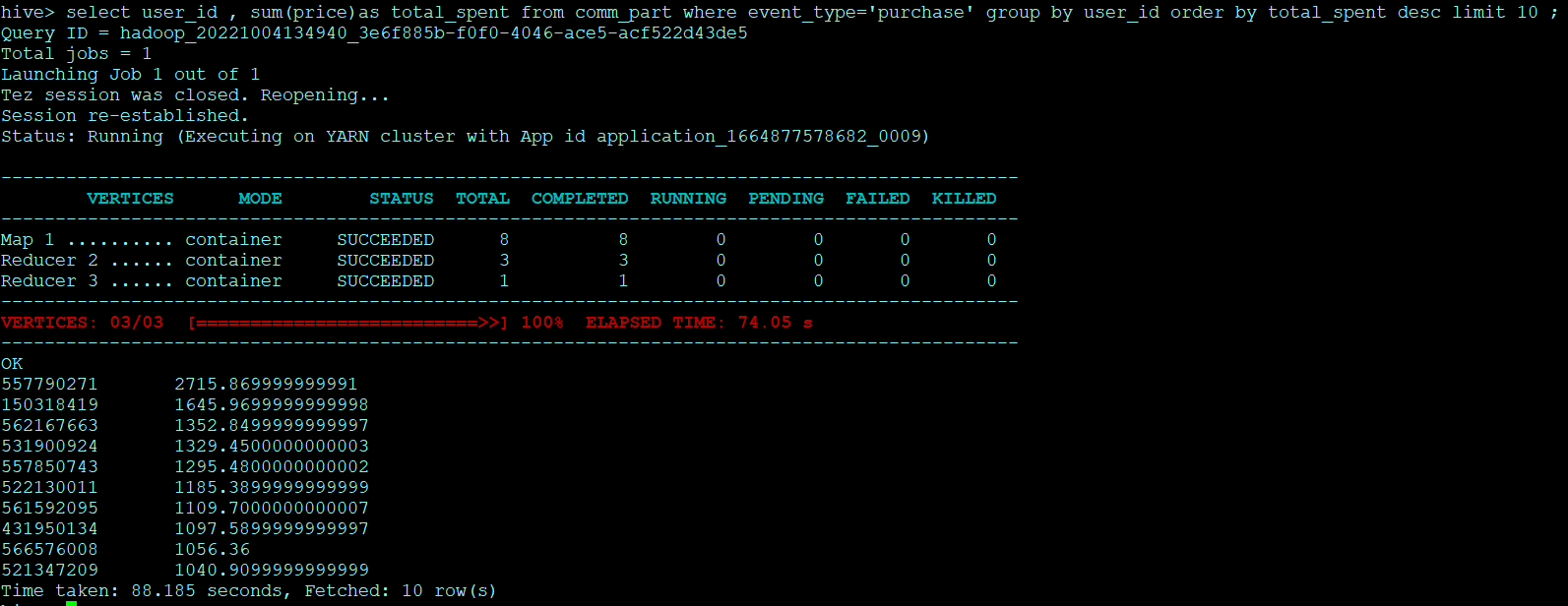
hive> WITH monthly\_sales AS (select brand , sum(CASE WHEN month(event\_time)='10' then price else 0 end) as oct\_rev,sum(CASE WHEN month(event\_time)='11' then price else 0 end ) as nov\_rev from comm\_part where event\_type ='purchase' group by brand ) select brand,nov\_rev,oct\_rev,(nov\_rev-oct\_rev) as inc\_sales from monthly\_sales where (nov\_rev-oct\_rev)>0 order by inc\_sales desc





From the output we can see that 161 brand were able to increase their sales from the month of October to November

8 Your company wants to reward the top 10 users of its website with a Golden Customer plan. Write a query to generate a list of top 10 users who spend the most.  
hive> select user\_id , sum(price)as total\_spent from comm\_part where event\_type='purchase' group by user\_id order by total\_spent desc limit 10 ;



We can see the top 10 users id with total purchases in the output who can be included in the Gold Plan.

Finishing Up

Once we are done, we can drop the databases, quit the hive and then terminate the EMR cluster.