Implement the below blog at your end and send the complete documentation.

https://drive.google.com/file/d/0B\_Qjau8wv1KobUlaOEtfNEtQNkU/view?usp=sharing

3. Output

Documentation containing code and screenshots

Air traffic analysis

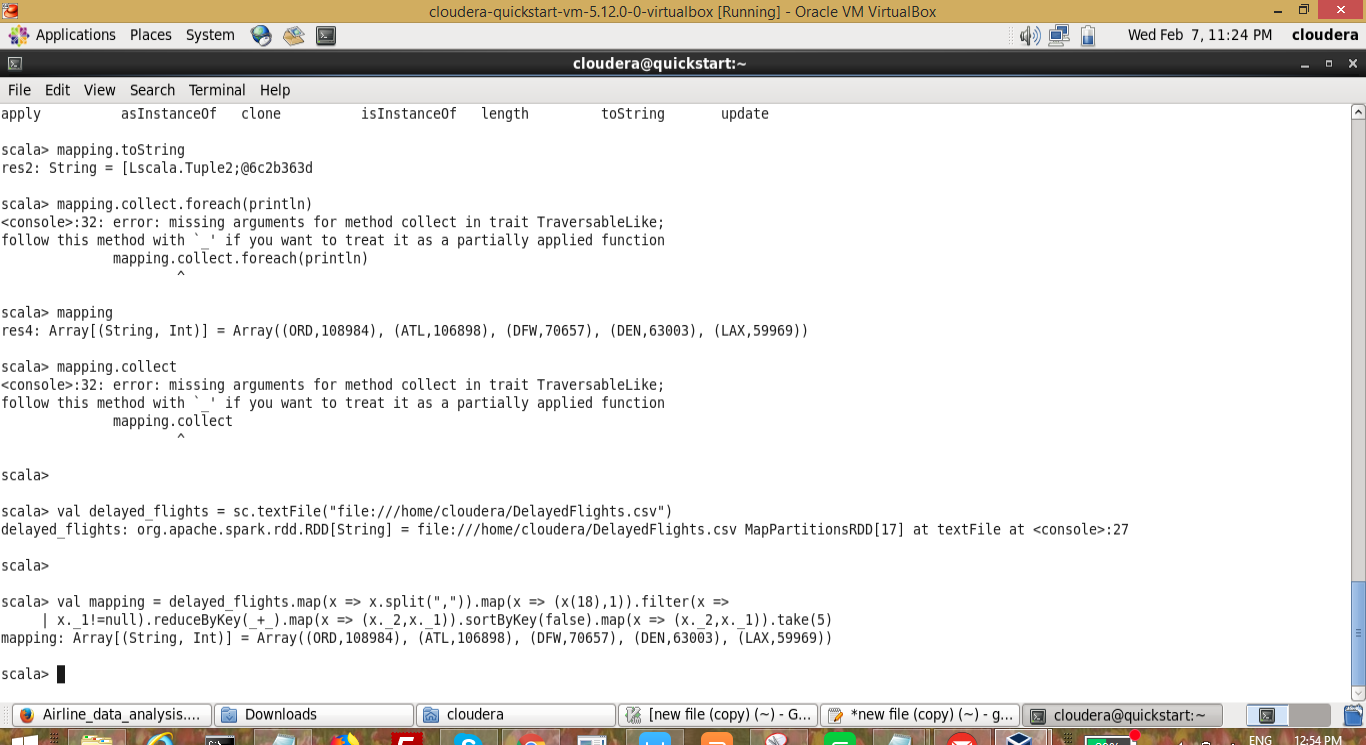
1.

Find out the top 5 most visited destinations.

*val delayed\_flights = sc.textFile("file:///home/cloudera/DelayedFlights.csv")*

*val mapping = delayed\_flights.map(x => x.split(",")).map(x => (x(18),1)).filter(x =>*

*x.\_1!=null).reduceByKey(\_+\_).map(x => (x.\_2,x.\_1)).sortByKey(false).map(x => (x.\_2,x.\_1)).take(5)*



2.

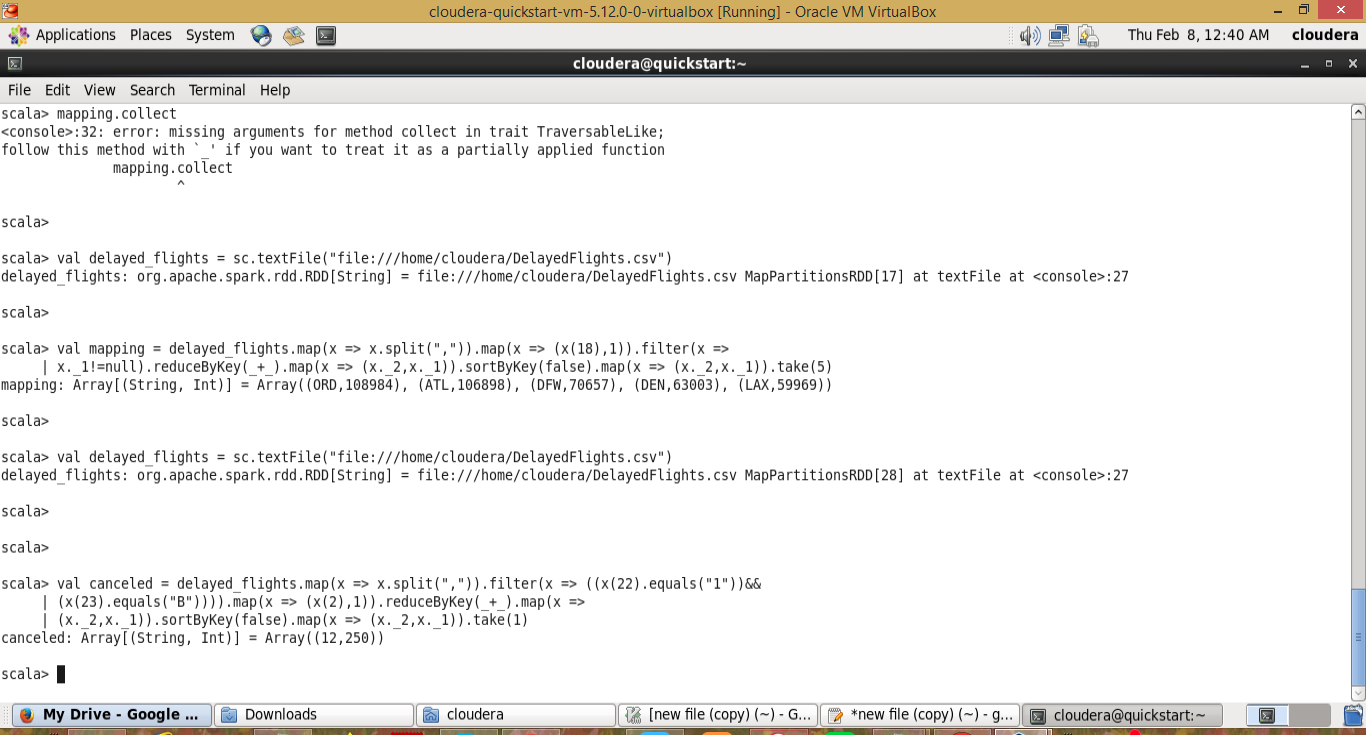
Which month has seen the most number of cancellations due to bad weather?

*val delayed\_flights = sc.textFile("file:///home/cloudera/DelayedFlights.csv")*

*val canceled = delayed\_flights.map(x => x.split(",")).filter(x => ((x(22).equals("1"))&&*

*(x(23).equals("B")))).map(x => (x(2),1)).reduceByKey(\_+\_).map(x =>*

*(x.\_2,x.\_1)).sortByKey(false).map(x => (x.\_2,x.\_1)).take(1)*



3.Top ten origins with the highest AVG departure delay

***val dataset = sc.textFile("file:///home/cloudera/DelayedFlights.csv")***

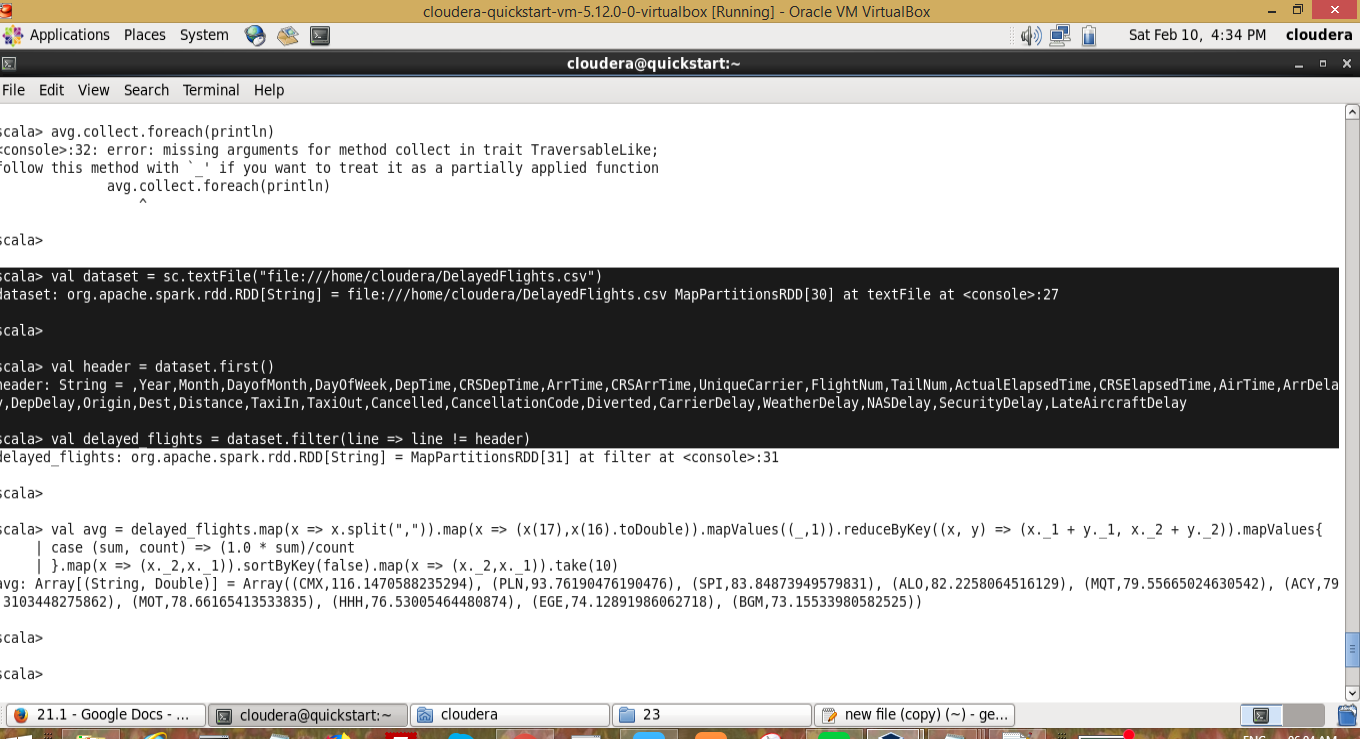
***val header = dataset.first()***

***val delayed\_flights = dataset.filter(line => line != header)***

***val avg = delayed\_flights.map(x => x.split(",")).map(x => (x(17),x(16).toDouble)).mapValues((\_,1)).reduceByKey((x, y) => (x.\_1 + y.\_1, x.\_2 + y.\_2)).mapValues{***

***case (sum, count) => (1.0 \* sum)/count***

***}.map(x => (x.\_2,x.\_1)).sortByKey(false).map(x => (x.\_2,x.\_1)).take(10)***



4.Which route (origin & destination) has seen the maximum diversion?

*val delayed\_flights = sc.textFile("file:///home/cloudera/DelayedFlights.csv")*

*val diversion = delayed\_flights.map(x => x.split(",")).filter(x => ((x(24).equals("1")))).map(x =>*

*((x(17)+","+x(18)),1)).reduceByKey(\_+\_).map(x => (x.\_2,x.\_1)).sortByKey(false).map(x =>*

*(x.\_2,x.\_1)).take(10).foreach(println)*

