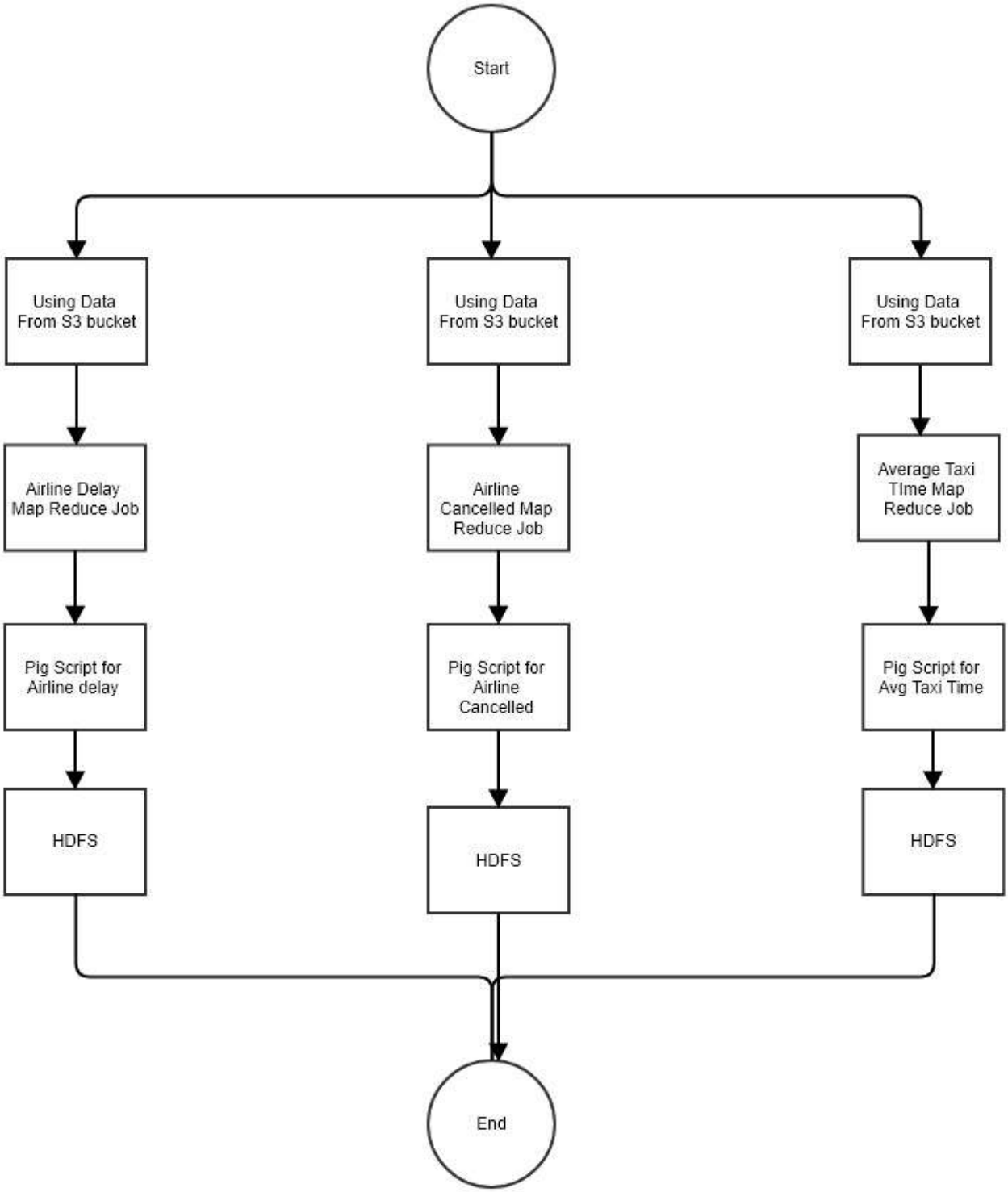
FLIGHT DATA ANALYSIS

Intro to Big Data Project

DHRUMIL VORA (DV253), ABHI SHAH (AVS43)

Workflow


Start
Using Data From S3 bucket
Using Data From S3 bucket
Using Data From S3 bucket
Airline Delay Map Reduce Job
Airline Cancelled Map Reduce Job
Average Taxi TIme Map Reduce Job
Pig Script for Airline delay
Pig Script for Airline Cancelled
Pig Script for Avg Taxi Time
HDFS
HDFS
HDFS
End

Question 1

The 3 airlines with the highest and lowest probability, respectively, for being on schedule

Algorithm

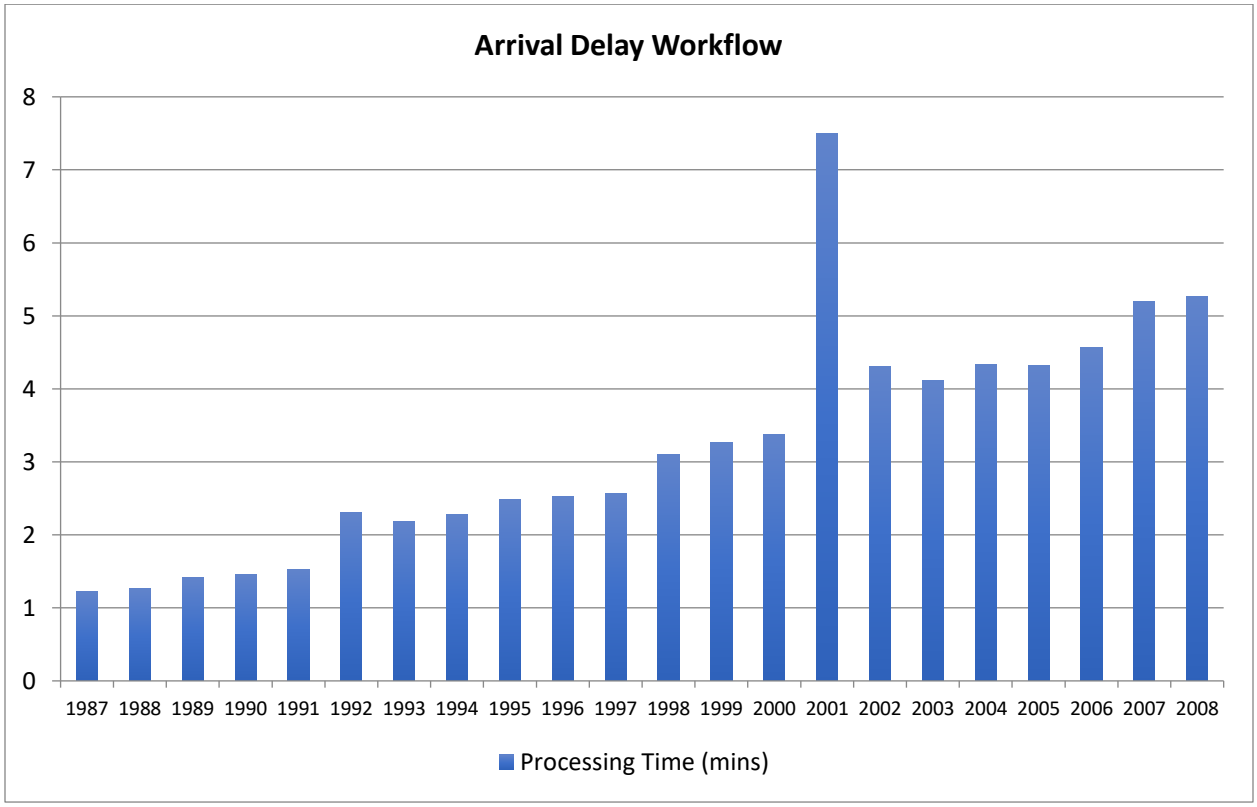
The Map Reduce Job takes input from S3 bucket. The Mapper class splits the csv using ',' as a separator and add it to string array. From the string array based on column index values for Airport and Arrival delay is fetched. The delay had positive, negative numerical values and NAs so will check for values which are not NAs and will generate an intermediate <Key,Value> pair as <Airport, arrDelayInMinutes>. The Reducer class then processes the key value pair by checking the delay if it is positive or negative (Here positive is delay and negative states early departure). So, the Reducer only considers the positive delay and the total count of the values. Probability to be on schedule is calculated by the formula

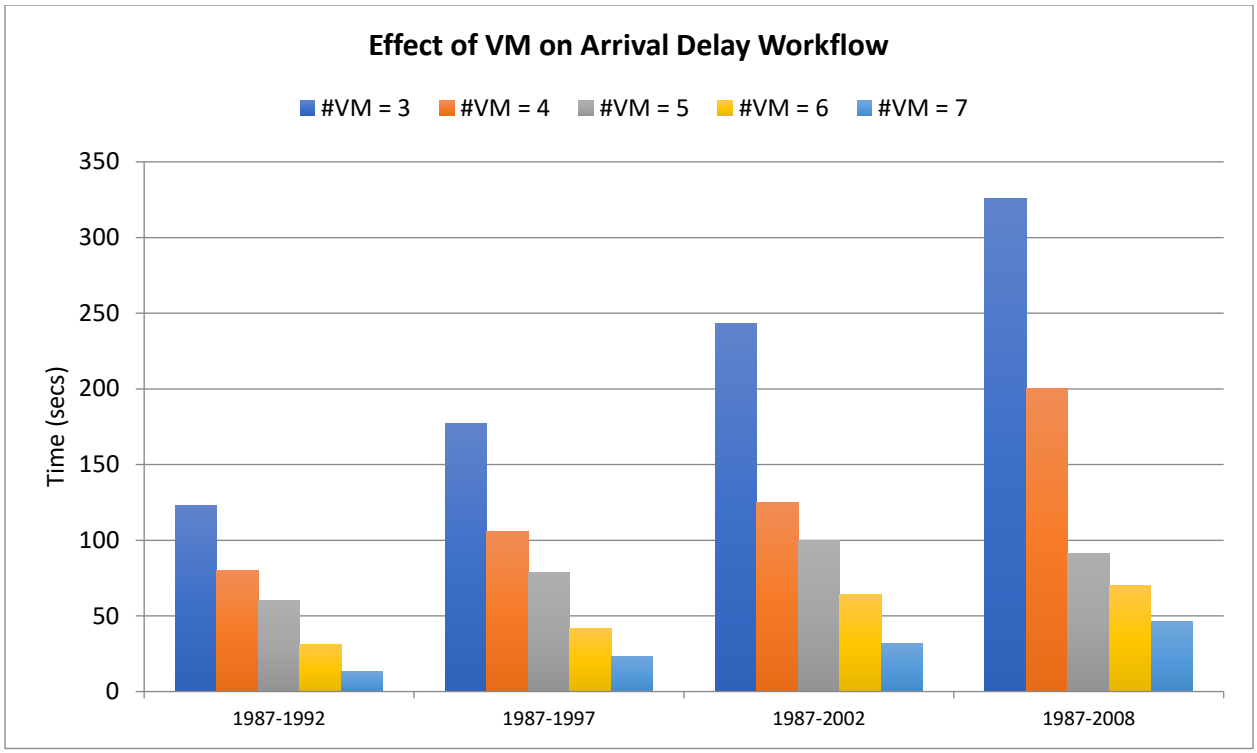
Probability *= number of positive delay/total number of records for that airline*

The final <Key,Value> pair is stored as <Airport, probability, onschedulecount, totalcount>

Pig Script Sorts the result in descending based on probability and picks the top 3 and bottom 3 and provides the output.

Performance Plots




Effect of VM on Arrival Delay Workflow
#VM = 3
#VM = 4
#VM = 5
#VM = 6
#VM = 7
Time (secs)
350
300
250
200
150
100
50
0
1987-1992
1987-1997
1987-2002
1987-2008

Question 2

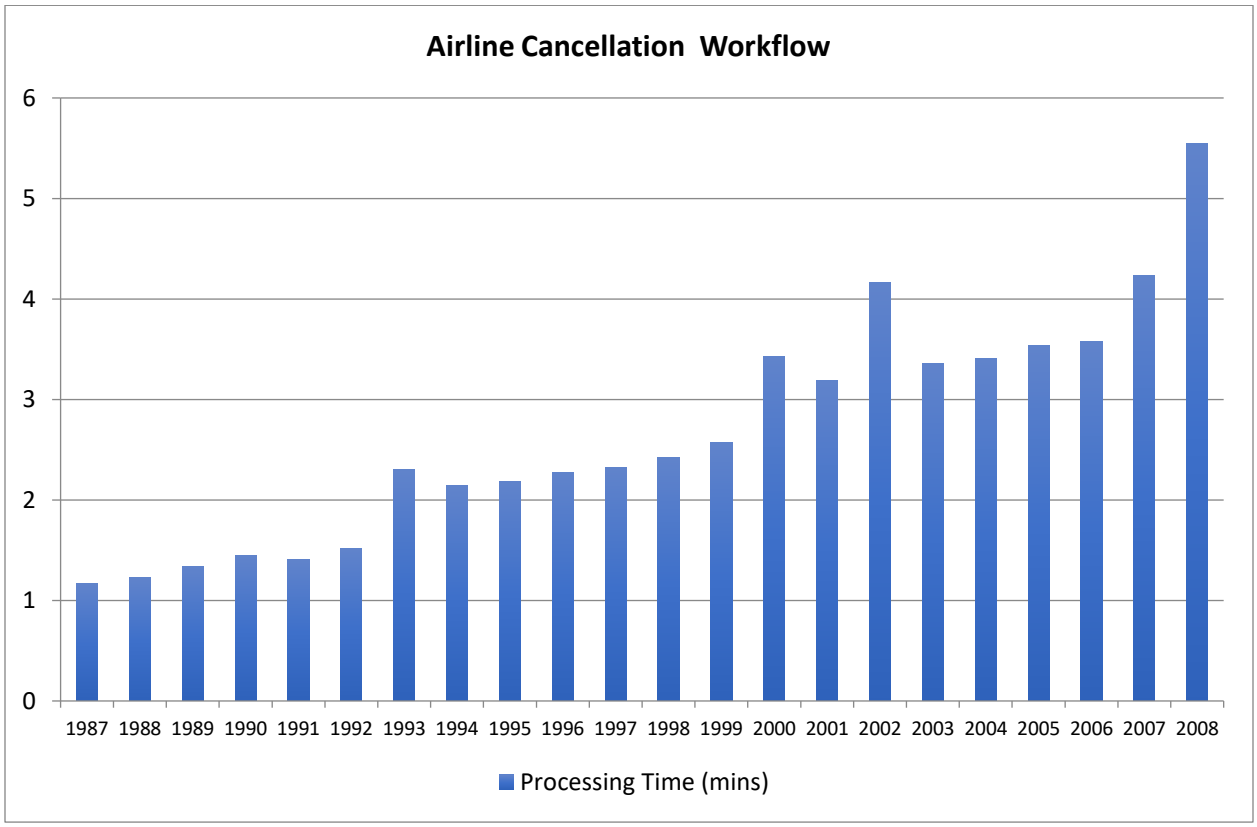
The most common reason for flight cancellations.

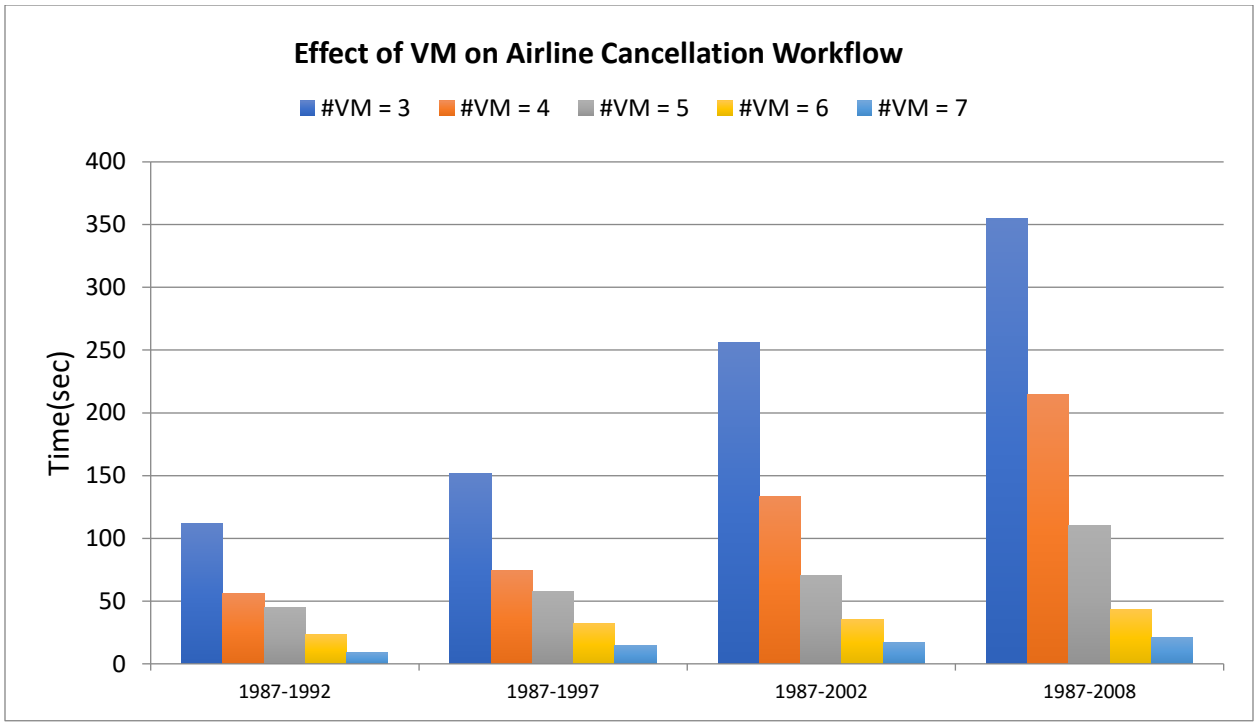
Algorithm

The Map Reduce Job takes input from S3 bucket. The Mapper class splits the csv using ',' as a separator and add it to string array. From the string array based on column index values for CancellationCode, Year is fetched and intermediate <Key,Value> pair is written as <Year, CancellationCode >. The Reducer class then processes the key value pair by calculating the sum for each CancellationCode. So, the Reducer then writes the final <Key,Value> pair is stored as < CancellationCode, sumofCancellationCode >

Pig Script Sorts the result in descending based on sumofCancellationCode.

Performance Plots




Effect of VM on Airline Cancellation Workflow
#VM = 3
#VM = 4
#VM = 5
#VM = 6
#VM = 7
Time(sec)
400
350
300
250
200
150
100
50
0
1987-1992
1987-1997
1987-2002
1987-2008

Question 3

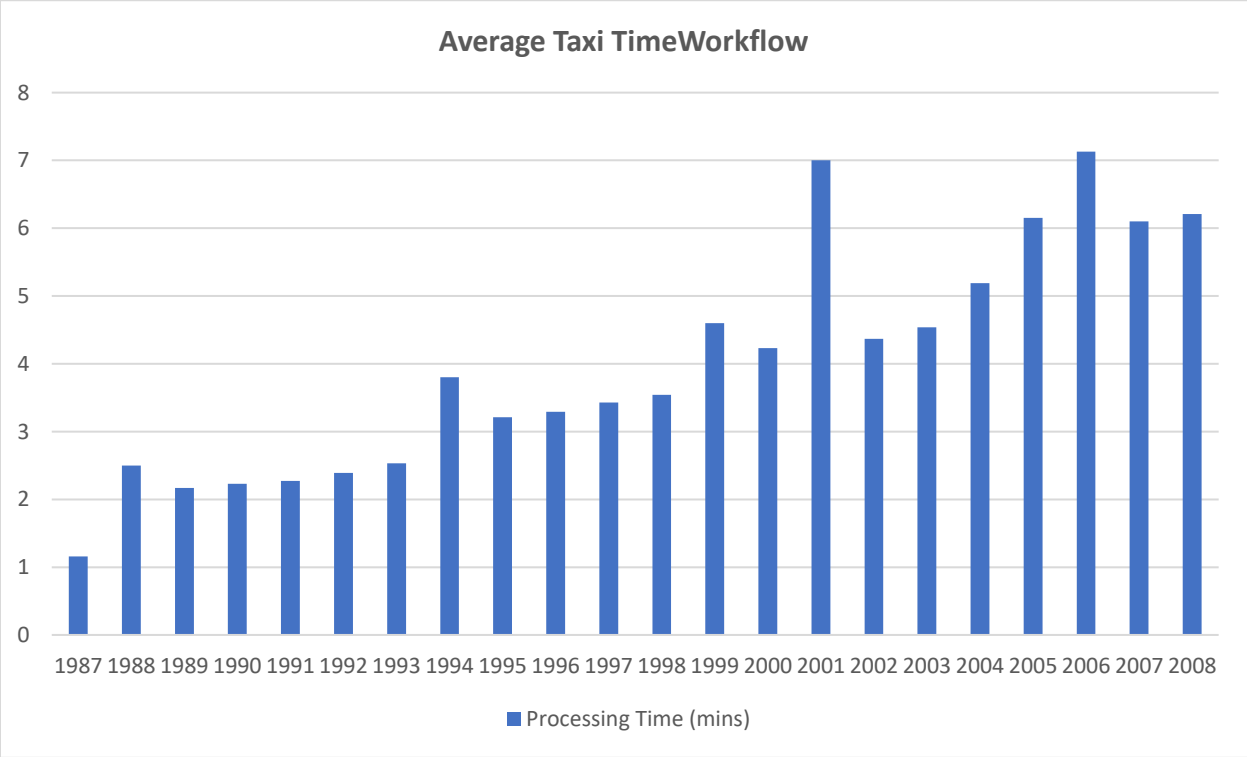
The 3 airports with the longest and shortest average taxi time per flight (both in and out), respectively.

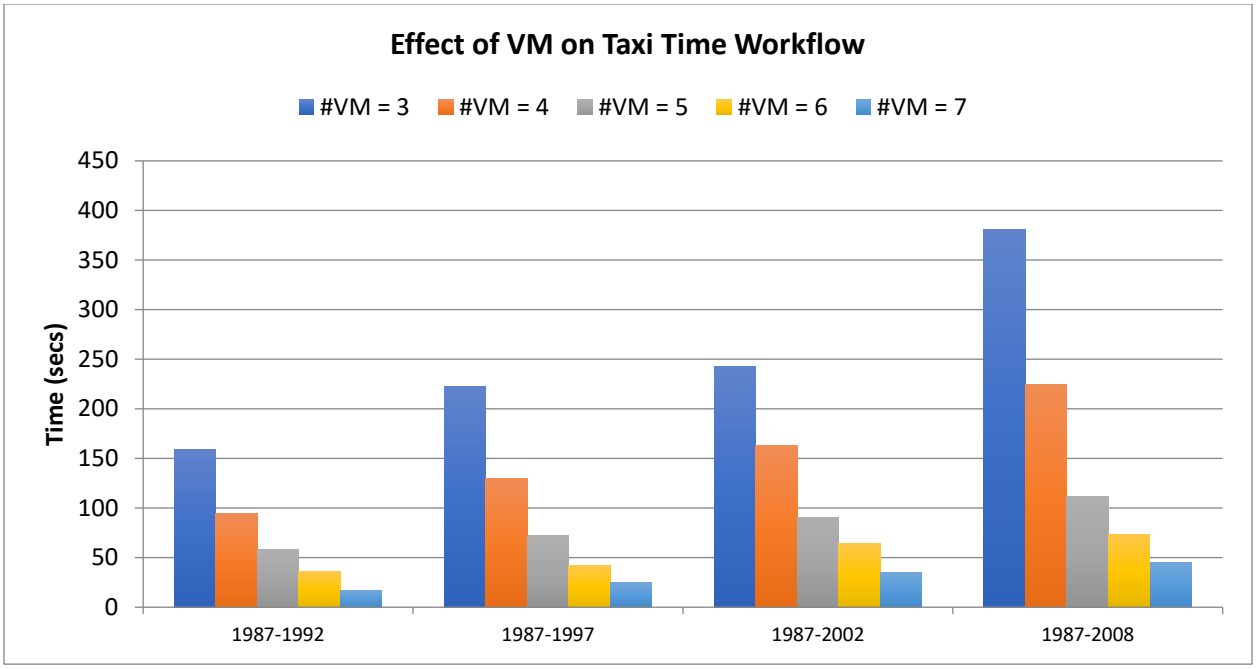
Algorithm

The Map Reduce Job takes input from S3 bucket. The Mapper class splits the csv using ',' as a separator and add it to string array. From the string array based on column index values for AirportDeparture, TaxitOuttime, AirportArrival and TaxiIntime is fetched and intermediate <Key,Value> pair is written as <AirportDeparture \t IN, TaxitOuttime > and < AirportArrival \t OUT , TaxiIntime >. The Reducer class then processes the key value pair by calculating the average for each unique key taxiintime or taxiouttime. So, the Reducer then writes the final <Key,Value> pair is stored as < AirportDeparture \t IN | AirportArrival \t OUT , TaxitOuttime | TaxiIntime >

Pig Script Sorts the result in descending based on averagetaxitime and picks top 3 and bottom 3 from the result.

Performance Plots



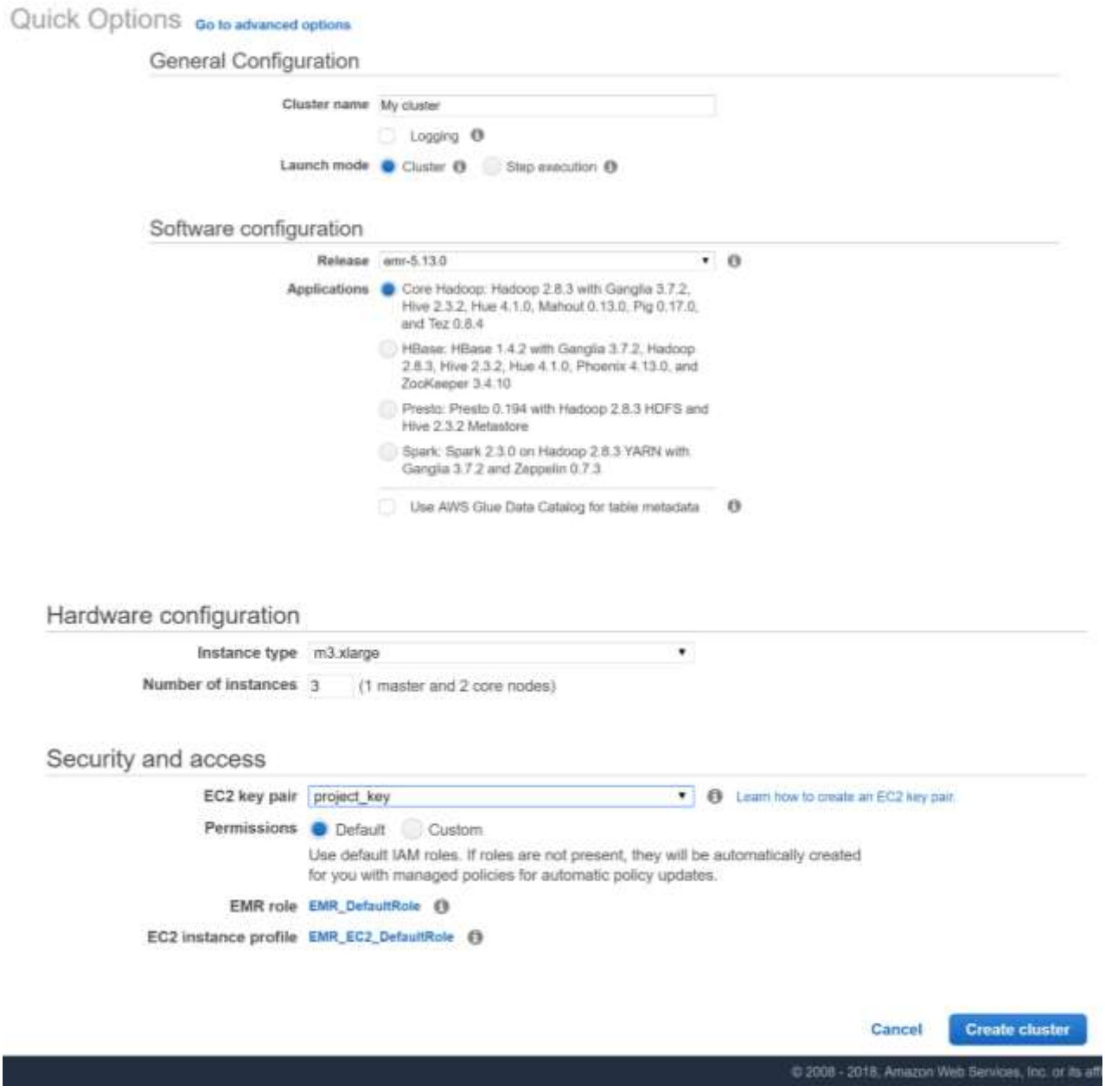


Analyzing Performance Plot

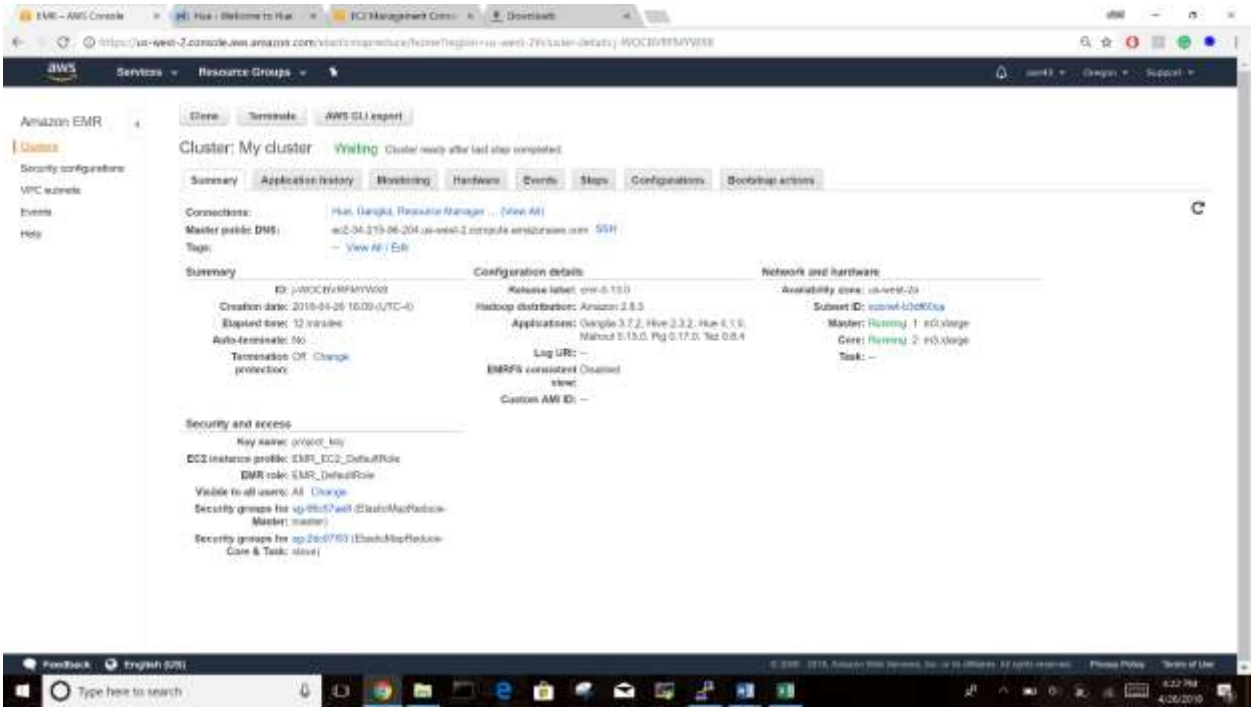
A common trend is induced after visualizing the performance results for all workflow. As all the individual runs were cumulative data. i.e. with all subsequent runs uses all files previously added and new file. The graph is almost linear except a spike in year 2001. This spike is caused due to an experiment we made to load test the cluster. We ran all three workflows in parallel and wanted to examine the results. Clearly, running 3x the usual load has increased execution time of each workflow by at least 2x times. This explains the reliability and concurrency handling of Hadoop.

STEPS to run the jar file

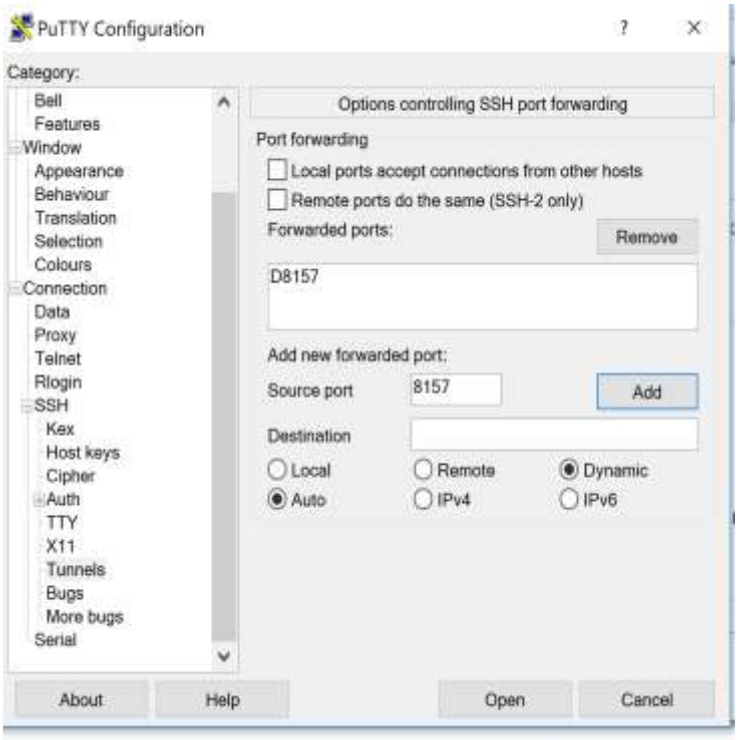
- Go to aws console home ->EMR->create cluster
- Configure cluster as below screenshots



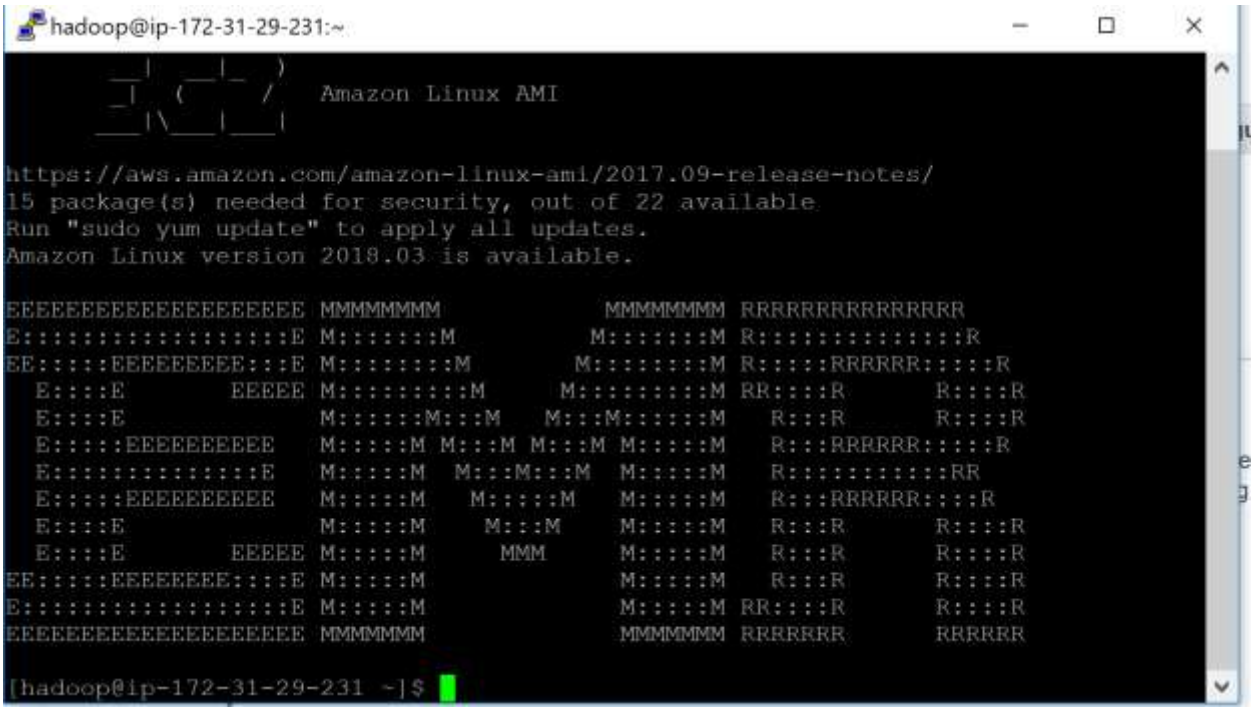
- Once the cluster is started Connect it with ssh using putty



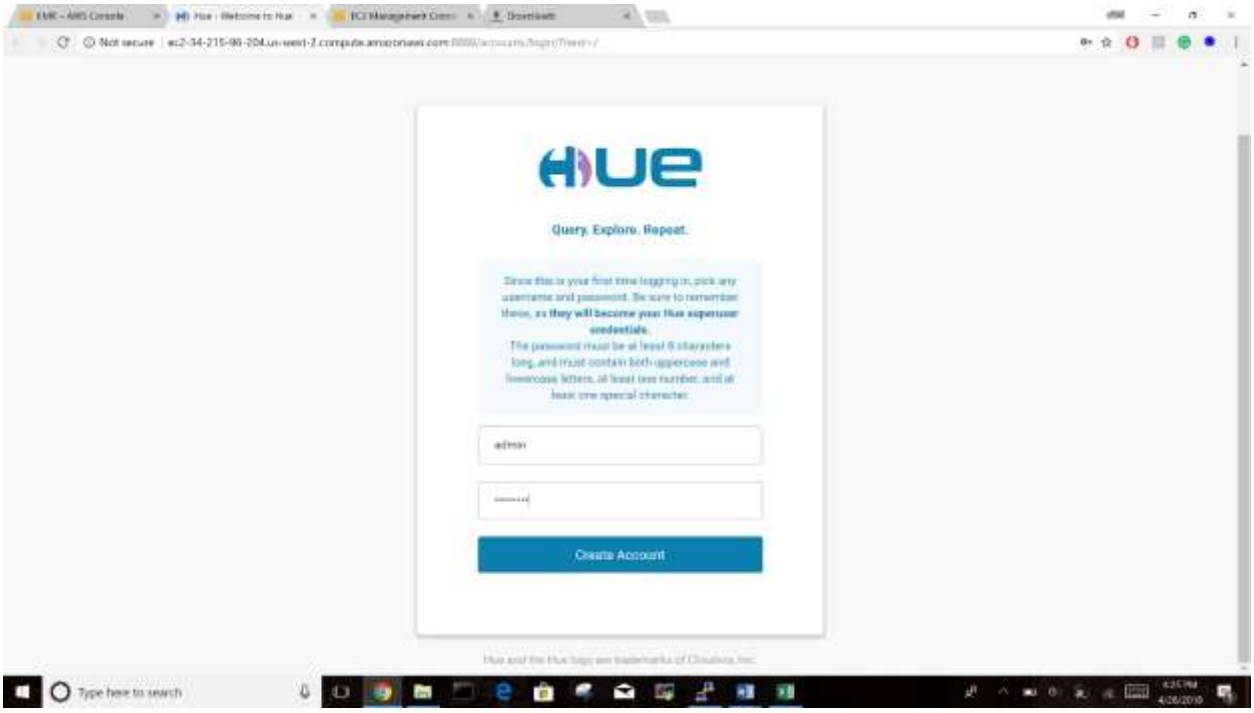
- Open putty paste the public dns of cluster and Tunnel with port 8157 as shown below

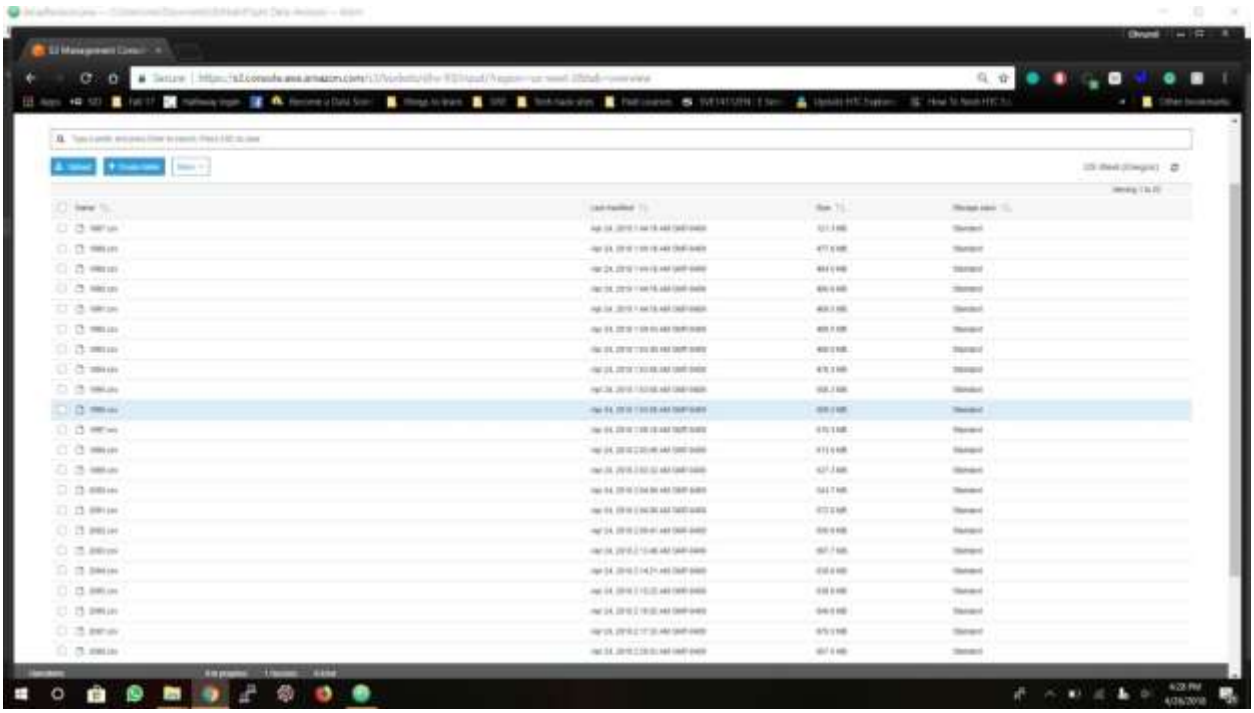


- Click open and you are now connected to EMR

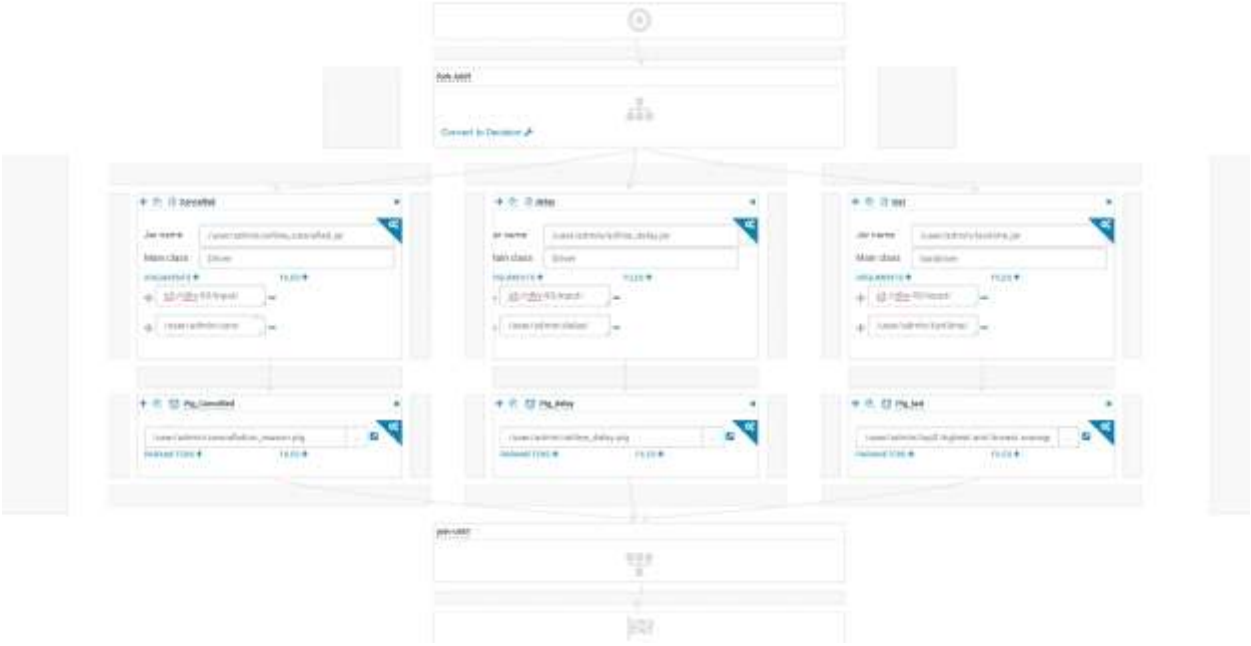


- Create account in HUE and place the input files to S3 bucket


Query. Explore. Repeat.
admin
Create Account
Type here to search



- Design the workflow as below upload the jar files and Pig Script as below.



- Save and Run the workflow wait for some time
- Once the execution is completed the output files are generated at provided location
  In this case it is **/user/admin/**