Task 2 – Hadoop Confidence Test

The following will demonstrate that you are able to maneuver around different parts of the cluster and operate both within Ambari and without it.

HDFS

Create a folder called consumercomplaints.

Use the following [link](https://s3.inmar.com/dbaexercise/Consumer_Complaints.csv) to download a CSV file from S3 and place it into your new Hadoop cluster in the folder you created above.

Change permissions on the file so that it is readable by the world, but only can be updated by the owner.

1)   Downloaded Consumer-Complaints data file from the given link and uploaded it to HDFS dir Consumer Complaints where the file should be modified by the owner only and the world will be able to read it.

Commands used to load data to HDFS:

a) Created user Mahesh in linux using > useradd  Mahesh

b) Copied file to  /home/Mahesh using Winscp

c)  Copied file from local file system (/home/Mahesh) to HDFS by creating user space(dir) for Mahesh in HDFS and changed the permission of the file using below commands

>hdfs  dfs –mkdir  /user/Mahesh/

>hdfs  dfs –copyFromLocal / -put home/Mahesh/consumercomplaints.csv  /user/Mahesh

>hdfs  dfs –chmod 744 /user/Mahesh/consumercomplaints

d)  as per the requirement moved file from /user/Mahesh to /consumercompalints directory

>hdfs  dfs –mv /user/Mahesh/consumercomplaints.CSV  /consumercomplaints

HIVE

Utilizing the field names provided in the CSV file above, create an external table over the file in the consumercomplaints folder called consumercomplaints. Create another table called consumercomplaints\_partitioned and partition the table by state.

1. Created table named consumercomplaints in hive using below script

CREATE EXTERNAL TABLE IF NOT EXISTS consumercomplaints

/(Date\_received timestamp,Product string,

/Sub-product string,Issue string,Sub-issue string,

/Consumer\_Complaint\_narrative string,Company\_public\_response string,

/Company string,State string,ZIP\_code int,

/Tags string,Consumer\_consent\_provided string,

/Submitted\_via varchar(20),Date\_sent\_to\_company timestamp,

/Company\_response\_to\_consumer string,Timely\_response varchar(5),

/Consumer\_disputed varchar(8),Complaint\_ID int)

/ROW FORMAT DELIMITED FIELDS TERMINATED BY '\001'

/LOCATION '/user/consumercomplaints';

1. Loaded data to the Hive table using the below script

LOAD DATA LOCAL INPATH '/home/pboddeti/Consumer\_Complaints.csv' /INTO

/TABLE consumercomplaints;

1. Created another ext table using the below script used partitions

CREATE EXTERNAL TABLE IF NOT EXISTS consumercomplaints\_partitioned

/(Date\_received timestamp,Product string,

/Sub-product string,Issue string,Sub-issue string,

/Consumer\_Complaint\_narrative string,Company\_public\_response string,

/Company string,ZIP\_code int,

/Tags string,Consumer\_consent\_provided string,

/Submitted\_via varchar(20),Date\_sent\_to\_company timestamp,

/Company\_response\_to\_consumer string,Timely\_response varchar(5),

/Consumer\_disputed varchar(8),Complaint\_ID int)

/PARTITIONED BY (STATE STRING)

/ROW FORMAT DELIMITED

/FIELDS TERMINATED BY ','

/STORED AS TEXTFILE

/LOCATION '/user/consumercomplaints1';

Task 3 - Streaming Architecture w/ Hive

Hive

1. Store the twitter data called sample\_twitter\_data in HDFS. You can use [this link](https://s3.inmar.com/dbaexercise/sample_twitter_data.txt) to download it. The data is in json format and should not be altered.

Once the data is in HDFS, create an hcat/hive schema to be able to answer the following question: What are all the tweets by the twitter user "Aimee\_Cottle"? You will need to provide the query that answers this question

a:Store the twitter data called sample\_twitter\_data in HDFS

[hdfs@ip-10-0-0-249 ec2-user]$ hadoop fs -put /home/sample\_twitter\_data /sample\_twitter\_data

[hdfs@ip-10-0-0-249 ec2-user]$ hadoop fs -ls /sample\_twitter\_data

-rw-r--r-- 3 hdfs hdfs 2766 2018-06-26 16:35 /sample\_twitter\_data

b:create an hcat/hive schema

hive (mahesh)> CREATE EXTERNAL TABLE tweets (

> createddate string,

> geolocation string,

> tweetmessage string,

> `user` struct<geoenabled:boolean, id:int, name:string, screenname:string, userlocation:string>)

> ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe';

OK

Time taken: 1.117 seconds

hive (mahesh)> LOAD data local inpath '/home/sample\_twitter\_data' into table tweets;

Loading data to table mahesh.tweets

Table mahesh.tweets stats: [numFiles=2, numRows=0, totalSize=5532, rawDataSize=0]

OK

Time taken: 2.884 seconds

hive (mahesh)> SET hive.support.sql11.reserved.keywords=false;

hive (mahesh)> SELECT DISTINCT tweetmessage, user.name, createddate

> FROM tweets WHERE user.name = 'Aimee\_Cottle'

> ORDER BY createddate;

Below are the screen shots taken

General Administration:

[hdfs@ip-10-0-0-249 home]$ hdfs dfsadmin -report

Configured Capacity: 138211175936 (128.72 GB)

Present Capacity: 118458391807 (110.32 GB)

DFS Remaining: 113832044408 (106.01 GB)

DFS Used: 4626347399 (4.31 GB)

DFS Used%: 3.91%

Under replicated blocks: 91

Blocks with corrupt replicas: 0

Missing blocks: 0

Missing blocks (with replication factor 1): 0

-------------------------------------------------

Live datanodes (3):

Sample pi job:

[hdfs@ip-10-0-0-249 hdp]$ hadoop jar /usr/hdp/2.6.3.0-235/hadoop-mapreduce/hadoop-mapreduce-examples.jar pi

Usage: org.apache.hadoop.examples.QuasiMonteCarlo <nMaps> <nSamples>

hadoop jar /usr/hdp/2.6.3.0-235/hadoop-mapreduce/hadoop-mapreduce-examples.jar pi 5 10

Shuffle Errors

BAD\_ID=0

CONNECTION=0

IO\_ERROR=0

WRONG\_LENGTH=0

WRONG\_MAP=0

WRONG\_REDUCE=0

File Input Format Counters

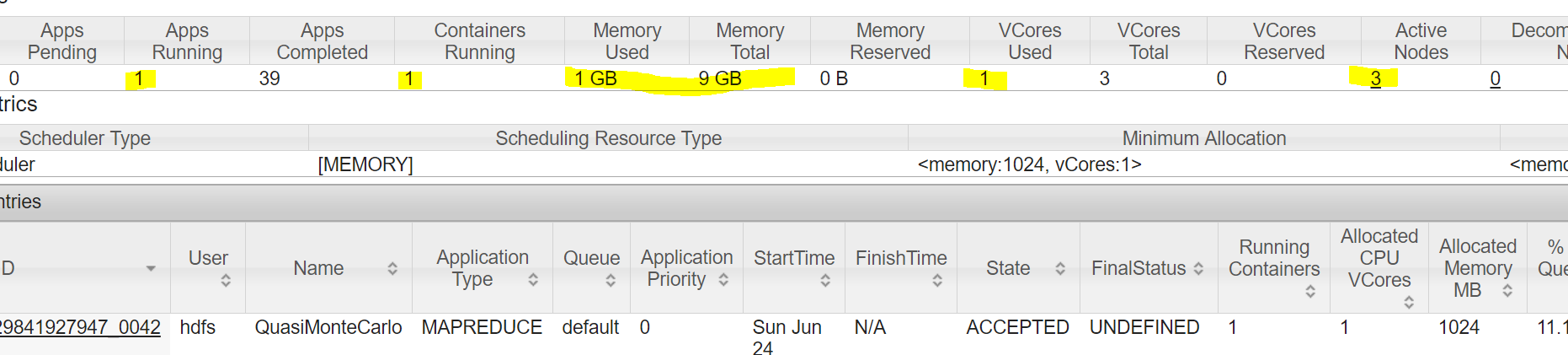
Bytes Read=590

File Output Format Counters

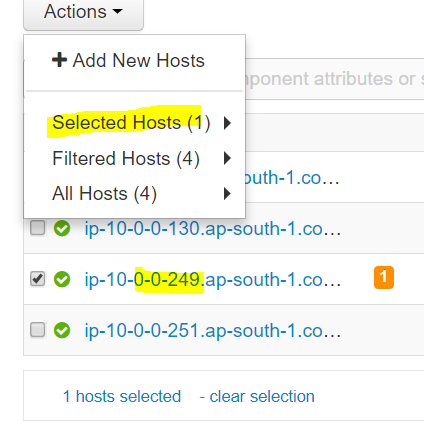
Bytes Written=97

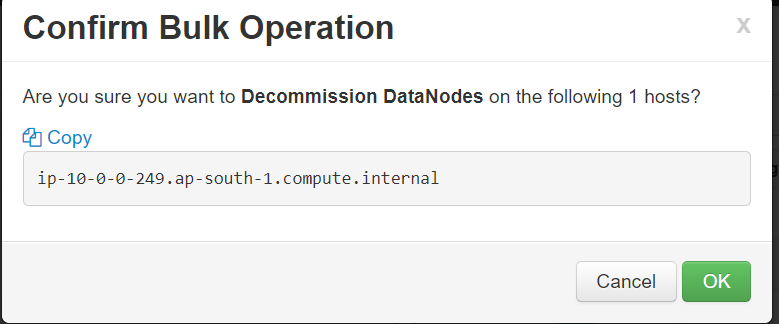
Job Finished in 49.282 seconds

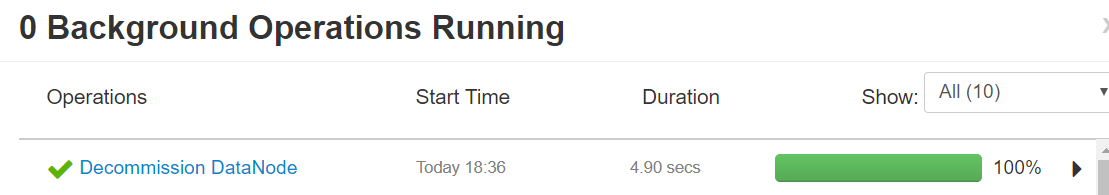
Estimated value of Pi is 3.28000000000000000000

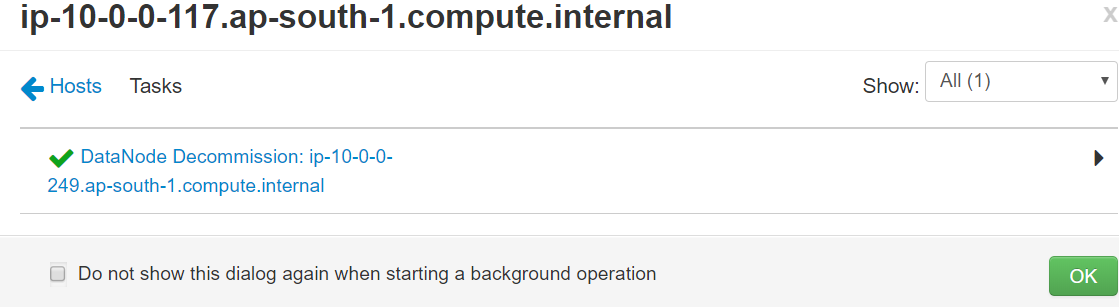


Decommisioning 249 datanode:









Hdfs dfsadmin –report

Decommissioning datanodes (1):

Name: 10.0.0.249:50010 (ip-10-0-0-249.ap-south-1.compute.internal)

Hostname: ip-10-0-0-249.ap-south-1.compute.internal

Decommission Status : Decommission in progress

