



Big Data

Lab Manual

**Department of Computer Science and
Engineering
The NorthCap University, Gurugram**

Big Data Lab Manual

CSL 311

Dr. Anuradha

Dr. Meghna Sharma

Ms. Meghna Luthra



Department of Computer Science and Engineering

NorthCap University, Gurugram- 122001, India

Session 2020-21

Published by:

School of Engineering and Technology

Department of Computer Science & Engineering

The NorthCap University Gurugram

- Laboratory Manual is for Internal Circulation only**

© Copyright Reserved

*No part of this Practical Record Book may be
reproduced, used, stored without prior permission of The NorthCap University*

Copying or facilitating copying of lab work comes under cheating and is considered as use of unfair means. Students indulging in copying or facilitating copying shall be awarded zero marks for that particular experiment. Frequent cases of copying may lead to disciplinary action. Attendance in lab classes is mandatory.

Labs are open up to 7 PM upon request. Students are encouraged to make full use of labs beyond normal lab hours.

PREFACE

Big Data Lab Manual is designed to meet the course and program requirements of NCU curriculum for B.Tech III year students of CSE branch. The concept of the lab work is to give brief practical experience for basic lab skills to students. It provides the space and scope for self-study so that students can come up with new and creative ideas.

The Lab manual is written on the basis of “teach yourself pattern” and expected that students who come with proper preparation should be able to perform the experiments without any difficulty. Brief introduction to each experiment with information about self-study material is provided. The laboratory exercises provide an in-depth understanding of terminologies and the core concepts behind big data problems, applications, systems and the techniques, that underlie today big data computing technologies. It provides an introduction to some of the most common frameworks such as Apache Spark, Hadoop, MapReduce, Large scale data storage technologies. Finally, the students would apply the big data analytics knowledge for batch and stream processing of big data on various case studies. At the start of each experiment a question bank for preparation and practice is suggested which may be used to test the basic understanding of the students about the experiment. Students are expected to come thoroughly prepared for the lab. General disciplines, safety guidelines and report writing are also discussed.

The lab manual is a part of curriculum for the TheNorthCap University, Gurugram. Teacher's copy of the experimental results and answer for the questions are available as sample guidelines.

We hope that lab manual would be useful to students of CSE, IT, ECE and BSc branches and author requests the readers to kindly forward their suggestions / constructive criticism for further improvement of the work book.

Author expresses deep gratitude to Members, Governing Body-NCU for encouragement and motivation.

Authors
The NorthCap University
Gurugram, India

CONTENTS

S.N.	Details	Page No.
	Syllabus	
1	Introduction	
2	Lab Requirement	
3	General Instructions	
4	List of Experiments	
5	List of Flip Assignment	
6	List of Projects	
7	Rubrics	
8	Annexure 1 (Format of Lab Report)	
9	Annexure 2 (Format of Lab Certificate)	

SYLLABUS

1. Department:	Department of Computer Science and Engineering						
2. Course Name: Big Data	3. Course Code	4. L-T-P	5. Credits				
	CSL311	2- 0-4	4				
6. Type of Course (Check one):	Programme Core <input type="checkbox"/> Programme Elective <input checked="" type="checkbox"/> Open Elective <input type="checkbox"/>						
7. Pre-requisite(s), if any:	Data Engineering						
8. Frequency of offering (check one):	Odd <input type="checkbox"/>	Even <input checked="" type="checkbox"/>	Either semester <input type="checkbox"/>	Every semester <input type="checkbox"/>			
9. Brief Syllabus:	<p>Characteristics of big data, Big Data and its importance, Challenges of big data, Big data applications, Hadoop Architecture, HDFS, Common Hadoop Shell commands, Introduction to GCP, GCP Fundamentals, quick labs, setting up GCP, GCP services, GCP Regions & Zones,IAM, Billing, Resource Hierarchy, Google Compute Services, Google Storage Services, Introduction to Big query and Machine Learning, Introduction to App Engine. Anatomy of File Write and Read NameNode, Secondary NameNode and DataNode, Hadoop Technologies , Understanding Inputs and Outputs of MapReduce,MRjobs,multistep MRjobs, Pig,Hive scripting, Getting Started with Spark, Setting up Python with Spark, RDD, Functional Programming, PySpark Set-up,Running Spark on a Cluster, SparkSQL, Spark DataFrame Basics, Spark Graph X, Collaborative Filtering for Recommender Systems, Natural Language Processing in Spark, Real-time analytics with Spark Streaming.</p>						
Total lecture, Tutorial and Practical Hours for this course (Take 15 teaching weeks per semester): 90 hours The class size is maximum 30 learners.							
Lectures: 45 hours	Tutorials : 0 hours	Practice					
10. Course Outcomes (COs)	On successful completion of this course students will be able to:						
CO 1	Deploy big data architecture for data analytics on cloud.						
CO 2	Understand various Big Data tools and terminologies and where they fit in the grand scheme of things.						
CO 3	Perform data preprocessing on large datasets.						
CO 4	Map big data concepts with potential use in a corporate environment.						
CO 5	Design predictive analytics projects on big data.						
11. UNIT WISE DETAILS							
Units: 4							

Unit Number: 1	Title: Introduction	No. of hours:17
Content Summary: Introduction, Characteristics of Big Data, Importance of Big data, Types of Big data, Structured vs Unstructured data, Challenges of Big Data, Big data Applications, Sequential Vs Parallel Memory Models, Introduction to GCP, GCP Fundamentals, quick labs, setting up GCP, GCP services, GCP Regions & Zones, IAM, Billing, Resource Hierarchy, Google Compute Services, Google Storage Services, Introduction to Big query and Machine Learning, Introduction to App Engine.		
Unit Number: 2	Title: Hadoop,Map Redue Architecture & Programming	No. of hours:25
Content Summary: Apache Hadoop and Hadoop EcoSystem, Understanding Inputs and Outputs of MapReduce; Hadoop Architecture- Hadoop Storage, HDFS, Common Hadoop Shell commands, NameNode, Secondary NameNode, and DataNode, Hadoop MapReduce paradigm, Map and Reduce tasks, Job, Task trackers, Introduction to MapReduce, Architecture of Map-Reduce, Understanding the concept of Mappers & Reducers MapReduce: Word Count Example, Phases of a MapReduce program, Break down tasks into Map and Reduce Phases, Optimize Map Reduce using a Combiner Word Count Example, Constraints in Using Reducer as Combiner, Phases of a Map Reduce program, and Data-types in Hadoop Map Reduce Driver.		
Unit Number: 3	Title: Pig & Hive	No. of hours: 18
Content Summary: Pig and HBase, Hive: Hive Shell, Hive Services, Hive Megastore, Comparison with Traditional Databases, HiveQL, Tables, Querying Data and User Defined Functions		
Unit Number: 4	Title: Spark	No. of hours: 30
Content Summary: Getting Started with Spark, Functional Programming, Resilient Distributed Data sets, Local Virtual Box Set-up, google cloud dataproc Spark Set-up, Running Spark on a google cloud, Spark SQL, Spark Data Frame Basics, Real-time analytics with Spark Streaming, Machine learning on Big Data		
12. Brief Description of Self-learning components by students (through books/resource material etc.):		
Executing BigQuery, CloudSQL and BigTable on Google Cloud Platforms <ul style="list-style-type: none"> • https://www.qwiklabs.com/quests/50 		
Supplementary MOOC Courses <ul style="list-style-type: none"> • https://www.coursera.org/specializations/big-data • https://www.coursera.org/learn/gcp-fundamentals 		
Certification courses/programs for Skill Development <ul style="list-style-type: none"> • https://cloudxlab.com/course/67/data-engineering-with-hadoop-and-spark • https://www.cloudera.com/about/training/certification.html • https://www.cloudera.com/about/training/certification.html 		

13. Books Recommended :

Text Books:

- Boris Lublinsky, Kevin T. Smith, Alexey Yakubovich, *Professional Hadoop Solutions*, Wiley, First Edition, 2015
- Michael Minelli, Michehe Chambers, *Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Business*, Wiley CIO Series, First Edition, 2013

Reference Books:

- O'Reilly, White, *Hadoop: The Definitive Guide*, Third Edition, 2012.
- Tom Plunkett and Brian Macdonald, *Oracle Big Data Handbook*, Oracle Press, 2014.
- Furht, Borko, Villanustre, Flavio, *Big Data Technologies and Applications*, First Edition, Springer Series, 2016

Reference Websites: (nptel, swayam, coursera, edx, udemy, lms, official documentation weblink)

- <https://www.coursera.org/specializations/big-data>
- <https://www.coursera.org/learn/gcp-fundamentals>
- www.lms.ncuindia.edu/lms

Practical Content

Sr. No.	Title of the Experiment	Software based	Unit covered	Time Required(Hrs)
1.	To understand Google Cloud Platform, its services and deploying a project on Cloud	GCP Console	1	2 hrs
2.	To create a Virtual Machine on Google Cloud Platform on Console & gcloud Shell	GCP Console	1	2 hrs
3.	Create a Cloud Storage Bucket , placing image & connecting to web server.	GCP Console	1	2 hrs
4.	To create App Engine application running locally in Cloud Shell.	GCP Console	1	2 hrs

5.	To study about Big Query,loading data from cloud storage & performing query on GCP.	GCP Console	1	2 hrs
6	To study Hadoop architecture with setting up and studying various system parameters for virtual environment set up	Pseudo distributed mode on windows	2	2 hrs
7	To implement HDFS commands	Pseudo distributed mode on windows and goggle cloud	2	2 hrs
8	To implement map reduce scheme with word counting program	Pseudo distributed mode on windows	2	2 hrs
9	To implement MRJob package for map reduce programming	Windows and GCP dataproc	2	4 hrs
10	To implement multistep jobs using MRJob and design movie recommendation system	Windows and GCP dataproc	2	4 hrs
11	To implement data retrieval using Pig programming	Windows and GCP dataproc	2	2 hrs
12	Compare Pig and map reduce programming for word counting program	Windows and GCP dataproc	2	4 hrs
13	To implement web log server analytics using Pig Script	Windows and GCP dataproc	2	4 hrs
14	To implement simple data operations using Spark and compare the speed with Hadoop	Windows and GCP dataproc	3	4 hrs
15	To implement regression with Spark	Windows and GCP dataproc	3	4 hrs
16	To implement big data analytics using Spark	Windows and GCP dataproc	3	4 hrs

17	To implement Spark SQL queries on real world datasets	Windows and GCP dataproc	3	4 hrs
18	To implement Hive queries on real world datasets	Windows and GCP dataproc	4	4 hrs

Value Added Experiments

1.	KDD Cup Analysis	Windows and GCP dataproc	1,2, 3,4	8 hrs
2.	SPARK Recommender System	Windows and GCP dataproc	1,2, 3,4	8 hrs
3.	SPARK STREAMING Project	Windows and GCP dataproc	1,2, 3,4	10 hrs

Project (To be done as individual/in group): No

Experiential Learning Components

S No.	Topic	Type of Submission/Assessment Mode	Cos covered
1	Working on Cloud SQL Queries with Google Cloud Platform	Lab Sessions on Google Cloud Platform and online evaluation	CO1
2	Guest Lecture on case studies with Big Data as tool	Quiz assessment	CO1-CO5
4	Mini Projects	End Term practical assessment through presentations and viva	CO1-CO5
5	Self-Learning (Certificates earned by pursuing Supplementary MOOC courses from Coursera/NPTEL or Any Certification Program)	Completion certificate submitted by student. Component of internal marks in practical assessment	CO1-CO5

Evaluation Scheme

TYPE OF COURSE	PARTICULAR	ALLOTTED RANGE OF MARKS	PASS CRITERIA

Theory+ Practical (L-T-P/L-0-P)	Minor Test	15%	Must Secure 30% Marks Out of Combined Marks of Major Test Plus Minor Test with Overall 40% Marks in Total.
	Major Test	35%	
	Continuous Evaluation Through Class Tests/Practice/Assignments/Presentation/Quiz	10%	
	Online Quiz	5%	
	Lab Work	35%	

Mapping of PO's and CO's

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO2
CO 1	2	2	2	3	3	2	2	3	3	2	2	3	3	3
CO 2	2	2	1	3	3	2	2	2	2	2	2	2	3	1
CO 3	2	2	1	2	3	2	2	2	1	2	2	3	2	1
CO 4	2	2	2	2	2	2	2	3	2	3	3	2	2	1
CO 5	2	2	3	3	3	2	2	2	2	2	3	3	3	2

1. GENERAL INSTRUCTIONS

1.1 General discipline in the lab

- Students must turn up in time and contact concerned faculty for the experiment they are supposed to perform.
- Students will not be allowed to enter late in the lab.
- Students will not leave the class till the period is over.
- Students should come prepared for their experiment.
- Experimental results should be entered in the lab report format and certified/signed by concerned faculty/ lab Instructor.
- Students must get the connection of the hardware setup verified before switching on the power supply.
- Students should maintain silence while performing the experiments. If any necessity arises for discussion amongst them, they should discuss with a very low pitch without disturbing the adjacent groups.

- Violating the above code of conduct may attract disciplinary action.
- Damaging lab equipment or removing any component from the lab may invite penalties and strict disciplinary action.

3.1 Attendance

- Attendance in the lab class is compulsory.
- Students should not attend a different lab group/section other than the one assigned at the beginning of the session.
- On account of illness or some family problems, if a student misses his/her lab classes, he/she may be assigned a different group to make up the losses in consultation with the concerned faculty / lab instructor. Or he/she may work in the lab during spare/extra hours to complete the experiment. No attendance will be granted for such case.

3.2 Preparation and Performance

- Students should come to the lab thoroughly prepared on the experiments they are assigned to perform on that day. Brief introduction to each experiment with information about self study reference is provided on LMS.
- Students must bring the lab report during each practical class with written records of the last experiments performed complete in all respect.
- Each student is required to write a complete report of the experiment he has performed and bring to lab class for evaluation in the next working lab. Sufficient space in work book is provided for independent writing of theory, observation, calculation and conclusion.
- Students should follow the Zero tolerance policy for copying / plagiarism. Zero marks will be awarded if found copied. If caught further, it will lead to disciplinary action.
- Refer **Annexure 1** for Lab Report Format

2. RUBRICS

Marks Distribution	
Continuous Evaluation(50 Marks)	End Semester Exam (20 Marks)
<p>Each experiment shall be evaluated for 10 marks and at the end of the semester proportional marks shall be awarded out of 50.</p> <p>Following is the breakup of 10 marks for each</p> <p>4 Marks: Observation & conduct of experiment. Teacher may ask questions about experiment.</p> <p>3 Marks: For report writing</p> <p>3 Marks: For the 15 minutes quiz to be conducted in every lab.</p>	<p>End semester practical evaluation including Mini project (if any) carries 20 marks.</p>

Big Data

(CSL 311)

Lab Practical Report



Faculty name: Dr. Anuradha Dhull

Student name: Chayan Gulati

Roll No.: 18CSU054

Semester: Vth

Group: A2

Department of Computer Science and Engineering

NorthCap University, Gurugram- 122001, India

Session 2020-21

INDEX

Lab No.	Date	Experiment
1	5 Aug, 2020	Create a VM with the Cloud Console and Command Line, deploy a web server and connect it to VM
2	10 Aug, 2020	Exploring the Big Query and Cloud SQL for SQL Queries
3	10 Aug, 2020	BigQuery execution on GCP through Console and Command Line
4	18 Aug, 2020	To install Hadoop on local machine.
5	19 Aug, 2020	To write basic commands for Hadoop
6	20 Aug, 2020	MapReduce on a file to count the word appearing in it
7	24 Aug, 2020	One Mapper and One Reducer on the local machine
8	24 Aug, 2020	Two Mapper and Two Reducer on the local machine
9	24 Aug, 2020	Merge two files on the basis of a particular column on the local machine
10	26 Aug, 2020	To execute a MapReduce job using Dataproc service on GCP
11	26 Aug, 2020	To execute a MapReduce job using Dataproc service on GCP
12	16 Sep, 2020	To execute Pig Latin queries on dataset on Gcloud using Dataproc services
13	23 Sep, 2020	To execute Hive queries on item details data on Gcloud using Dataproc services.
14	7 Oct, 2020	To execute Spark program (word count) On GCP using pyspark
15	22 Oct, 2020	To count the words frequency in a file in ascending order
16	27 Oct, 2020	To find top 5 popular superhero names and their ID's
17	9 Nov, 2020	To find the lowest rated movie of all time. Consider only those movies who have more than 10 user ratings
18	27 Nov, 2020	Spark Machine Learning implementation of Algorithms

Lab – 1

Student Name and Roll Number: Chayan 18csu054

Semester /Section: 5/A

Link to Code: <https://github.com/chayangulati321/Big-Data>

Date: 05/08/20

Faculty Signature:

Remarks:

Creating a Virtual Machine on GCP and deploying web server “NGINX”

AIM : Create a VM with the Cloud Console and Command Line , deploy a web server and connect it to VM

VM creation using Cloud Shell

Step – 1 : Activate Google Cloud Shell. Use the following commands

- List the active account name \$gcloud auth list
- List the project ID \$gcloud config list project

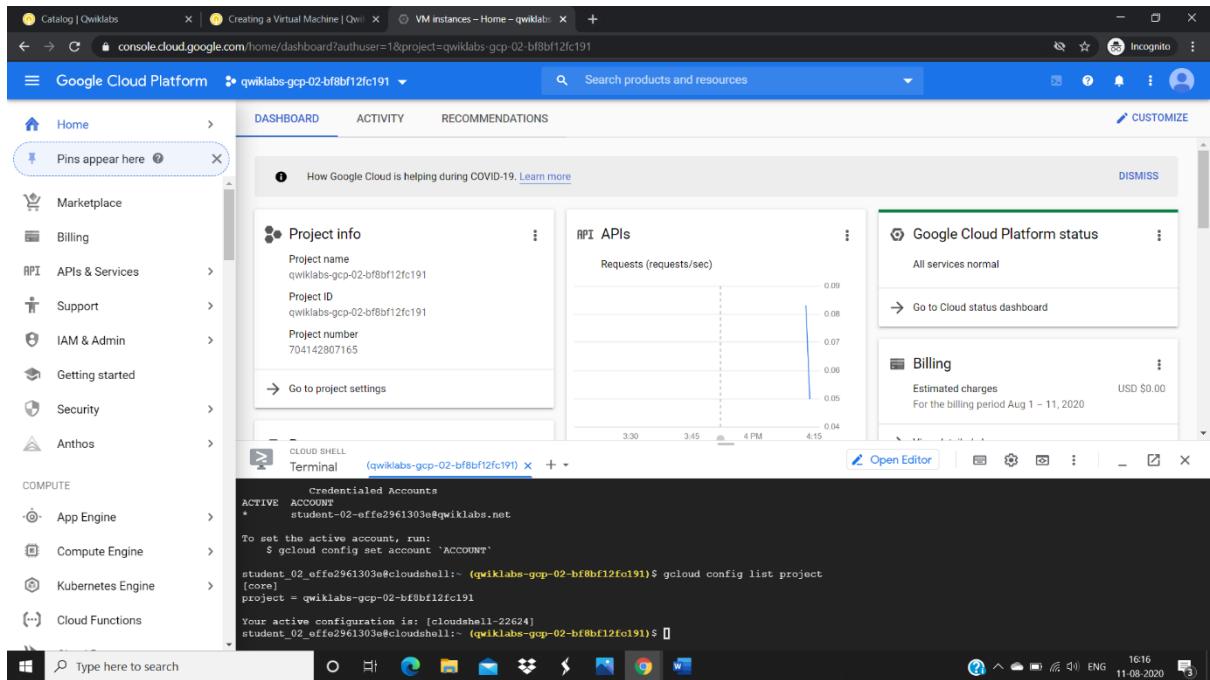


Fig1

Step -2 : Create a new instance from the Cloud Console. In the Cloud Console, on the top left of the screen, select **Navigation menu > Compute Engine > VM Instances**:

To create a new instance, click **Create**.

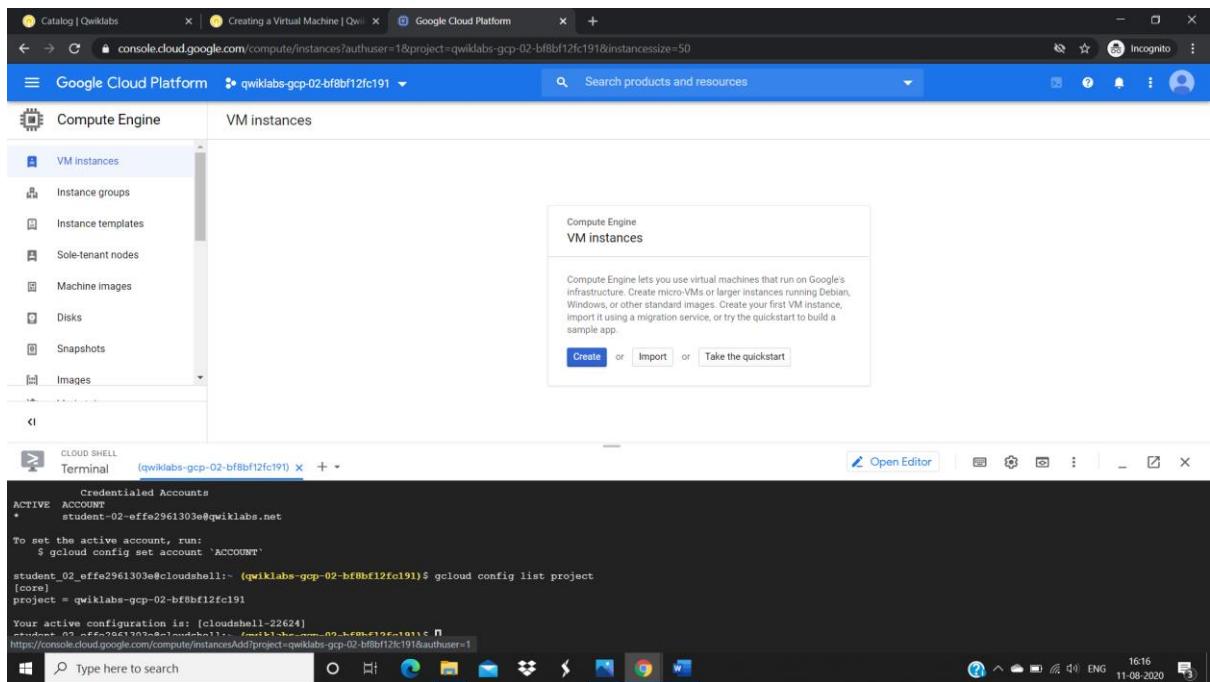


Fig2

Step – 3 : Use the following for this lab:

Name	gcelab	Name for the VM instance
Region	us-central1 (Iowa)	Learn more about regions in Regions & Zones documentation .
Zone	us-central1-c Note: remember the zone that you selected, you'll need it later.	Learn more about zones in Regions & Zones documentation .
Machine Type	2 vCPUs This is a (n1-standard-2), 2-CPU, 7.5GB RAM instance. There are a number of machine types, ranging from micro instance types to 32-core/208GB RAM instance types. Learn more in the Machine Types documentation .	Note: A new project has a default resource quota , which may limit the number of CPU cores. You can request more when you work on projects outside of this lab.
Boot Disk	New 10 GB standard persistent disk OS Image: Debian GNU/Linux 9 (stretch)	There are a number of images to choose from, including: Debian, Ubuntu, CoreOS as well as premium images such as Red Hat Enterprise Linux and Windows Server. See Operating System documentation for more detail.
Firewall	Check Allow HTTP traffic Check this option so to access a webserver that you'll install later.	Note: This will automatically create firewall rule to allow HTTP traffic on port 80.

Fig3

Step – 4 : To SSH into the virtual machine

click on **SSH** on the right hand side. This launches a SSH client directly from your browser

<input type="checkbox"/> Name ^	Zone	Recommendation	Internal IP	External IP	Connect
<input checked="" type="checkbox"/> gcelab	us-central1-c		10.128.0.2 (nic0)	35.194.40.76	SSH 

Fig4

Step – 5: Install a NGINX web server using following commands

- \$sudo su –
- \$apt-get update

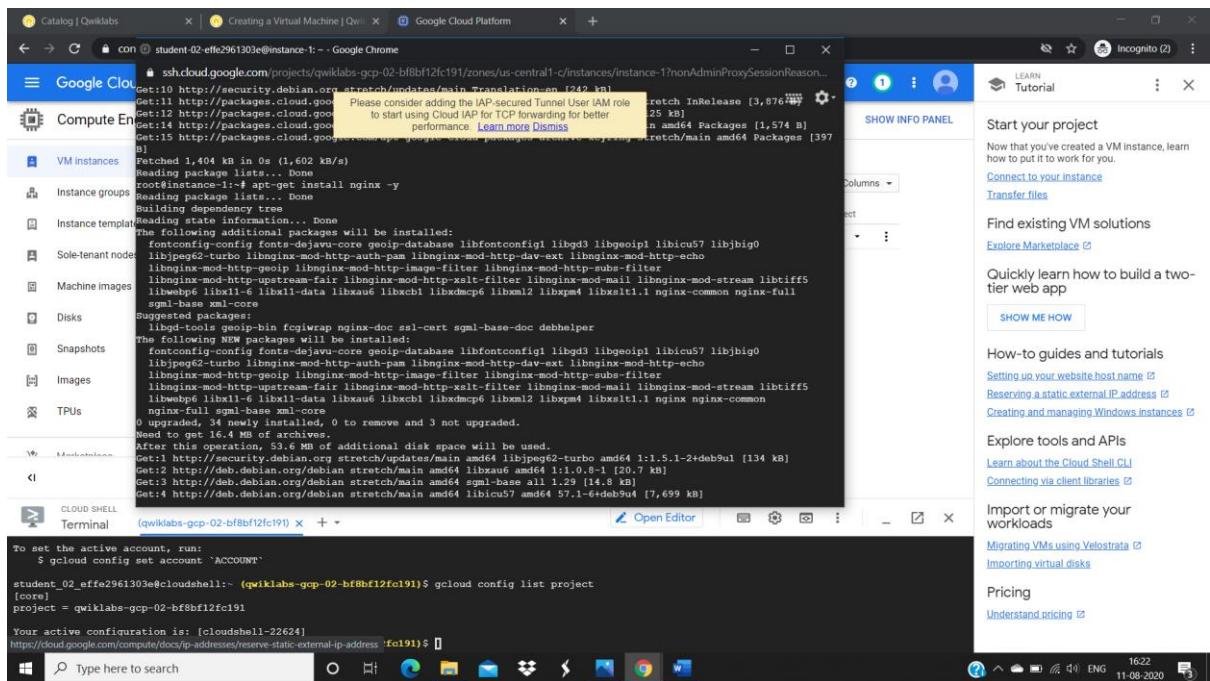


Fig5

Step – 6: click the External IP link of the virtual machine instance

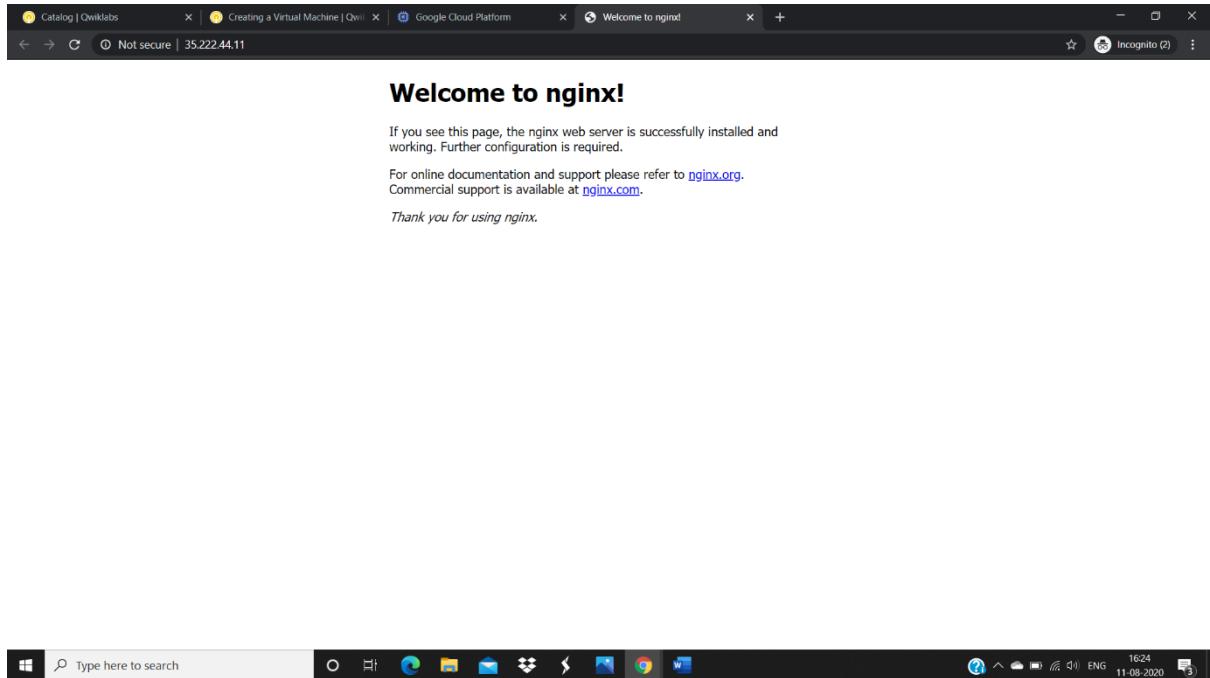


Fig6

VM creation using gcloud

Step – 7: In the Cloud Shell, create a new virtual machine instance from the command line using gcloud:

- \$gcloud compute instances create gcelab2 --machine-type n1-standart-2 --zone us-central1-c

Select Navigation menu > Compute Engine > VM instances

Create a new instance with gcloud

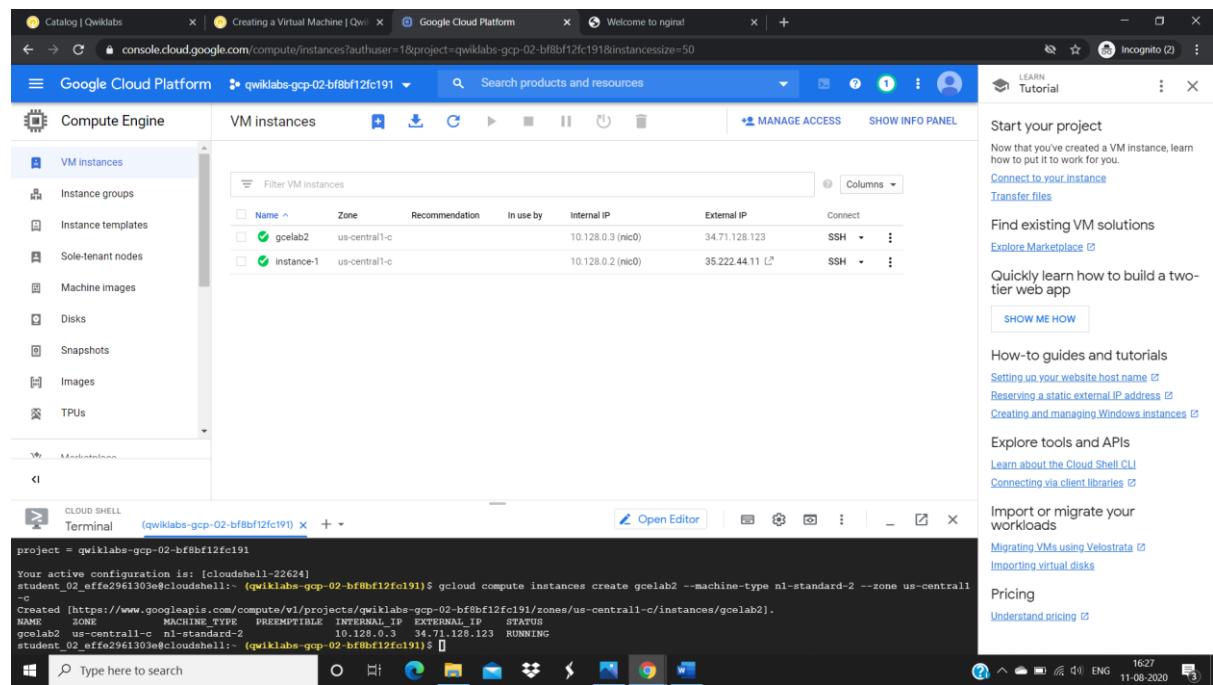


Fig7

Step – 8: SSH into instance using gcloud. Make sure add , or omit the –zone flag if it set the option globally:

- gcloud compute ssh gcelab2 –zone us-central1-c
- Now you'll type Y to continue
- exit

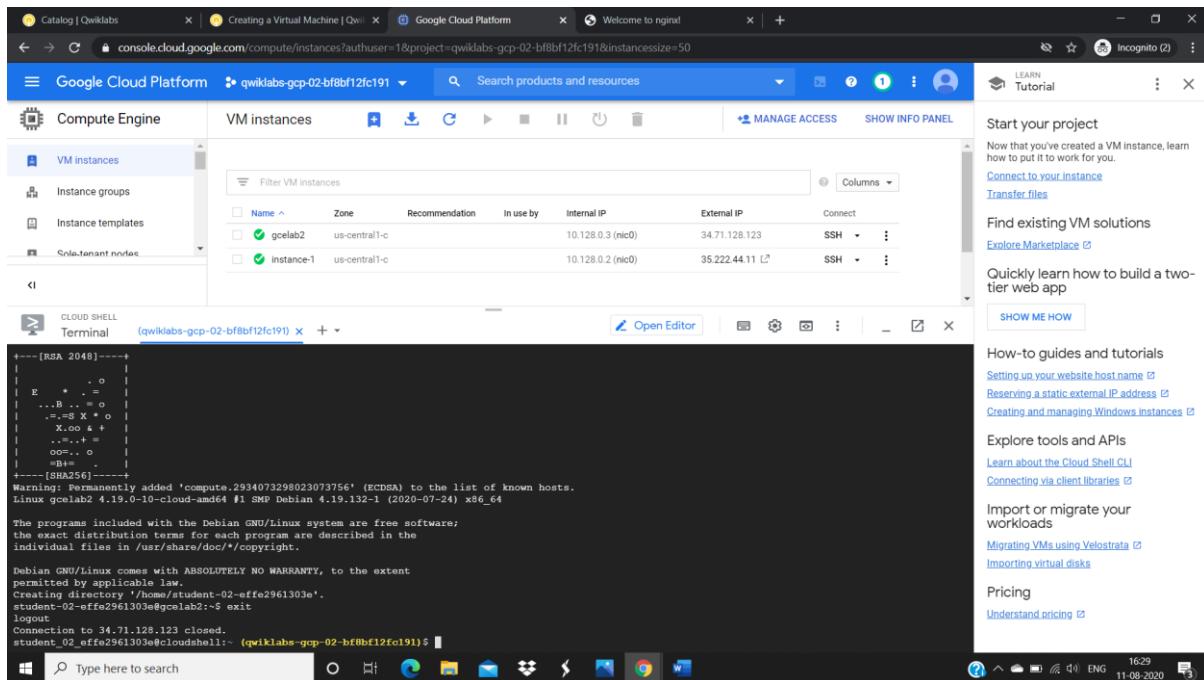


Fig8

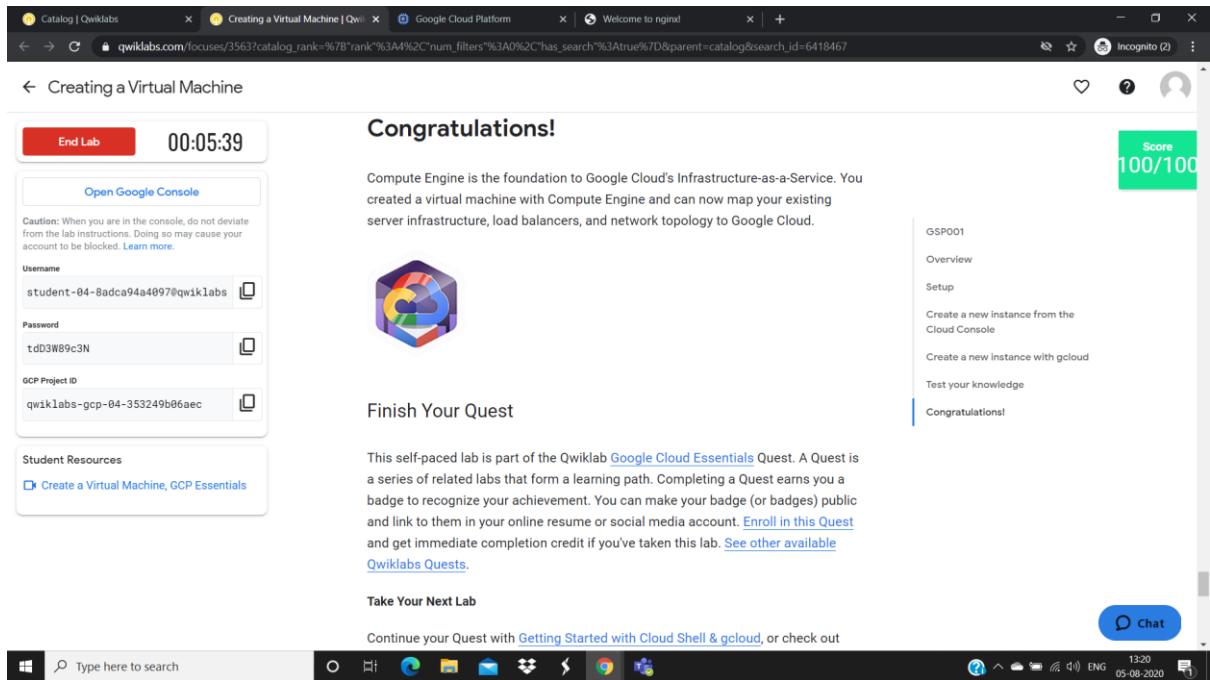


Fig9

Lab – 2

Student Name and Roll Number: Chayan 18csu054

Semester /Section: 5/A

Link to Code: <https://github.com/chayangulati321/Big-Data>

Date: 10/08/20

Faculty Signature:

Remarks:

Introduction to SQL for Big Query and Cloud SQL

AIM : Exploring the Big Query and Cloud SQL for SQL Queries

Step – 1 : Open BigQuery Console. In the Google Cloud Console, select Navigation menu > BigQuery:

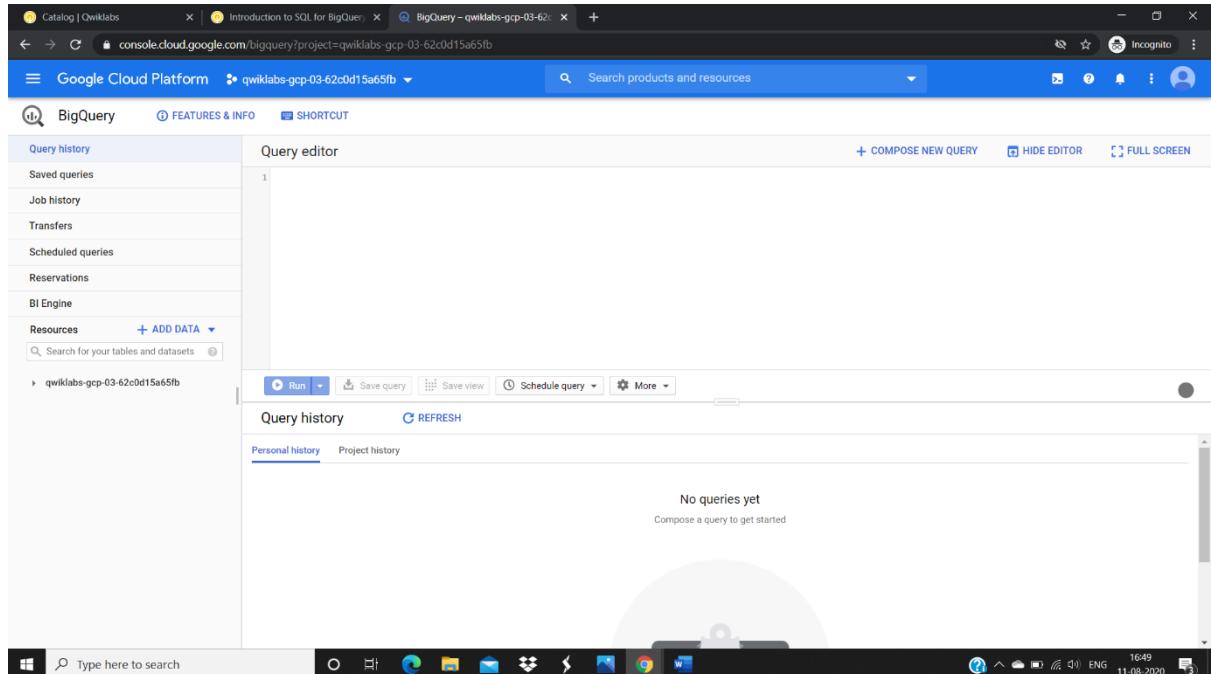


Fig1

Step -2 : Uploading queryable data. Click on the **+ ADD DATA** link then select **Explore public datasets**:

- In the search bar, enter "london", then select the **London Bicycle Hires** tile, then **View Dataset**.
- A new tab will open, and you will now have a new project called **bigrquery-public-data** added to the Resources panel
- Click on **bigrquery-public-data > london_bicycles > cycle_hire**

The screenshot shows the Google Cloud Platform BigQuery interface. The top navigation bar includes tabs for Catalog, Introduction to SQL for BigQuery, BigQuery - qwiklabs-gcp-03-62c0d15a65fb, and BigQuery - qwiklabs-gcp-03-62c0d15a65fb. The main window has a sidebar with links like Query history, Saved queries, Job history, Transfers, Scheduled queries, Reservations, BI Engine, and Resources. The Resources section is expanded, showing a search bar and a '+ ADD DATA' button. Below this, a tree view shows datasets: 'labeled_patents', 'libraries_io', 'london_bicycles', and 'ml_datasets'. Under 'london_bicycles', the 'cycle_hire' table is selected. The 'Preview' tab is active, displaying the schema and a table with four rows of data. The schema columns are: Row, rental_id, duration, bike_id, end_date, end_station_id, end_station_name, start_date, start_station_id, and start_station_name. The data rows are:

Row	rental_id	duration	bike_id	end_date	end_station_id	end_station_name	start_date	start_station_id	start_station_name
1	47469109	3180	7054	2015-09-03 12:45:00 UTC	111	Park Lane , Hyde Park	2015-09-03 11:52:00 UTC	300	Serpentine Car Park, Hyde Park
2	46915469	7380	3792	2015-08-16 11:59:00 UTC	407	Speakers' Corner 1, Hyde Park	2015-08-16 09:56:00 UTC	407	Speakers' Corner 1, Hyde Park
3	65899423	2040	3038	2017-06-09 18:30:00 UTC	165	Orsett Terrace, Bayswater	2017-06-09 17:56:00 UTC	579	Queen Street 2, Bank
4	64280726	2280	10868	2017-04-22 10:14:00 UTC	553	Regent's Row , Haggerston	2017-04-22 09:36:00 UTC	519	Teviot Street, Poplar

Below the table, there are buttons for Run, Save query, Save view, Schedule query, More, QUERY TABLE, COPY TABLE, DELETE TABLE, and EXPORT. The bottom of the screen shows a taskbar with various icons and a system tray.

Fig2

Step – 3 : Running SELECT, FROM, and WHERE in BigQuery. Commands as follows :

- \$SELECT end_station_name FROM 'bigquery-public-data.london.bicycles.cycle.hire';

Click Run

Click **COMPOSE NEW QUERY** to clear the Query editor, then run the following query that utilizes the WHERE keyword

- \$SELECT * FROM 'bigquery-public-data.london.bicycles.cycle.hire' WHERE duration>=1200;

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, there is a sidebar with various datasets and tables listed under categories like IRS, London, and Medicare. The main area is divided into two tabs: 'Query editor' and 'Query results'. In the 'Query editor' tab, a single line of SQL code is present: '1 SELECT end_station_name FROM "bigquery-public-data.london.bicycles.cycle_hire";'. Below this, the 'Query results' tab displays the output of the query. The results table has one column, 'end_station_name', with four rows: '1 Storey's Gate, Westminster', '2 Russell Square Station, Bloomsbury', '3 Grosvenor Square, Mayfair', and '4 Austin Road, Battersea Park'. At the bottom of the results table, there are navigation buttons for 'First page', 'Last page', and 'Rows per page: 100'. The status bar at the bottom right indicates the date '11-08-2020' and time '16:51'.

Fig3

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, there's a sidebar with options like Query history, Saved queries, Job history, Transfers, Scheduled queries, Reservations, BI Engine, and Resources. Below the sidebar is a search bar for tables and datasets. The main area is titled 'Query editor' and contains a code editor with the following SQL query:

```
1 SELECT * FROM `bigquery-public-data.london_bicycles.cycle_hire` WHERE duration>=1200;
```

Below the code editor are buttons for Run, Save query, Save view, Schedule query, and More. The results section shows a table with the following data:

Row	rental_id	duration	bike_id	end_date	end_station_id	end_station_name	start_date	start_station_id	start_station_name
1	47469109	3180	7054	2015-09-03 12:45:00 UTC	111	Park Lane, Hyde Park	2015-09-03 11:52:00 UTC	300	Serpentine Car Park, Hyde Park
2	46915469	7380	3792	2015-08-16 11:59:00 UTC	407	Speakers' Corner 1, Hyde Park	2015-08-16 09:56:00 UTC	407	Speakers' Corner 1, Hyde Park
3	65899423	2040	3038	2017-06-09 18:30:00 UTC	165	Orsett Terrace, Bayswater	2017-06-09 17:56:00 UTC	579	Queen Street 2, Bank
4	64280726	2280	10868	2017-04-22 10:14:00 UTC	553	Regent's Row, Haggerston	2017-04-22 09:36:00 UTC	519	Teviot Street, Poplar

At the bottom of the results page, there are buttons for Rows per page (set to 100), First page, Last page, and navigation arrows.

Fig4

Step – 4 : More SQL Keywords: GROUP BY, COUNT, AS, and ORDER BY

- `SELECT start_station_name FROM 'bigquery-public-data.london.bicycles.cycle.hire'`
GROUP BY start station name;

Click Run

Click **COMPOSE NEW QUERY** to clear the Query editor

- `SELECT start_station_name, COUNT(*) FROM 'bigquery-public-data.london.bicycles.cycle.hire'` GROUP BY start station name;

Click Run

Click **COMPOSE NEW QUERY** to clear the Query editor

- `SELECT start_station_name, COUNT(*) AS num_start FROM 'bigquery-public-data.london.bicycles.cycle.hire'` GROUP BY start station name;

Click Run

Click **COMPOSE NEW QUERY** to clear the Query editor

- `SELECT start_station_name, COUNT(*) AS num FROM 'bigquery-public-data.london.bicycles.cycle.hire'` GROUP BY start station name ORDER BY num DESC;

Click Run

Click **COMPOSE NEW QUERY** to clear the Query editor

- `SELECT end_station_name, COUNT(*) AS num FROM 'bigquery-public-data.london.bicycles.cycle.hire'` GROUP BY end station name ORDER BY num DESC;

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, there is a sidebar with various options like Query history, Saved queries, Job history, Transfers, Scheduled queries, Reservations, BI Engine, and Resources. Below the sidebar, there is a search bar and a list of datasets. The main area is the Query editor, which contains a code editor with the following SQL query:

```
1 SELECT start_station_name FROM `bigquery-public-data.london_bicycles.cycle_hire` GROUP BY start_station_name;
```

Below the code editor is the Query results section. It shows the following data:

Row	start_station_name
1	Serpentine Car Park, Hyde Park
2	Speakers' Corner 1, Hyde Park
3	Queen Street 2, Bank
4	Teviot Street, Poplar

At the bottom of the interface, there are buttons for Run, Save query, Save view, Schedule query, More, and a note indicating the query will process 676.2 MB when run.

Fig5

This screenshot is similar to Fig5, showing the Google Cloud Platform BigQuery interface. The sidebar and search bar are identical. The Query editor contains the following SQL query:

```
1 SELECT start_station_name, COUNT(*) FROM `bigquery-public-data.london_bicycles.cycle_hire` GROUP BY start_station_name;
```

The Query results section shows the following data:

Row	start_station_name	f0_
1	Serpentine Car Park, Hyde Park	74903
2	Speakers' Corner 1, Hyde Park	85835
3	Queen Street 2, Bank	51658
4	Teviot Street, Poplar	3885

At the bottom, there is a note indicating the query will process 676.2 MB when run.

Fig6

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, there is a sidebar with various options like Query history, Saved queries, Job history, Transfers, Scheduled queries, Reservations, BI Engine, and Resources. Below the sidebar, a search bar says "Search for your tables and datasets". The main area is divided into two sections: "Query editor" and "Query results".

In the "Query editor" section, the following SQL query is displayed:

```
1 SELECT start_station_name, COUNT(*) AS num_starts FROM `bigquery-public-data.london_bicycles.cycle_hire` GROUP BY start_station_name;
```

Below the query editor, there are buttons for Run, Save query, Save view, Schedule query, and More. A note indicates that the query will process 676.2 MB when run.

The "Query results" section shows the output of the query. It includes tabs for Job information, Results (which is selected), JSON, and Execution details. The results table has columns "Row", "start_station_name", and "num_starts". The data is as follows:

Row	start_station_name	num_starts
1	Serpentine Car Park, Hyde Park	74903
2	Speakers' Corner 1, Hyde Park	85835
3	Queen Street 2, Bank	51658
4	Teviot Street, Poplar	3885

At the bottom of the results page, there are buttons for Rows per page (set to 100), First page, Last page, and navigation arrows.

Fig7

This screenshot is identical to Fig7, showing the same BigQuery interface and query results. The only difference is the order of the results. The query used is:

```
1 SELECT start_station_name, COUNT(*) AS num FROM `bigquery-public-data.london_bicycles.cycle_hire` GROUP BY start_station_name ORDER BY num DESC;
```

The results table now shows the stations with the highest counts at the top:

Row	start_station_name	num
1	Belgrave Street, King's Cross	234458
2	Hyde Park Corner, Hyde Park	215629
3	Waterloo Station 3, Waterloo	201630
4	Black Lion Gate, Kensington Gardens	161952

Fig8

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, there is a sidebar with various navigation links: Catalog | Qwiklabs, Introduction to SQL for BigQuery, BigQuery - qwiklabs-gcp-03-62c0d15a65fb, and BigQuery - qwiklabs-gcp-03-62c0d15a65fb. The main area is titled "Query editor" and contains a query:

```
1 SELECT end_station_name, COUNT(*) AS num FROM `bigquery-public-data.london_bicycles.cycle_hire` GROUP BY end_station_name ORDER BY num DESC;
```

Below the query editor is the "Query results" section. It shows the following data:

Row	end_station_name	num
1	Belgrave Street, King's Cross	231802
2	Hyde Park Corner, Hyde Park	215038
3	Waterloo Station 3, Waterloo	193200
4	Albert Gate, Hyde Park	157943
...

At the bottom of the results page, there are buttons for "Run", "Save query", "Save view", "Schedule query", and "More". A note says "This query will process 676 MB when run." Below the table, there are buttons for "Job information", "Results" (which is selected), "JSON", and "Execution details". At the very bottom, there are buttons for "SAVE RESULTS" and "EXPLORE DATA".

Fig9

Step – 5 : Working with Cloud SQL, Select **Navigation menu > Storage > Browser**, and then click **Create bucket**

Enter a unique name for your bucket, keep all other settings, and hit **Create**

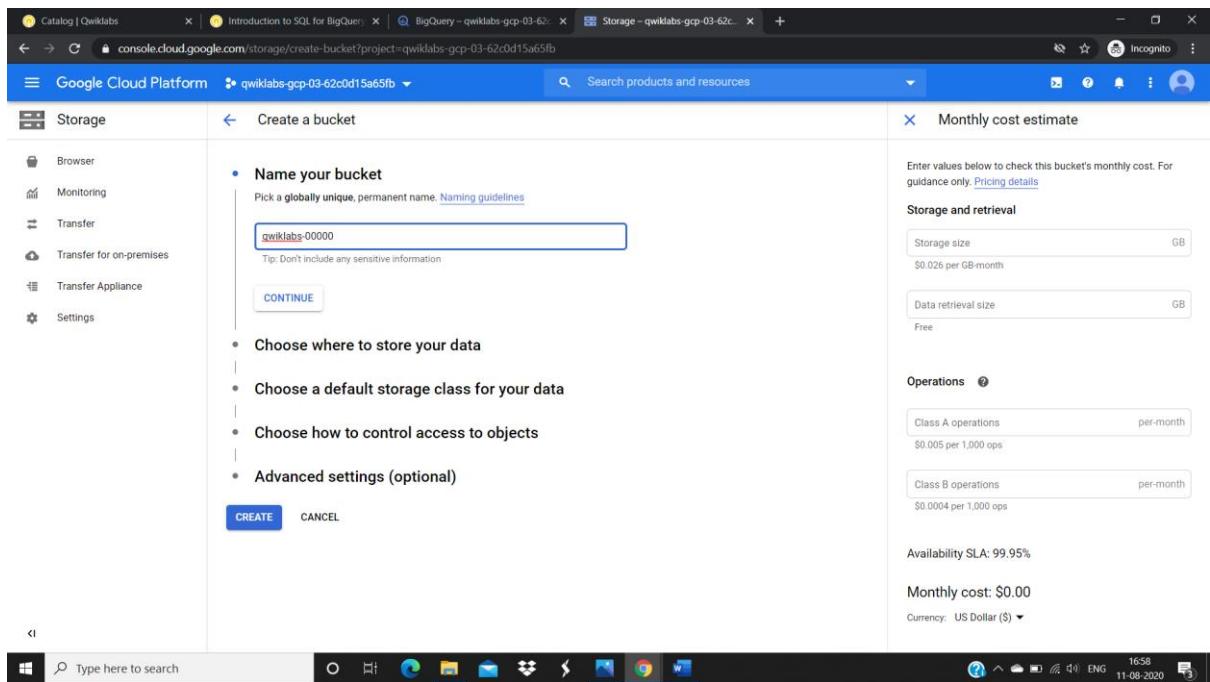


Fig10

Step – 6 : Click Upload files and select the CSV that contains start_station_name data. Then click Open. Repeat this for the end_station_name data.

Rename the file start_station_data to start_station_data.csv.

Rename the file end_station_data to end_station_data.csv.

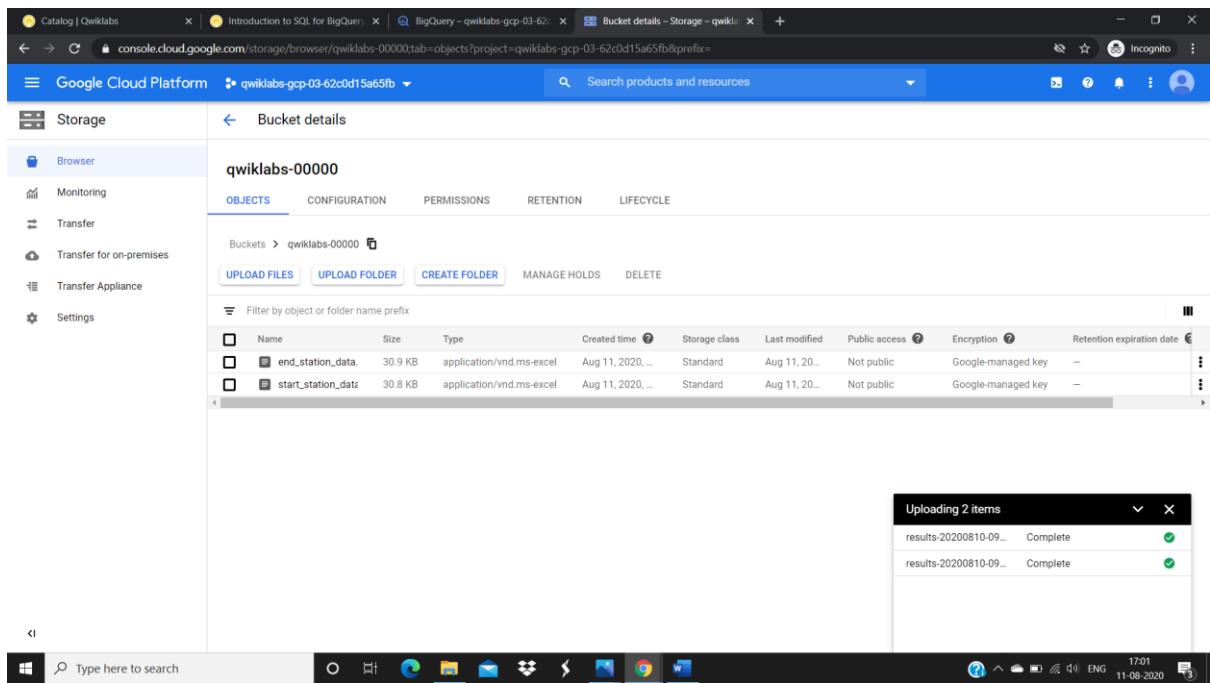


Fig11

Step -7 : Create a Cloud SQL instance. In the console, select **Navigation menu > SQL**. Click **Create Instance**.

Select **MySQL**. Now enter in a name for your instance (like "qwiklabs-demo") and enter in a secure password in the **Root password** field (remember it!), then click **Create**

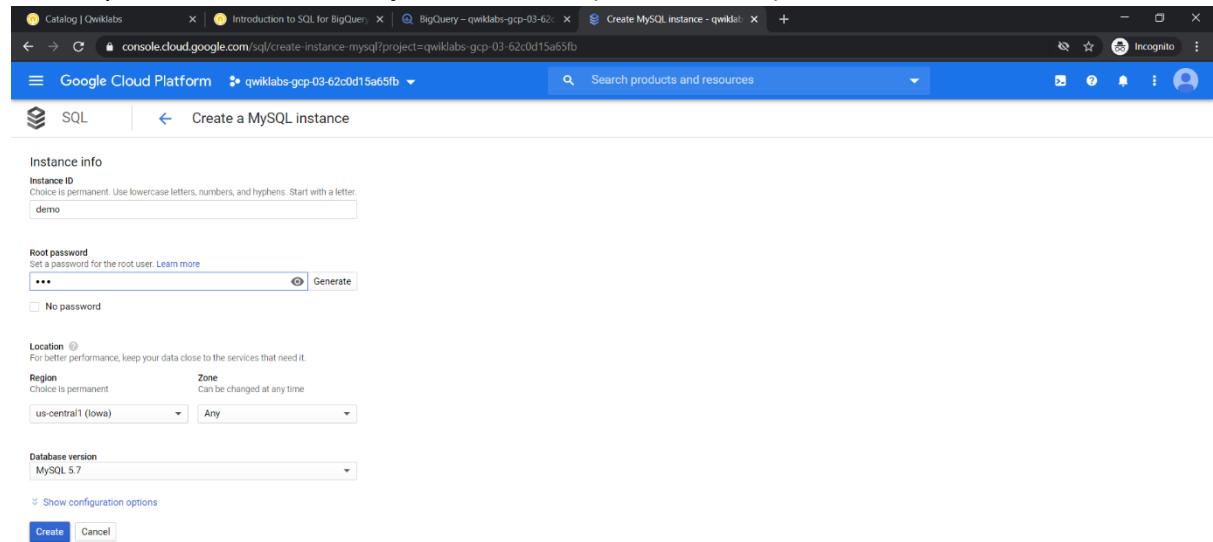


Fig12

Step – 8 : Click on the Cloud SQL instance

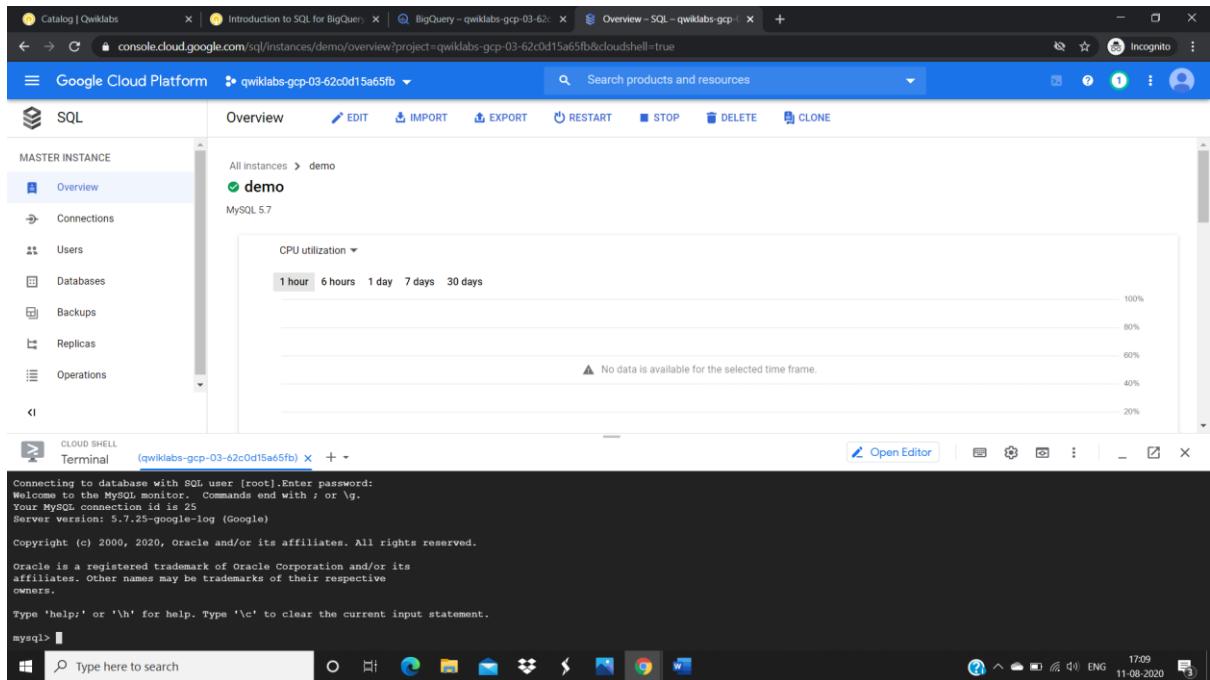


Fig13

Step – 9 : New Queries in Cloud SQL , Activate Cloud Shell

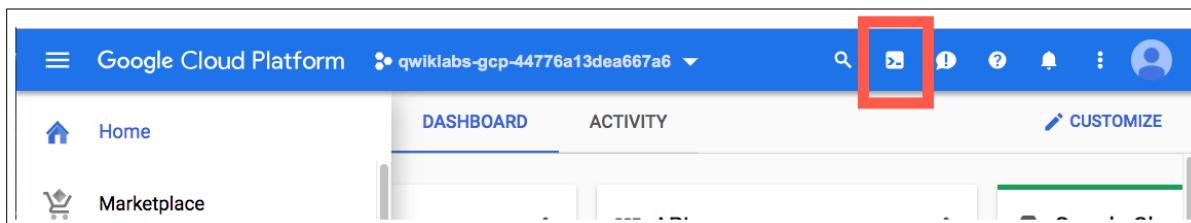


Fig14

Step – 10 : Use the following commands

- List the active account name \$gcloud auth list
- List the project ID \$gcloud config list project

Run the following command in Cloud Shell to connect to your SQL instance, replacing demo if you used a different name for your instance

- \$gcloud sql connect demo –user=root

enter the root password

Run the following command at the MySQL server prompt to create a database called bike:

- CREATE DATABASE bike;

Make a table inside of the bike database by running the following command:

- USE bike;
- CREATE TABLE london1 (start_startion_name VARCHAR(255), num INT);
- USE bike;
- CREATE TABLE london2 (end_startion_name VARCHAR(255), num INT);

confirm that your empty tables were created:

- SELECT * FROM london1;
- SELECT * FROM london2;

The screenshot shows the Google Cloud Platform SQL instance page for a project named 'demo'. In the terminal window, the following MySQL commands are run:

```

mysql> CREATE DATABASE bike;
Query OK, 1 row affected (0.19 sec)

mysql> USE bike;
Database changed
mysql> CREATE TABLE london1 (start_station_name VARCHAR(255), num INT);
Query OK, 0 rows affected (0.21 sec)

mysql> SELECT * FROM london1;
Empty set (0.19 sec)

mysql> CREATE TABLE london2 (end_station_name VARCHAR(255), num INT);
Query OK, 0 rows affected (0.21 sec)

mysql> SELECT * FROM london2;
Empty set (0.19 sec)

```

Fig15

Step – 11 : In your Cloud SQL instance page, click **IMPORT**.

In the Cloud Storage file field, click **Browse**, and then click the arrow opposite your bucket name, and then click start_station_data.csv. Click **Select**.

1. Select the bike database and type in "london1" as your table.
2. Click **Import**:

The screenshot shows the Google Cloud Platform SQL instance page for a project named 'demo Import - qwiklabs-gcp-03'. In the terminal window, the following MySQL commands are run to verify the import:

```

mysql> USE bike;
Database changed
mysql> CREATE TABLE london2 (end_station_name VARCHAR(255), num INT);
Query OK, 0 rows affected (0.21 sec)

mysql> SELECT * FROM london1;
Empty set (0.19 sec)

mysql> SELECT * FROM london2;
Empty set (0.19 sec)

```

Fig16

Do the same for the other CSV file

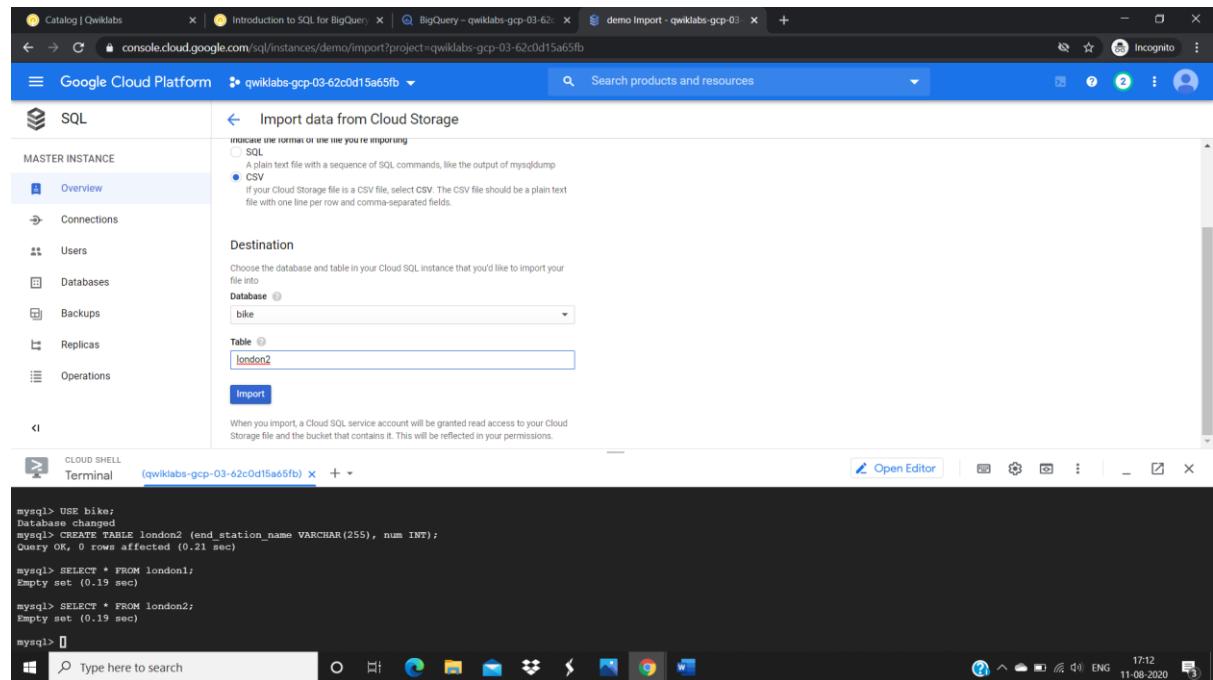


Fig17

Step – 12 : both CSV files uploaded to tables in the bike database.

Return to your Cloud Shell session and run the following command at the MySQL server prompt to inspect the contents of london1:

- \$SELECT * FROM london1;

Run the following command to make sure that london2 has been populated:

- \$SELECT * FROM london2;

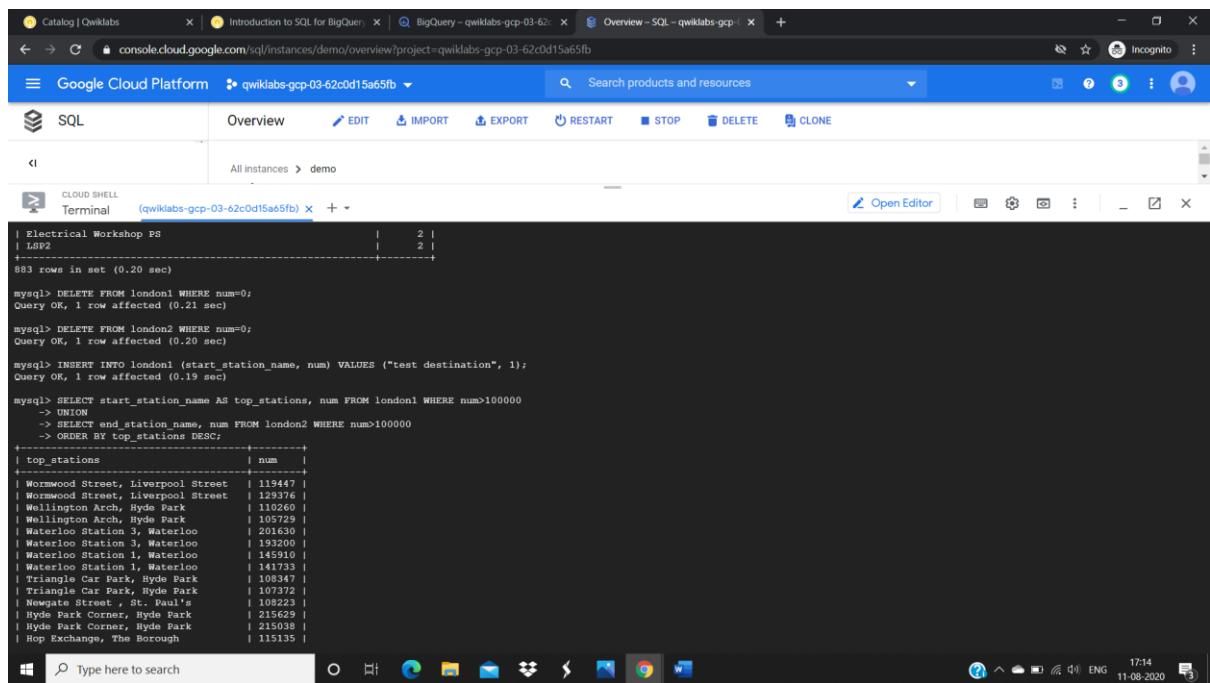
The screenshot shows the Google Cloud Platform interface. In the top navigation bar, there are tabs for Catalog | Qwiklabs, Introduction to SQL for BigQuery, BigQuery - qwiklabs-gcp-03-62c0d15a65fb, Overview - SQL - qwiklabs-gcp, and Incognito. Below the navigation bar, the main title is "Google Cloud Platform" followed by "qwiklabs-gcp-03-62c0d15a65fb". A search bar says "Search products and resources". On the left, there's a sidebar with "SQL" selected, showing "MASTER INSTANCE" with "Overview" (green status), "Connections", "Users", "Databases", and "Deadlock". In the center, there's an "Overview" section for the "demo" instance, showing CPU utilization (1 hour, 6 hours, 1 day, 7 days, 30 days) and a graph from 100% down to 80%. Below the overview is a "CLOUD SHELL" terminal window titled "(qwiklabs-gcp-03-62c0d15a65fb)". The terminal shows the following MySQL query results:

```
mysql> SELECT * FROM london1;
+-----+-----+
| start_station_name | num |
+-----+-----+
| Belgrave Street , King's Cross | 0 |
| Hyde Park Corner, Hyde Park | 234458 |
| Waterloo Station, Waterloo | 215629 |
| Black Lion Gate, Kensington Gardens | 201630 |
| Albert Gate, Hyde Park | 145910 |
| Waterloo Station 1, Waterloo | 119447 |
| Wormwood Street, Liverpool Street | 115135 |
| Hop Exchange, The Borough | 110260 |
| Wellington Arch, Hyde Park | 107777 |
| Triangular Buildings, Hyde Park | 105810 |
| Piccadilly Circus, Liverpool Street | 103114 |
| Brushfield Street, Liverpool Street | 100005 |
| Bethnal Green Road, Shoreditch | 99706 |
| Regent's Row , Haggerston | 99706 |
```

Fig18

Step - 13 : Run the following commands in your MySQL session to delete the first row of the london1 and london2:

- \$DELETE FROM london1 WHERE num=0;
- \$DELETE FROM london2 WHERE num=0;



The screenshot shows a Google Cloud Platform interface for a MySQL database named 'demo'. The terminal window displays the following MySQL session:

```
| Electrical Workshop PS |      2 |
| LSP2                   |      2 |
+-----+-----+
883 rows in set (0.20 sec)

mysql> DELETE FROM london1 WHERE num=0;
Query OK, 1 row affected (0.21 sec)

mysql> DELETE FROM london2 WHERE num=0;
Query OK, 1 row affected (0.20 sec)

mysql> INSERT INTO london1 (start_station_name, num) VALUES ("test destination", 1);
Query OK, 1 row affected (0.19 sec)

mysql> SELECT start_station_name AS top_stations, num FROM london1 WHERE num>100000
-> UNION
-> SELECT end_station_name, num FROM london2 WHERE num>100000
-> ORDER BY top_stations DESC
+-----+-----+
| top_stations           | num   |
+-----+-----+
| Wormwood Street, Liverpool Street | 119447 |
| Wormwood Street, Liverpool Street | 119396 |
| Wellington Arch, Hyde Park     | 110260 |
| Wellington Arch, Hyde Park     | 105729 |
| Waterloo Station 3, Waterloo    | 201630 |
| Waterloo Station 3, Waterloo    | 193200 |
| Waterloo Station 1, Waterloo    | 145910 |
| Waterloo Station 1, Waterloo    | 141733 |
| Triangle Car Park, Hyde Park    | 130347 |
| Triangle Car Park, Hyde Park    | 107122 |
| Newgate Street - St. Paul's     | 108223 |
| Hyde Park Corner, Hyde Park     | 215629 |
| Hyde Park Corner, Hyde Park     | 215038 |
| Hop Exchange, The Borough       | 115135 |
+-----+-----+
```

Fig19

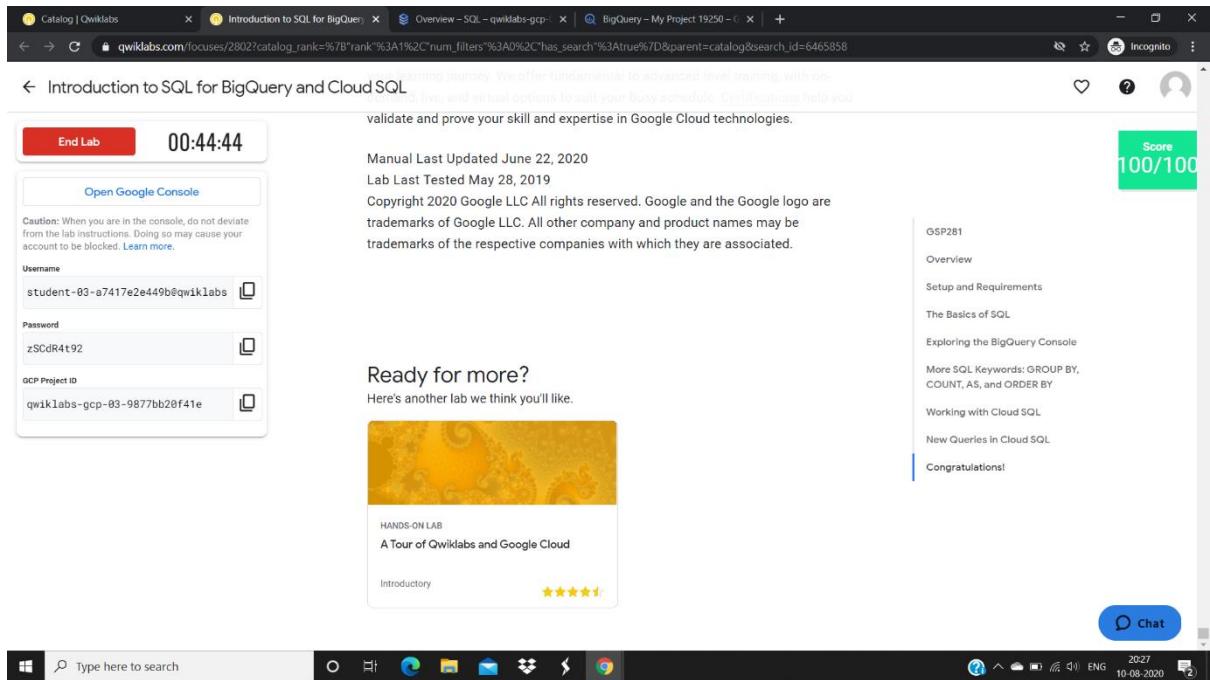


Fig20

Lab – 3

Student Name and Roll Number: Chayan 18csu054

Semester /Section: 5/A

Link to Code: <https://github.com/chayangulati321/Big-Data>

Date: 10/08/20

Faculty Signature:

Remarks:

AIM : BigQuery execution on GCP through Console and Command Line

BigQuery:- Console

Step – 1 : Activate Google Cloud Shell. By Clicking on the icon in Fig1.

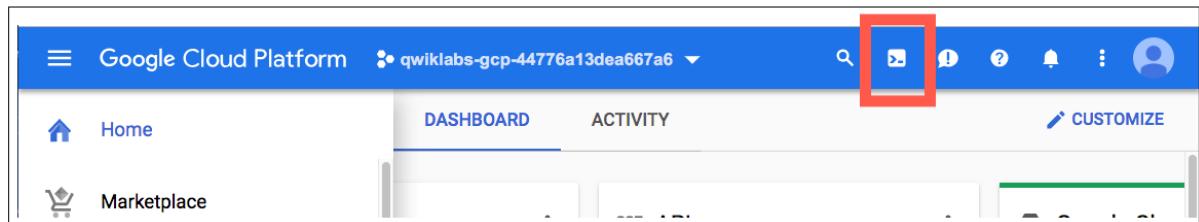


Fig1

Then Click Continue

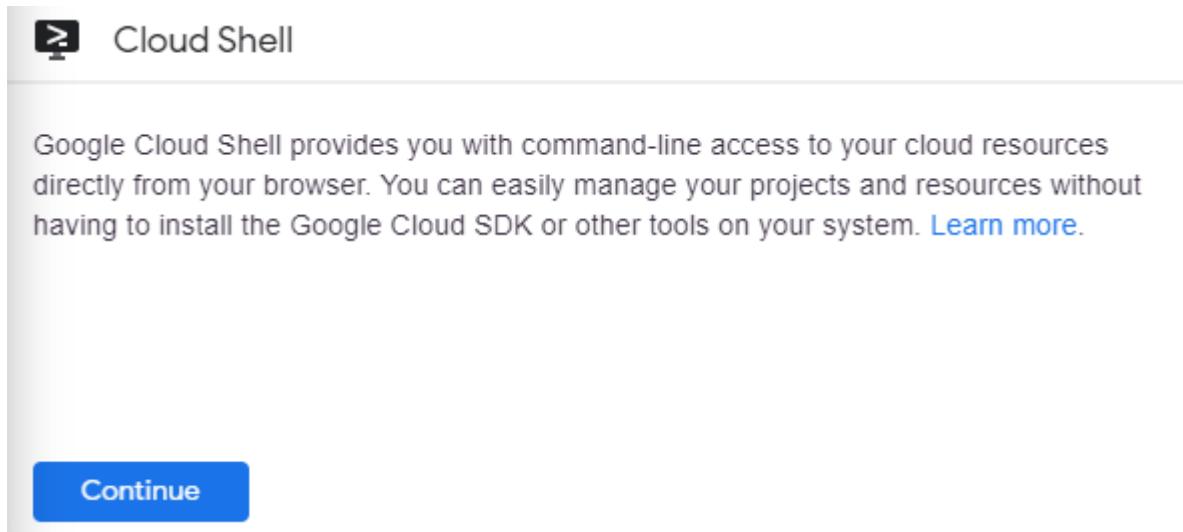


Fig2

Use the following commands

- List the active account name `$gcloud auth list`
- List the project ID `$gcloud config list project`

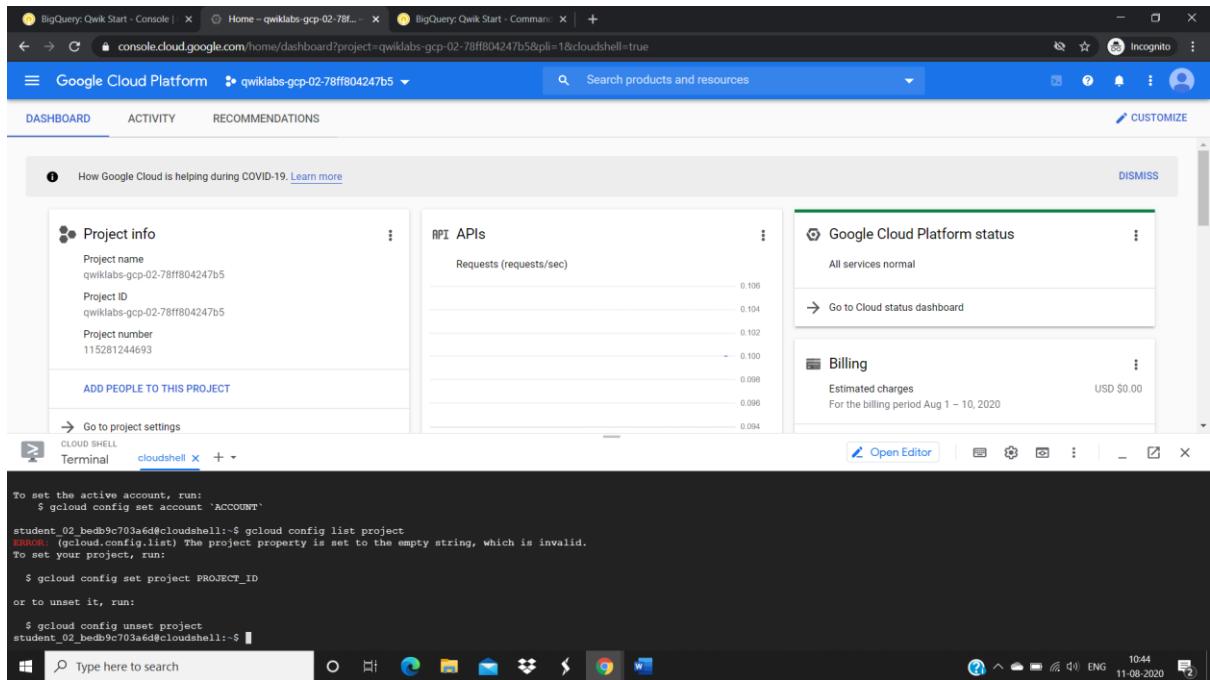


Fig3

Step -2 : Open **BigQuery** in the Cloud Console

- In the Google Cloud Console, select **Navigation menu > BigQuery**:



[Home](#)

BIG DATA

[BigQuery](#)

- [Pub/Sub](#)
- [Dataproc](#)
- [Dataflow](#)
- [ML Engine](#)

Fig4

Query into the BigQuery Query editor. Using following commands:

- \$ #standardSQL
- SELECT weight_pounds, state, year, gestation_weeks
- FROM
- 'Bigquery-public-data.samples.natality'
- ORDER BY weight_pounds DESC LIMIT 10;

```

1 #standardSQL
2 SELECT
3   weight_pounds, state, year, gestation_weeks
4 FROM
5   `bigquery-public-data.samples.natality`
6 ORDER BY weight_pounds DESC LIMIT 10;

```

Row	weight_pounds	state	year	gestation_weeks
1	18.0007436923	TX	1969	null
2	18.0007436923	null	2007	38
3	18.0007436923	OR	1972	40

```

ERROR: (gcloud.config.list) The project property is set to the empty string, which is invalid.
To set your project, run:
$ gcloud config set project PROJECT_ID
or to unset it, run:
$ gcloud config unset project
student_02_bedb9c703a6@cloudshell:~$ 

```

Fig5

Click Run

Query results [!\[\]\(8e8367aa9ae7269cd142a342b8d4e4b0_img.jpg\) SAVE RESULTS ▾](#)

Query complete (2.626 sec elapsed, 3.49 GB processed)

Job information [Results](#) [JSON](#) [Execution details](#)

Row	weight_pounds	state	year	gestation_weeks
1	18.0007436923	KY	2004	47
2	18.0007436923	OR	1972	40
3	18.0007436923	null	2007	39
4	18.0007436923	null	2008	null
5	18.0007436923	TX	1969	null
6	18.0007436923	null	2005	40
7	18.0007436923	null	2007	45
8	18.0007436923	null	2005	null
9	18.0007436923	GA	1979	34
10	18.0007436923	null	2007	38

Fig6

Step -3 : Load custom data into a table and **Create Dataset**.

In the left pane, click project name in the **Resources** navigation, then click **Create Dataset**. Widen browser window to see the **Create Dataset** option

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, there's a sidebar with 'Resources' selected. Below it, the 'Resources' section shows a dropdown menu with 'qwiklabs-gcp-gcpd-5eb9880b7ce7'. A red box highlights this dropdown. To the right is the 'Query editor' pane, which contains a SQL query:

```

1 #standardSQL
2 SELECT
3   `weight_pounds`, state, year, gestation_weeks
4 FROM
5   `bigquery-public-data.samples.natality`
6 ORDER BY weight_pounds DESC LIMIT 10;

```

Below the query editor are several buttons: 'Run', 'Save query', 'Save view', 'Schedule query', 'More', and 'CREATE DATASET'. A green checkmark icon is next to the 'CREATE DATASET' button, indicating it's available. The main area below the editor displays the message: 'This project has no datasets'.

Fig7

Set **Dataset ID** to **babynames**. Leave all other fields at their default settings. Click **Create dataset**.

This screenshot shows the 'Create dataset' dialog box open on the right side of the screen. The 'Dataset ID' field is filled with 'babynames'. The 'Data location (Optional)' dropdown is set to 'Default'. Under 'Default table expiration', the 'Never' radio button is selected. In the 'Encryption' section, the 'Google-managed key' radio button is selected, with the note 'Data is encrypted automatically. Select an encryption key management solution.' Below it are 'No configuration required' and 'Customer-managed key' options. At the bottom of the dialog are 'Create dataset' and 'Cancel' buttons. The background shows the BigQuery interface with the 'Resources' sidebar expanded to show the 'qwiklabs-gcp-02-78ff804247b5' project, and a terminal window at the bottom showing gcloud command output.

Fig8

Step – 4 : In Cloud Shell, run the following commands to add the data files to project:

- \$gsutil cp gs://spl/gsp072/baby-names.zip .
- \$unzip baby-names.zip

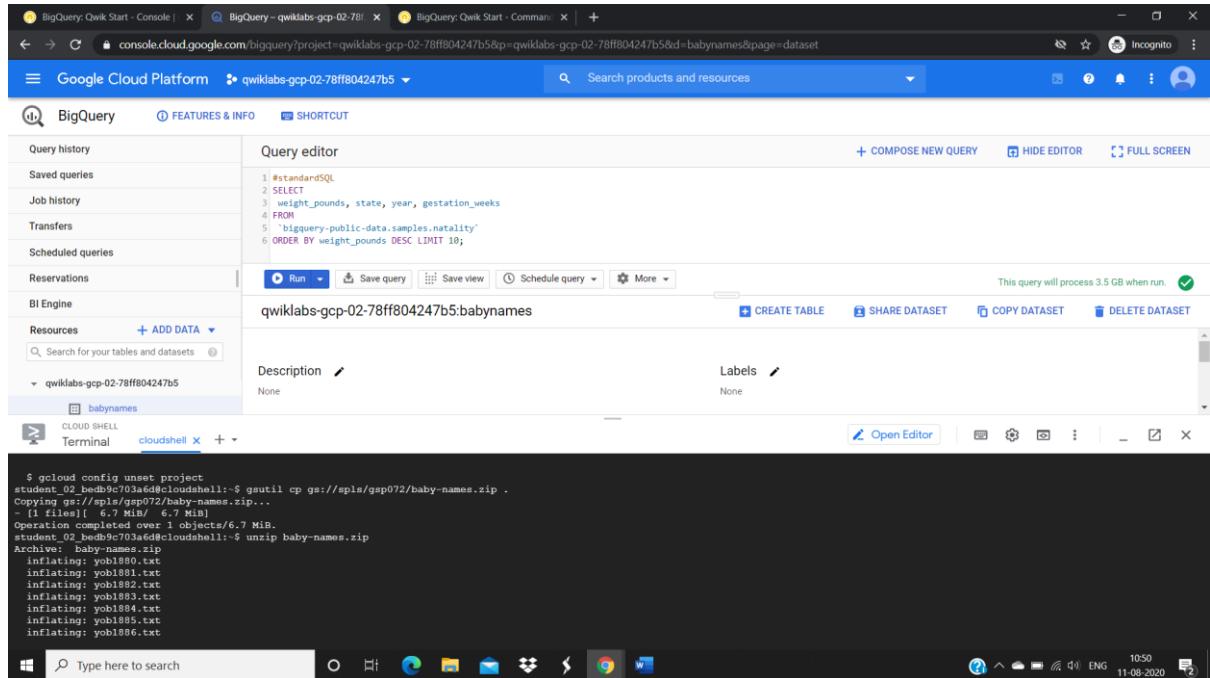


Fig9

Step – 5 : Now create a Cloud Storage bucket to hold the data files you downloaded.

1. In the Cloud Console, select **Navigation menu > Storage > Browser**, and then click **Create bucket**.
2. Give your bucket a universally unique name, then click **Create**.

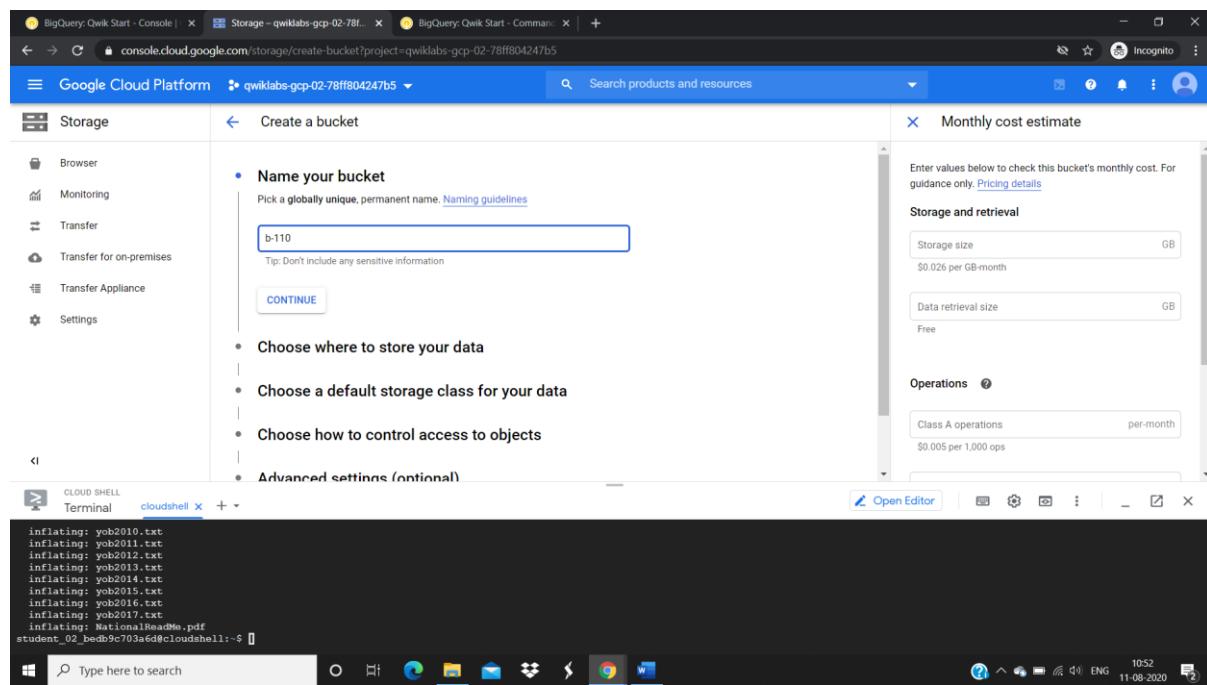


Fig10

Step – 6: In Cloud Shell, run the following to move file yob2014.txt into bucket.
Replace <bucket> with the name of the bucket just created:

- \$gsutil cp yob2014.txt gs://<bucket>

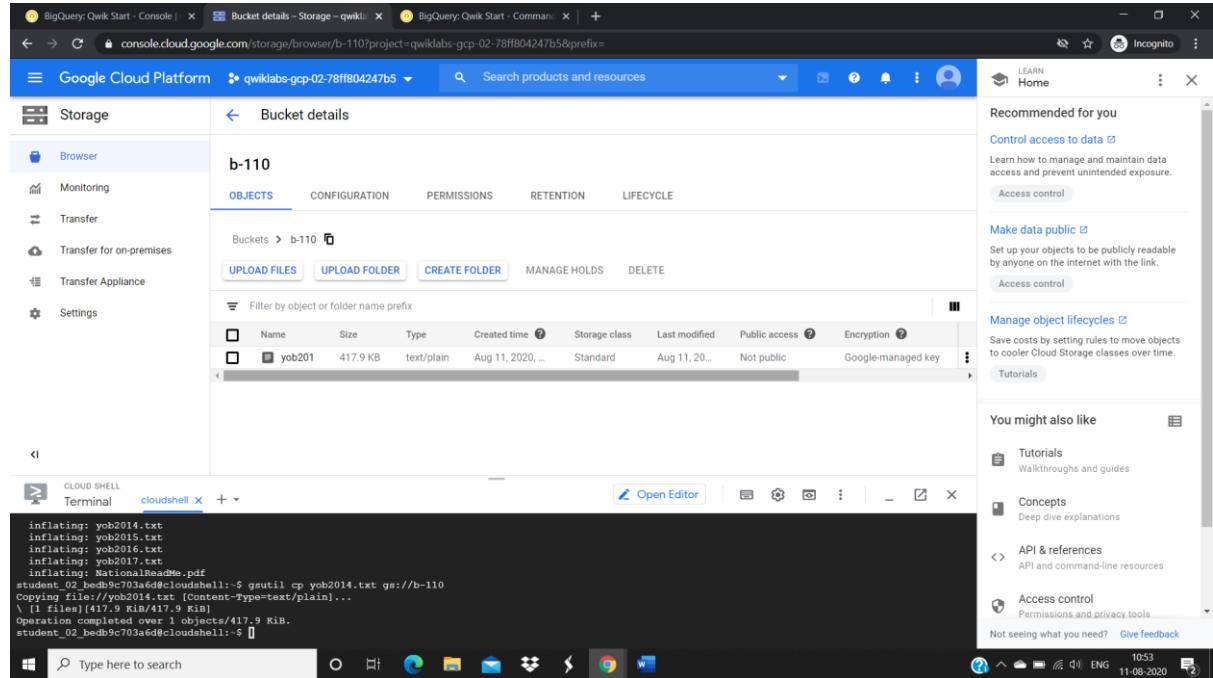


Fig11

Step - 7 : Load the data into a new table

Next create a table inside the babynames dataset, then load the data file from storage bucket into the new table.

1. In the Cloud Console, select **Navigation menu > BigQuery** to return to the BigQuery console.
2. Navigate to the **babynames** dataset, then click **Create table**.

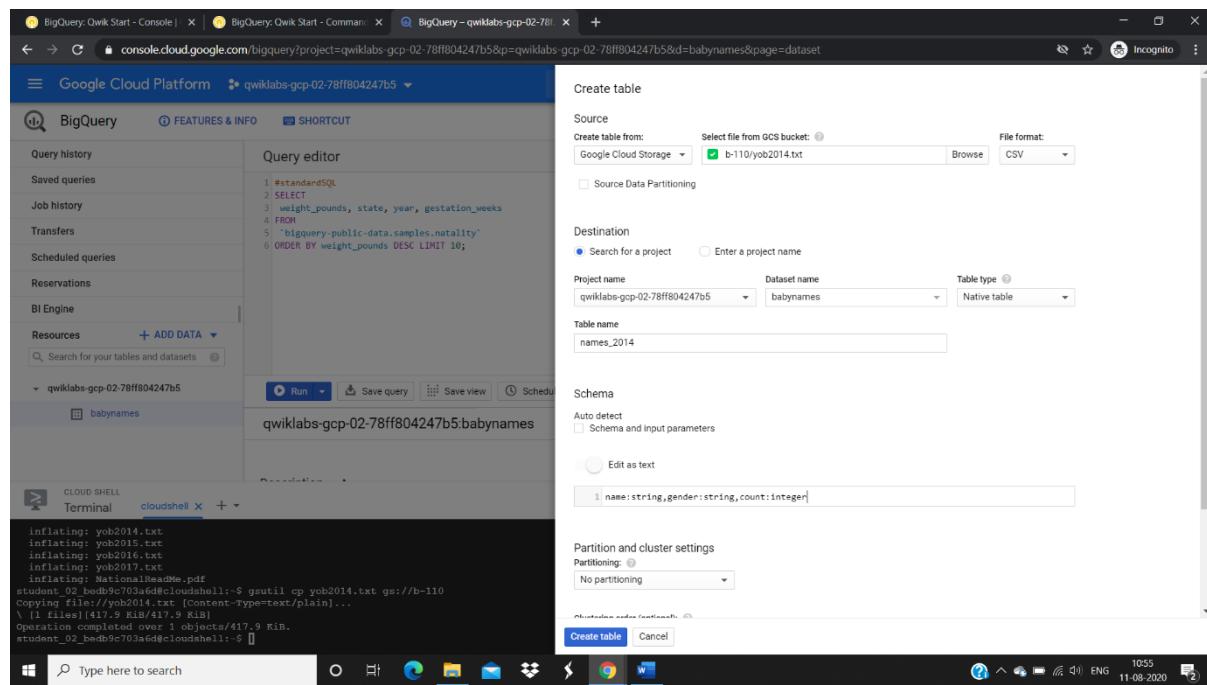


Fig12

Resources **+ ADD DATA** ▾

 Search for your tables and datasets 

▼ **qwiklabs-gcp-3ab2f5dfbad3b7f7**

▼  **babynames**

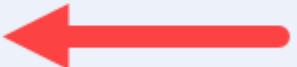
 **names_2014** 

Fig13

Step - 8 : Query a custom dataset

In BigQuery, click the **Compose New query** button in the top right corner to clear out previous query.

Compose a new query as follows:

- \$ #standardSQL
- SELECT
- name, count
- FROM
- 'babynames.names_2014'
- WHERE
- gender = 'M'
- ORDER BY count DESC LIMIT 5;

Click the **Run** button.

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays 'Query history', 'Saved queries', 'Job history', 'Transfers', 'Scheduled queries', 'Reservations', and 'BI Engine'. Under 'Resources', it shows 'qwiklabs-gcp-02-78ff804247b5' with 'babynames' and 'names_2014' datasets selected. The main area is the 'Query editor' containing the following SQL code:

```
1 #standardSQL
2 SELECT
3   name, count
4   FROM
5     `babynames.names_2014`
6   WHERE
7     gender = 'M'
8   ORDER BY count DESC LIMIT 5;
```

Below the editor, the 'Query results' section shows the output of the query:

Row	name	count
1	Noah	19286
2	Liam	18451
3	Mason	17192
4	Jacob	16669
5	William	16809

The status bar at the bottom indicates 'This query will process 622.2 KB when run.' and shows the date '11-08-2020'.

Fig14

The screenshot shows a browser window with the URL google.qwiklabs.com/focuses/1145?parent=catalog#. The main content is the 'BigQuery: Qwik Start - Console' page. At the top right, there is a green box displaying 'Score 100/100'. On the left, there's a sidebar with 'End Lab' and a timer showing '00:10:36'. Below the timer is a section titled 'Open Google Console' with a note about caution regarding lab instructions. The main area features a large 'Congratulations!' message and a badge icon. To the right, a vertical sidebar lists various lab topics: GSP072, Overview, Setup and Requirements, Open BigQuery, Query a public dataset, Load custom data into a table, Add custom data, Create a Cloud Storage bucket, Load the data into a new table, Test your Understanding, Preview the table, Query a custom dataset, and Congratulations!. At the bottom, there's a search bar, a taskbar with icons, and a system tray showing the date and time.

Fig15

BigQuery: Qwik Start – Command Line

Step – 1 : Activate Google Cloud Shell. By Clicking on the icon in Fig1.

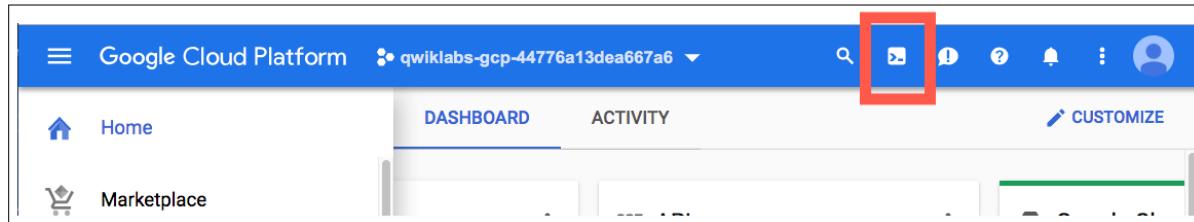


Fig1

Then Click Continue

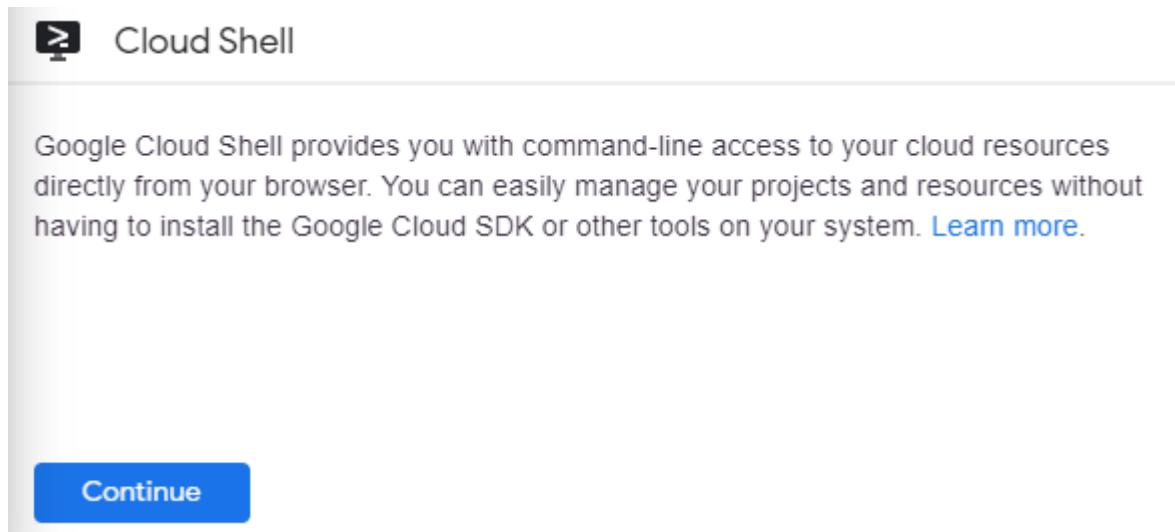
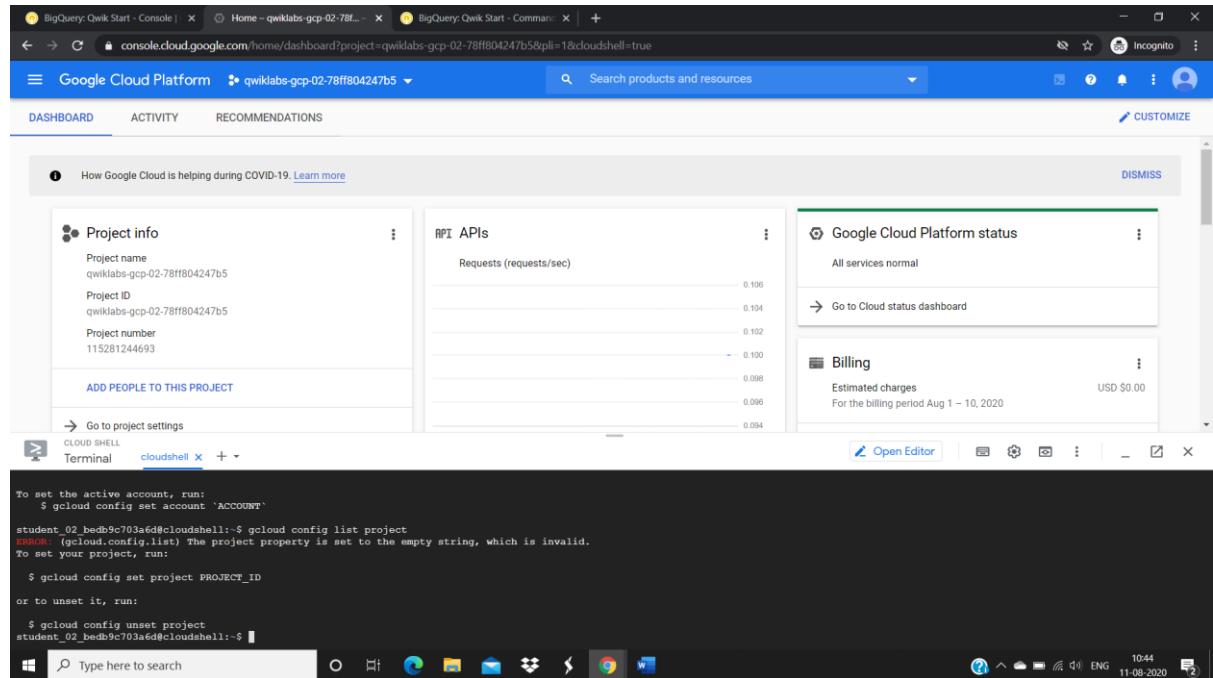


Fig2

Use the following commands

- List the active account name \$gcloud auth list
- List the project ID \$gcloud config list project



The screenshot shows the Google Cloud Platform dashboard for the project 'qwiklabs-gcp-02-78ff804247b5'. The dashboard includes sections for Project info, APIs, Google Cloud Platform status, and Billing. Below the dashboard is a Cloud Shell terminal window. The terminal output shows the following commands:

```
To set the active account, run:  
$ gcloud config set account `ACCOUNT`  
  
student_02_bedb9c703af6@cloudshell:~$ gcloud config list project  
ERROR: (gcloud.config.list) The project property is set to the empty string, which is invalid.  
To set your project, run:  
  
$ gcloud config set project PROJECT_ID  
or to unset it, run:  
  
$ gcloud config unset project  
student_02_bedb9c703af6@cloudshell:~$
```

Fig3

Step – 2 : Examine a table

Run queries against the shakespeare table, which contains an entry for every word in every play.

To examine the schema of the Shakespeare table in the samples dataset, run:

- \$bq show bq show bigquery-public-data:samples.shakespeare

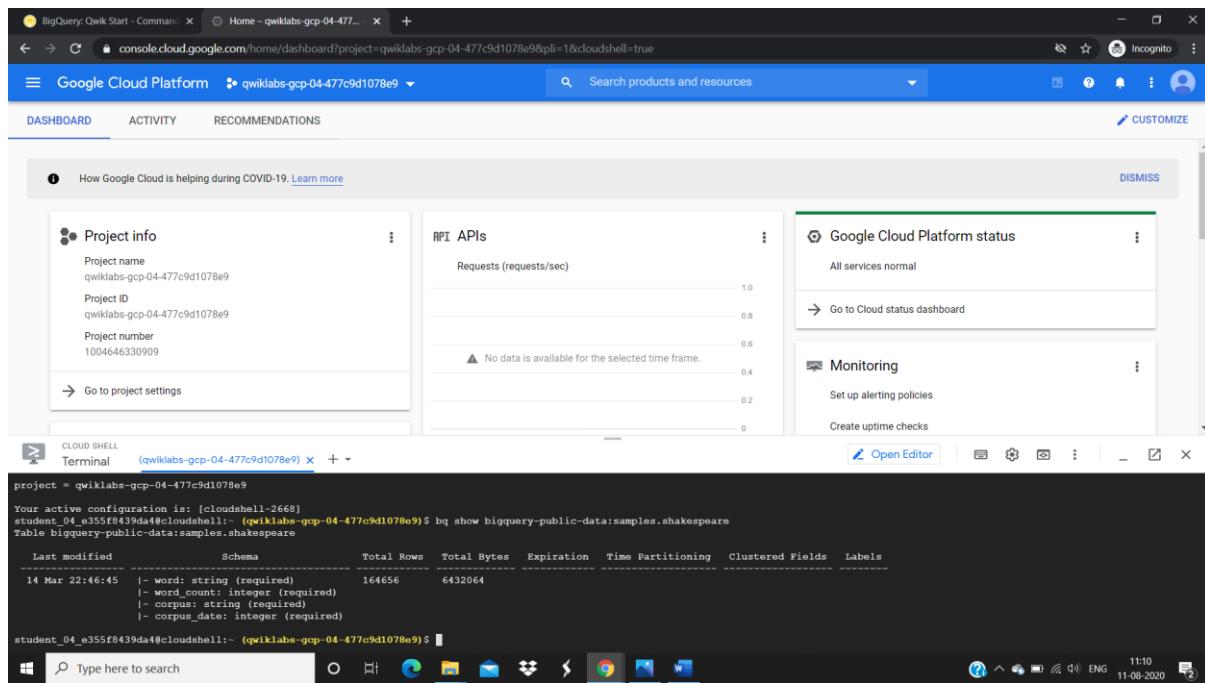


Fig4

Step – 3 : Run the help command. To see a list of all of the commands bq uses, run just bq help.

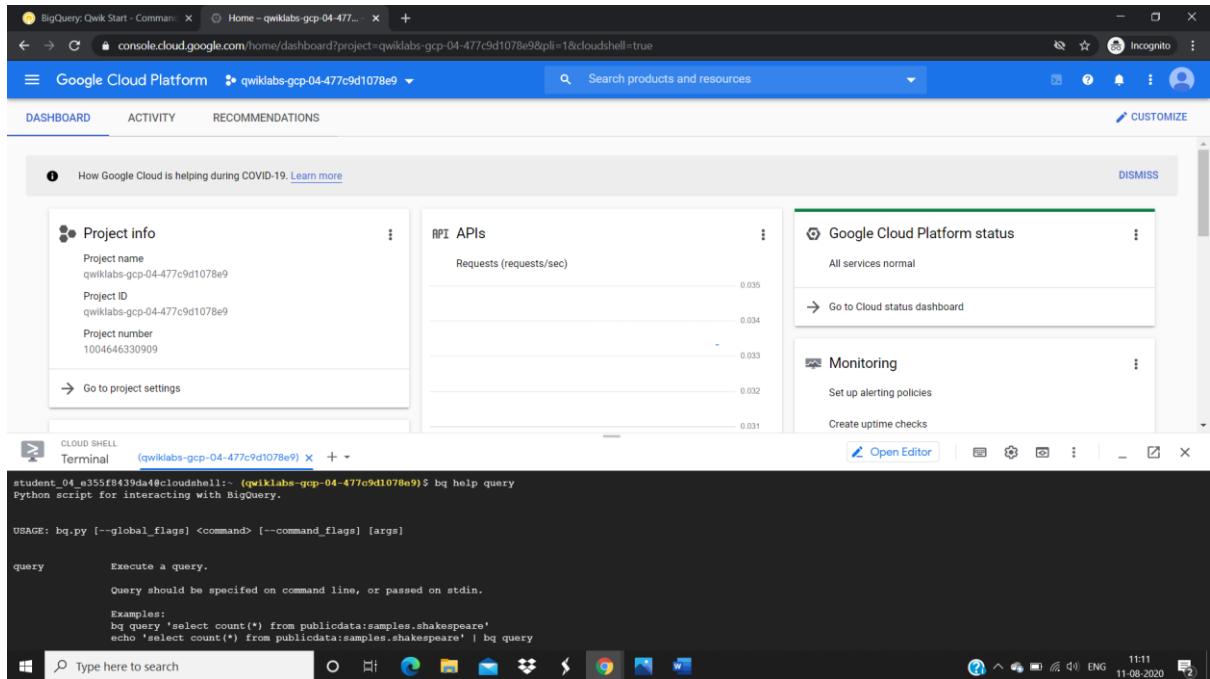
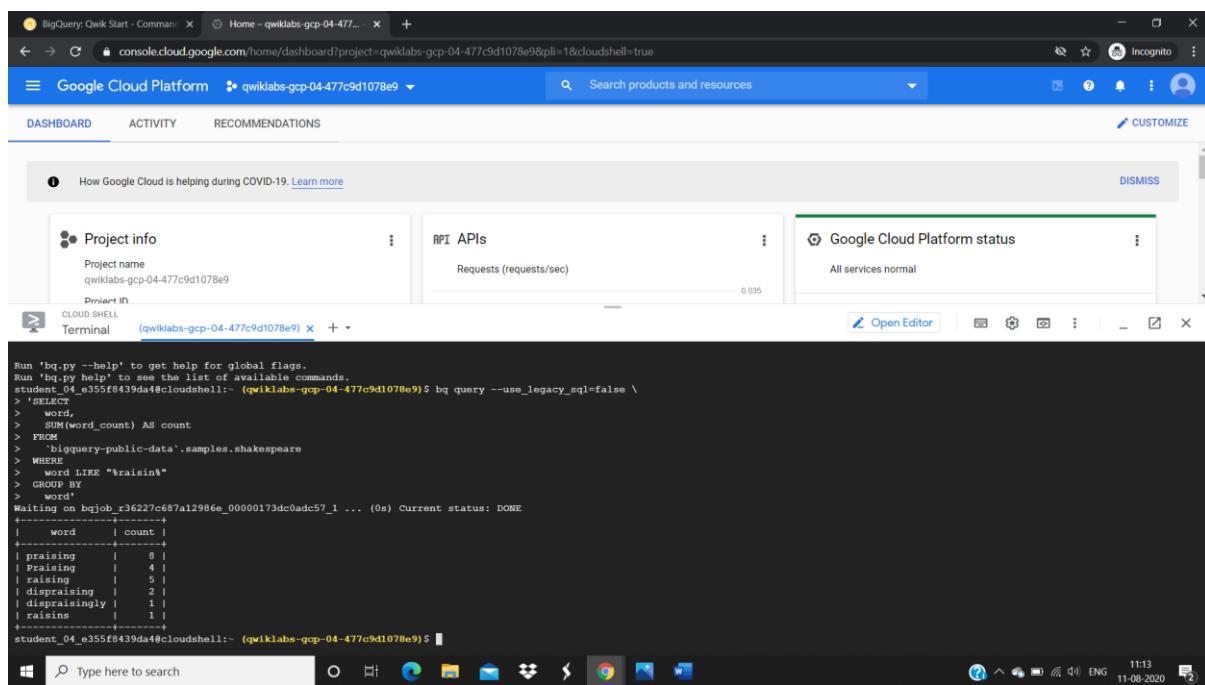


Fig5

Step – 4 : Run a query

Run the following standard SQL query in Cloud Shell to count the number of times that the substring "raisin" appears in all of Shakespeare's works:

- \$bq query --use legacy sql=false \
- 'SELECT
- word,
- SUM(word count) AS count
- FROM
- 'bigquery-public-data'.samples.shakespeare
- WHERE
- Word LIKE "%raisin%"
- GROUP BY
- Word'



The screenshot shows a Google Cloud Platform dashboard with a Cloud Shell terminal open. The terminal window title is '(qwiklabs-gcp-04-477c9d1078e9)'. The terminal content displays a standard SQL query being run against the 'bigquery-public-data'.samples.shakespeare dataset to count occurrences of the substring 'raisin'.

```
Run 'bq.py --help' to get help for global flags.  
Run 'bq.py help' to see the list of available commands.  
student_04_e355f8439da4@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$ bq query --use_legacy_sql=false \  
> 'SELECT  
> word,  
> SUM(word_count) AS count  
> FROM  
> 'bigquery-public-data'.samples.shakespeare  
> WHERE  
> Word LIKE "%raisin%"  
> GROUP BY  
> Word'  
Waiting on bqjob_r36227c687a12986e_00000173dc0adc57_1 ... (0s) Current status: DONE  
+-----+-----+  
| word | count |  
+-----+-----+  
| praising | 8 |  
| praisingly | 4 |  
| raisin | 5 |  
| dispraising | 2 |  
| dispraisingly | 1 |  
| raisins | 1 |  
+-----+-----+
```

Fig6

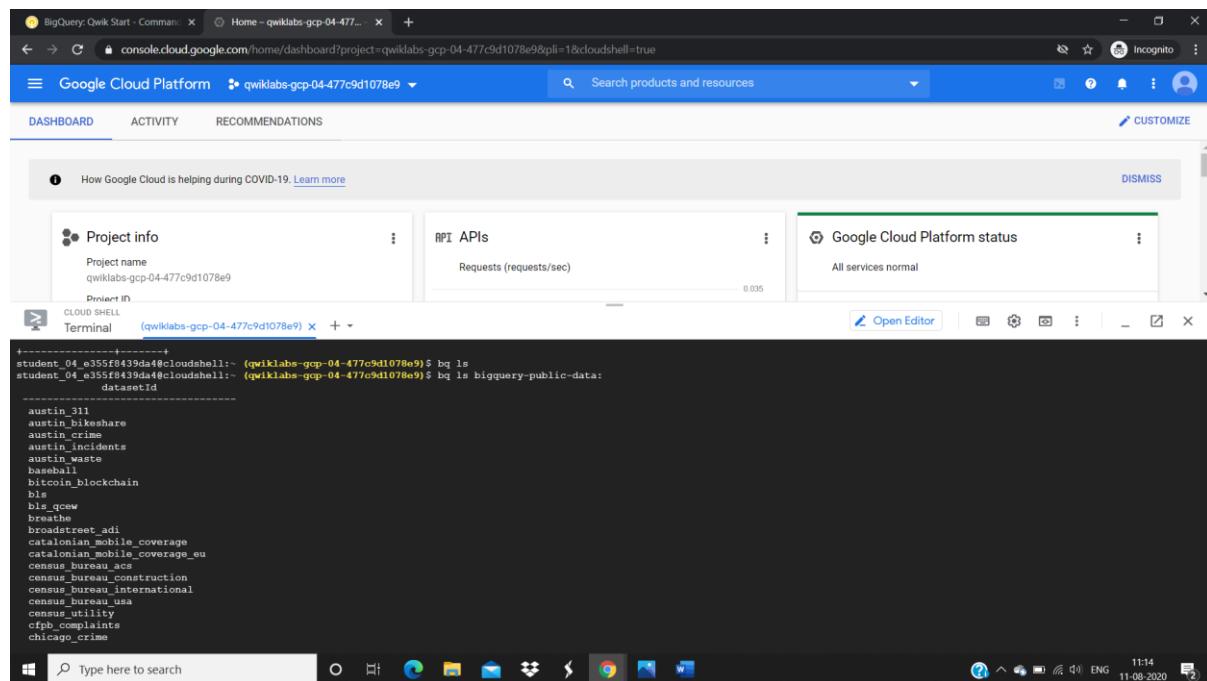
Step – 5: Create a new table and dataset.

Use the bq ls command to list any existing datasets in your project:

- \$bq ls

Run bq ls and the `bigrquery-public-data` Project ID to list the datasets in that specific project, followed by a colon (:).

- \$bq ls `bigrquery-public-data`:



The screenshot shows the Google Cloud Platform dashboard with a terminal window open. The terminal window displays the output of the `bq ls` command, listing various datasets such as `austin_bikeshare`, `austin_crime`, `austin_incidents`, `austin_waste`, `baseball`, `bitcoin_blockchain`, `btc`, `bts_qcow`, `breathe`, `broadstreet_adi`, `catalonian_mobile_coverage`, `catalonian_mobile_coverage_eu`, `census`, `census_bureau_construction`, `census_bureau_international`, `census_bureau_usa`, `census_utility`, `cfpb_complaints`, and `chicago_crime`. The terminal window is titled "Terminal (qwiklabs-gcp-04-477c9d1078e9)" and is running in Cloud Shell.

```
+-----+
student_04 e355f8439da4@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$ bq ls
student_04 e355f8439da4@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$ bq ls bigrquery-public-data:
datasetId
-----+
austin_bikeshare
austin_crime
austin_incidents
austin_waste
baseball
bitcoin_blockchain
btc
bts_qcow
breathe
broadstreet_adi
catalonian_mobile_coverage
catalonian_mobile_coverage_eu
census
census_bureau_construction
census_bureau_international
census_bureau_usa
census_utility
cfpb_complaints
chicago_crime
```

Fig7

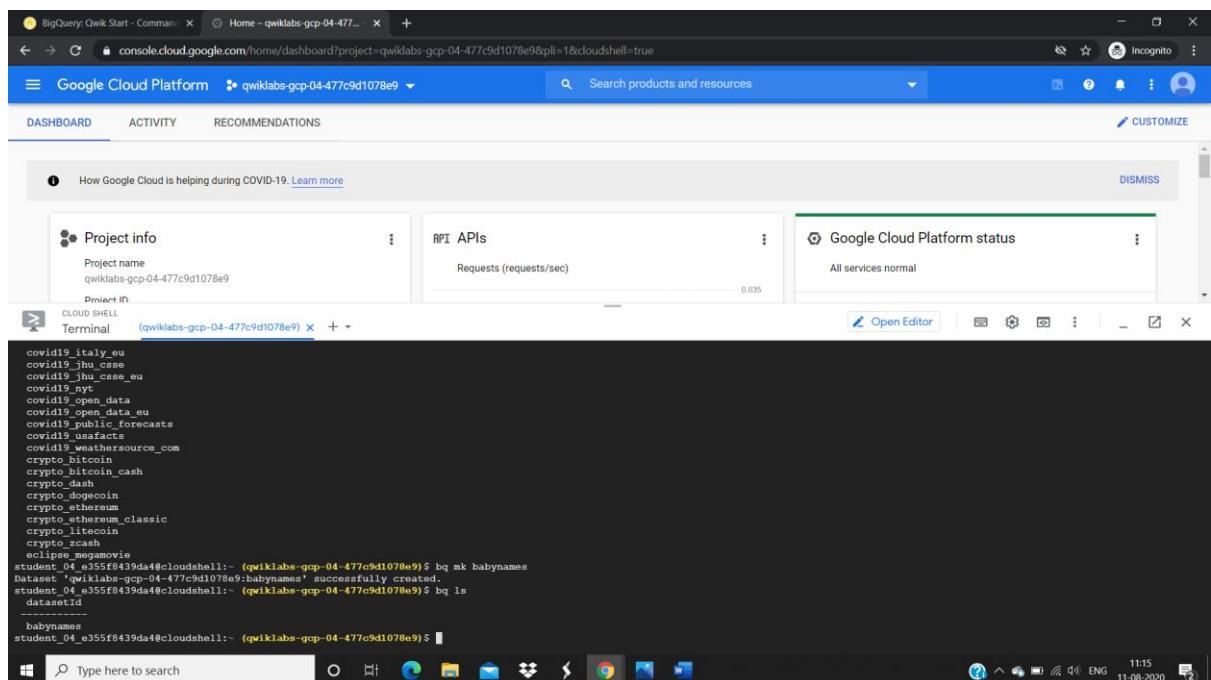
Step – 6 : Create a new dataset named babynames.

Use the bq mk command to create a new dataset named babynames in Qwiklabs project:

- \$bq mk babynames

Run bq ls to confirm that the dataset now appears as part of project:

- \$bq ls



The screenshot shows a Google Cloud Platform dashboard with a terminal window open. The terminal window displays the following commands and their output:

```
bq mk babynames
Dataset 'qwiklabs-gcp-04-477c9d1078e9:babynames' successfully created.
bq ls
datasetId
-----
babynames
```

Fig8

Step – 7 : Upload the dataset

Run this command to add the baby names zip file to your project, using the URL for the data file:

- \$wget http://www.ssa.gov/OACT/babynames/names.zip

List the file:

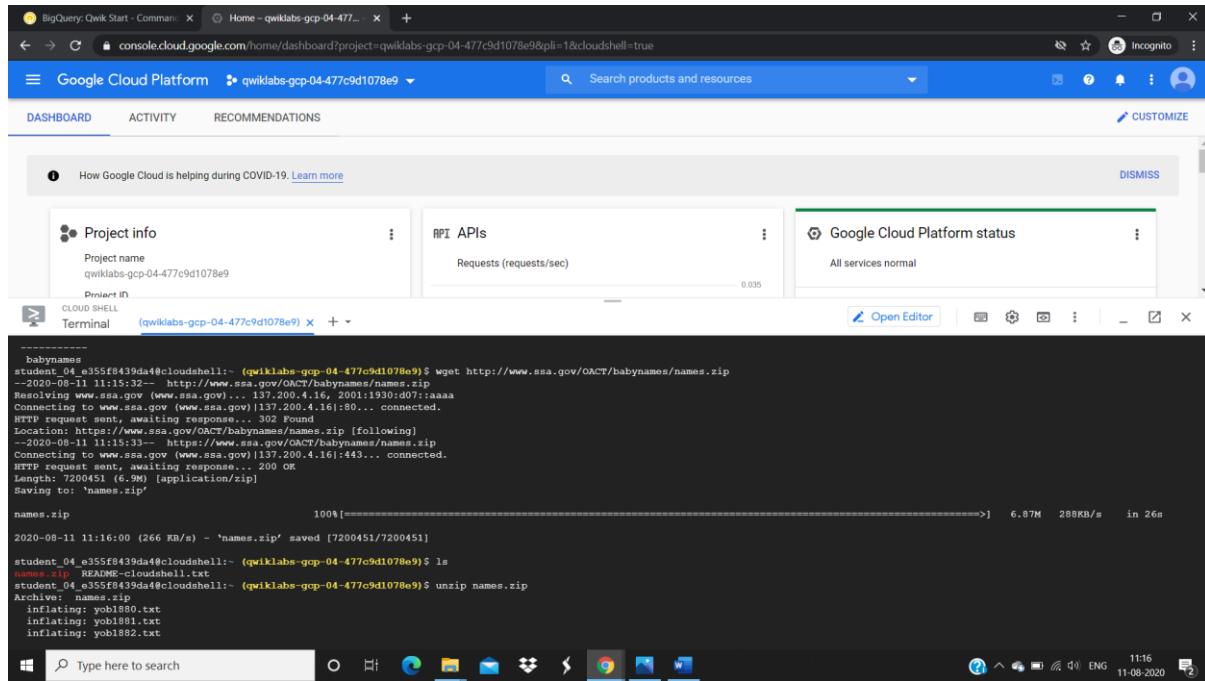
- \$ls

Now unzip the file:

- \$unzip names.zip

That's a pretty big list of text files! List the files again:

- \$ls



The screenshot shows a Google Cloud Platform dashboard with a terminal window open. The terminal window displays the following commands and their output:

```
student_04 e355f8439da@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$ wget http://www.ssa.gov/OACT/babynames/names.zip
--2020-08-11 11:15:32-- http://www.ssa.gov/OACT/babynames/names.zip
Resolving www.ssa.gov (www.ssa.gov)... 137.200.4.16, 2001:1930:d07::aaaa
Connecting to www.ssa.gov (www.ssa.gov)|137.200.4.16|:80... connected.
HTTP request sent, awaiting response... 202 Found
Location: https://www.ssa.gov/OACT/babynames/names.zip [following]
--2020-08-11 11:15:33-- https://www.ssa.gov/OACT/babynames/names.zip
Connecting to www.ssa.gov (www.ssa.gov)|137.200.4.16|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7200451 (6.9M) [application/zip]
Saving to: 'names.zip'

names.zip                                     100%[=====]   6.87M   288KB/s   in 26s

2020-08-11 11:16:00 (266 KB/s) - 'names.zip' saved [7200451/7200451]

student_04 e355f8439da@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$ ls
names.zip  README.cloudshell.txt
student_04 e355f8439da@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$ unzip names.zip
Archive:  names.zip
  inflating: yob1880.txt
  inflating: yob1881.txt
  inflating: yob1882.txt
```

Fig9

Step - 8 : The bq load command creates or updates a table and loads data in a single step.

The bq load arguments you'll be running are:

- \$datasetID: babynames
- tableID: names2010
- source: yob2010.txt
- schema: name:string,gender:string,count:integer

Create your table:

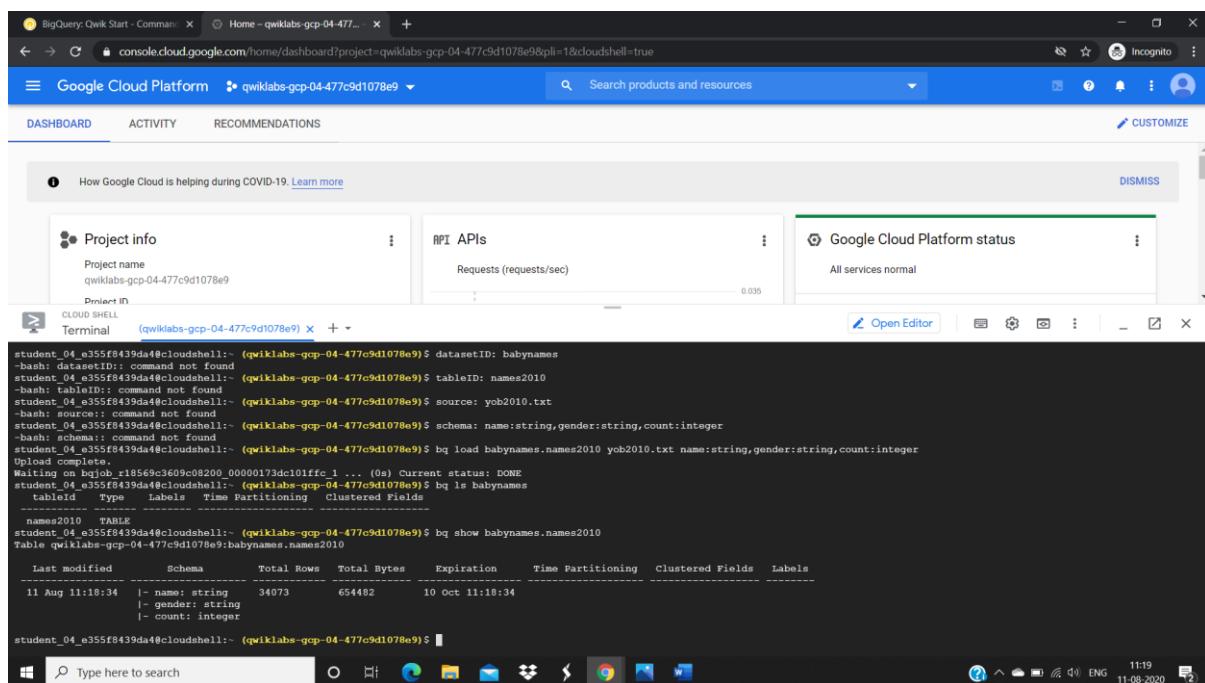
- \$bq load babynames.names2010 yob2010.txt
- names:string,gender:string,count:integer

Run bq ls and babynames to confirm that the table now appears your dataset:

- \$bq ls babynames

Run bq show and your dataset.table to see the schema:

- \$bq show babynames.names2010



The screenshot shows a Google Cloud Platform dashboard with a terminal window open in a browser tab. The terminal window displays the following sequence of commands:

```
student_04_e355f8439da4@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$ datasetID: babynames
student_04_e355f8439da4@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$ tableID: names2010
student_04_e355f8439da4@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$ source: yob2010.txt
student_04_e355f8439da4@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$ schema: name:string,gender:string,count:integer
student_04_e355f8439da4@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$ bq load babynames.names2010 yob2010.txt name:string,gender:string,count:integer
Upload complete.
Waiting for bqjob_r18569c360c08200_00000173dc10iffc_1 ... (0s) Current status: DONE
student_04_e355f8439da4@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$ bq ls babynames
  tableId   Type Labels Time Partitioning Clustered Fields
  names2010  TABLE
student_04_e355f8439da4@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$ bq show babynames.names2010
Table qwiklabs-gcp-04-477c9d1078e9:babynames.names2010
  Last modified      Schema      Total Rows      Total Bytes      Expiration      Time Partitioning      Clustered Fields      Labels
  11 Aug 11:18:34    |- name: string      34073      654462      10 Oct 11:18:34
                    |- gender: string
                    |- count: integer
student_04_e355f8439da4@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$
```

Fig10

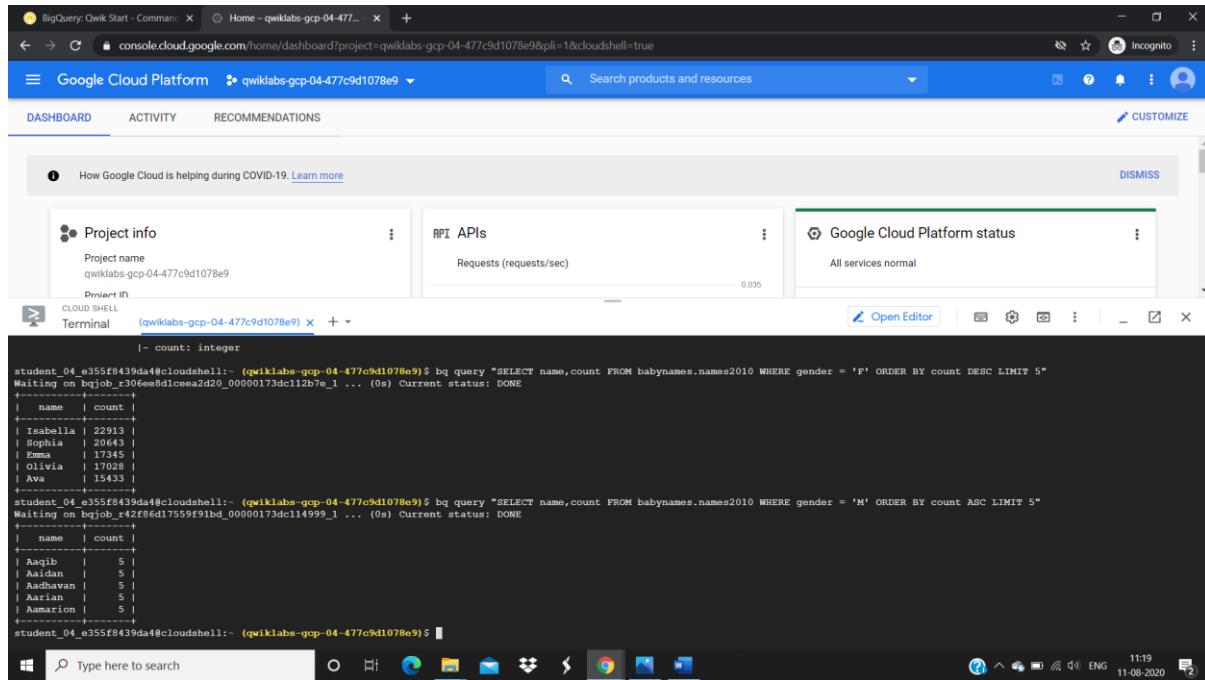
Step – 9 : Run Queries

Run the following command to return the top 5 most popular girls names:

- \$bq query “SELECT name,count FROM babynames.names2010 WHERE gender = ‘F’ ORDERBY count DESC LIMIT 5”

Run the following command to see the top 5 most unusual boys names.

- \$bq query “SELECT name,count FROM babynames.names2010 WHERE gender = ‘M’ ORDERBY count ASC LIMIT 5”



The screenshot shows the Google Cloud Platform dashboard with a terminal window open. The terminal window title is 'CLOUD SHELL Terminal (qwiklabs-gcp-04-477c9d1078e9)'. The terminal output shows two BigQuery commands being run:

```
student_04_e355f8439da4@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$ bq query "SELECT name, count FROM babynames.names2010 WHERE gender = 'F' ORDER BY count DESC LIMIT 5"
Waiting on bqjob_r306ee8d1ceea2d20_00000173dc112b7e_1 ... (0s) Current status: DONE
+-----+-----+
| name | count |
+-----+-----+
| Isabella | 22913 |
| Sophia | 20543 |
| Emma | 17345 |
| Olivia | 17028 |
| Ava | 15433 |
+-----+-----+
student_04_e355f8439da4@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$ bq query "SELECT name, count FROM babynames.names2010 WHERE gender = 'M' ORDER BY count ASC LIMIT 5"
Waiting on bqjob_r42f86d17559f91bd_00000173dc114999_1 ... (0s) Current status: DONE
+-----+-----+
| name | count |
+-----+-----+
| Aaqib | 5 |
| Aidan | 5 |
| Aadhavan | 5 |
| Aarian | 5 |
| Aarion | 5 |
+-----+-----+
student_04_e355f8439da4@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$
```

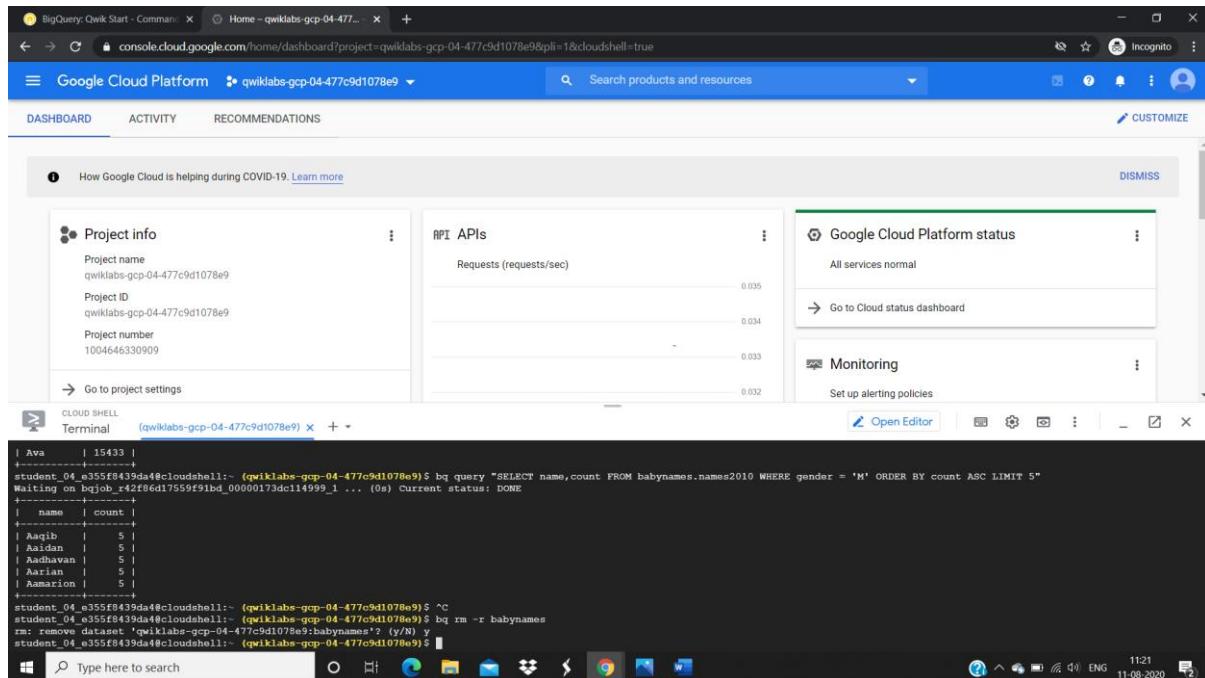
Fig11

Step – 10 : Clean up

Run the bq rm command to remove the babynames dataset with the -r flag to delete all tables in the dataset.

- \$bq rm -r babynames

Confirm the delete command by typing "y".



The screenshot shows the Google Cloud Platform dashboard with a terminal window open in Cloud Shell. The terminal window displays the following commands and output:

```
student_04_e355f8439da4@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$ bq query "SELECT name, count FROM babynames.names2010 WHERE gender = 'M' ORDER BY count ASC LIMIT 5"
Waiting on bqjob_r42f86d17559f91bd_00000173dc114999_1... (0s) Current status: DONE
+-----+-----+
| name | count |
+-----+-----+
| Aaqib | 5 |
| Aaidan | 5 |
| Aadhavan | 5 |
| Aarian | 5 |
| Aamaron | 5 |
+-----+
student_04_e355f8439da4@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$ bq rm -r babynames
rm: remove dataset 'qwiklabs-gcp-04-477c9d1078e9:babynames'? (y/N) y
student_04_e355f8439da4@cloudshell:~ (qwiklabs-gcp-04-477c9d1078e9)$
```

Fig12

BigQuery: Qwik Start - Console | BigQuery: Qwik Start - Command Line | Home - qwiklabs-gcp-02-4f9... - | You can access BigQuery using: | +

← BigQuery: Qwik Start - Command Line

End Lab 00:04:37

Open Google Console

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. [Learn more.](#)

Username: student-02-6c0e1950c5fd@qwiklabs

Password: dqX5g37QN9

GCP Project ID: qwiklabs-gcp-02-4f95fc34786b

Student Resources

- [Get Meaningful Insights with Google BigQuery](#)
- [BigQuery: Qwik Start - Qwiklabs Preview](#)

Score 100/100

Congratulations!

Now you can use the command line with BigQuery to manipulate data.

Finish Your Quest

This self-paced lab is part of the Qwiklabs [BigQuery for Data Warehousing, NCCA® March Madness® Bracketology with Google Cloud, BigQuery Basics for Data](#)

Chat

Type here to search

0 11-08-2020 09:38 ENG

Fig13

Lab – 4

Student Name and Roll Number: Chayan 18csu054

Semester /Section: 5/A

Link to Code: <https://github.com/chayangulati321/Big-Data>

Date: 18/08/20

Faculty Signature:

Remarks:

Hadoop Installation

AIM : To install Hadoop on local machine.

Step – 1 : Install Java on local machine through link given below:

- <https://www.oracle.com/in/java/technologies/javase/javase-jdk8-downloads.html>

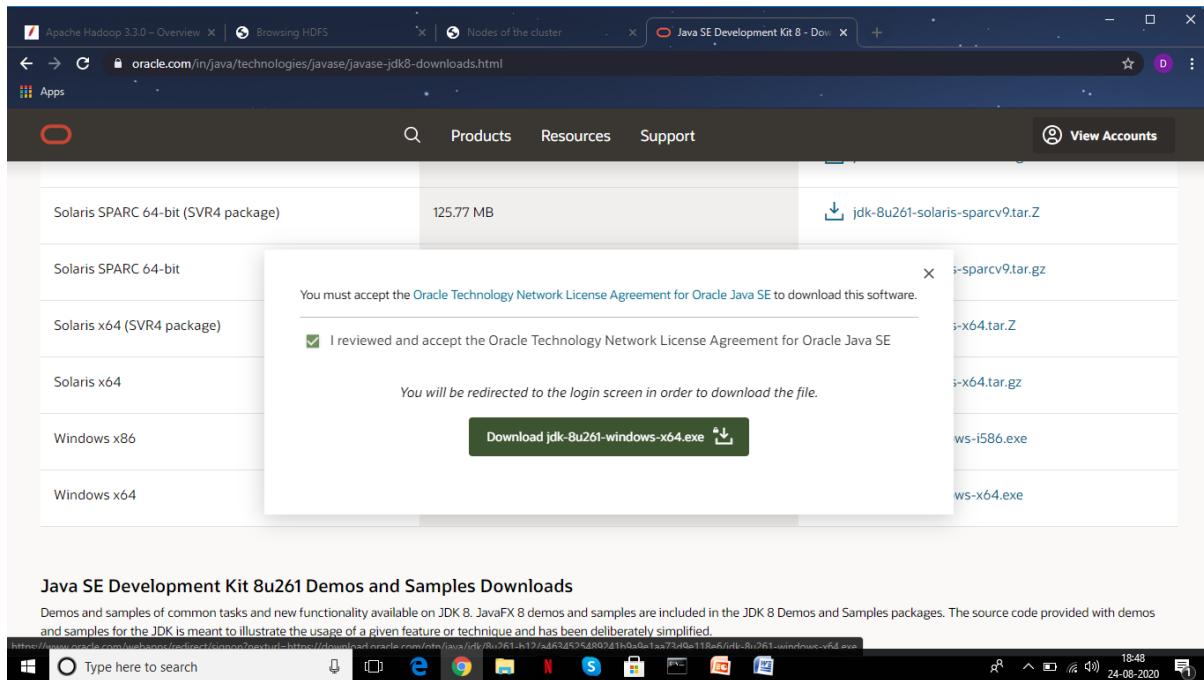


Fig1

Step – 2 : After downloading set the path in environment variable

- Keep the java folder directly under the local disk directory (C:\Java\jdk1.8.0_261) rather than in Program Files (C:\Program Files\Java\jdk1.8.0_261) as it can create errors afterwards.
- Create a new folder named “jre1.8.0_261” in jdk1.8.0_261 directory.
- In environment variable create a new user variable name “JAVA_HOME”=C:\Java\jdk1.8.0_261 and add path C:\Java\jdk1.8.0_261\bin.
- Add the same path into system variables.

Open Command Prompt (cmd). (NOTE: Open cmd as administrator only.)

- \$java -version
- \$javac

If these commands run properly i.e., Java is successfully installed.

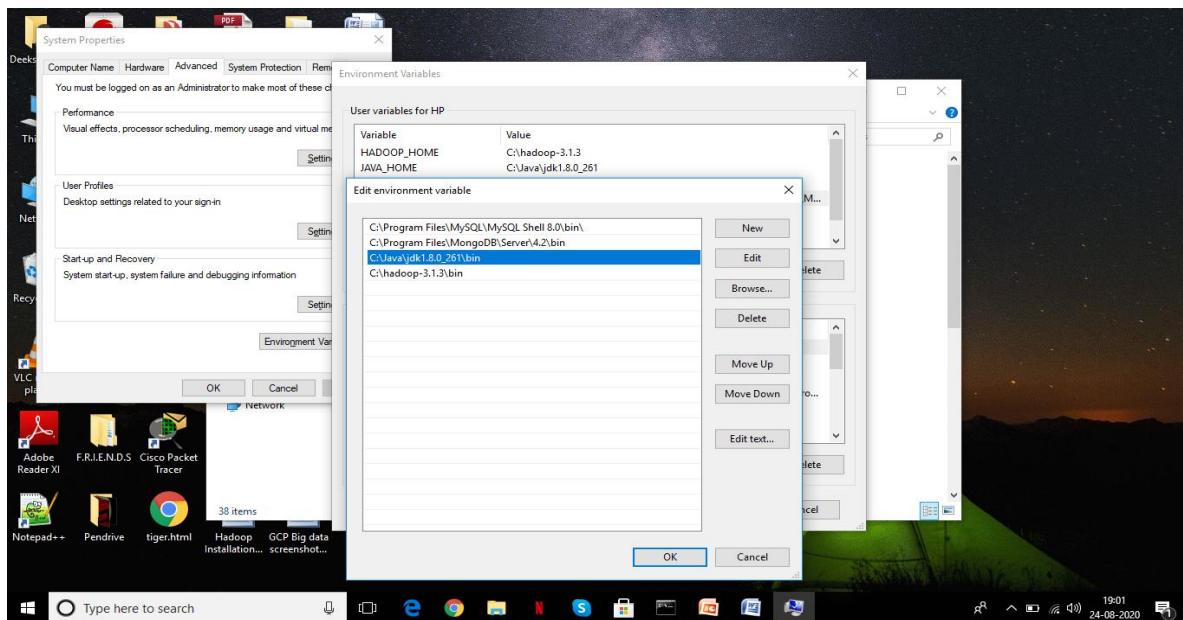


Fig2

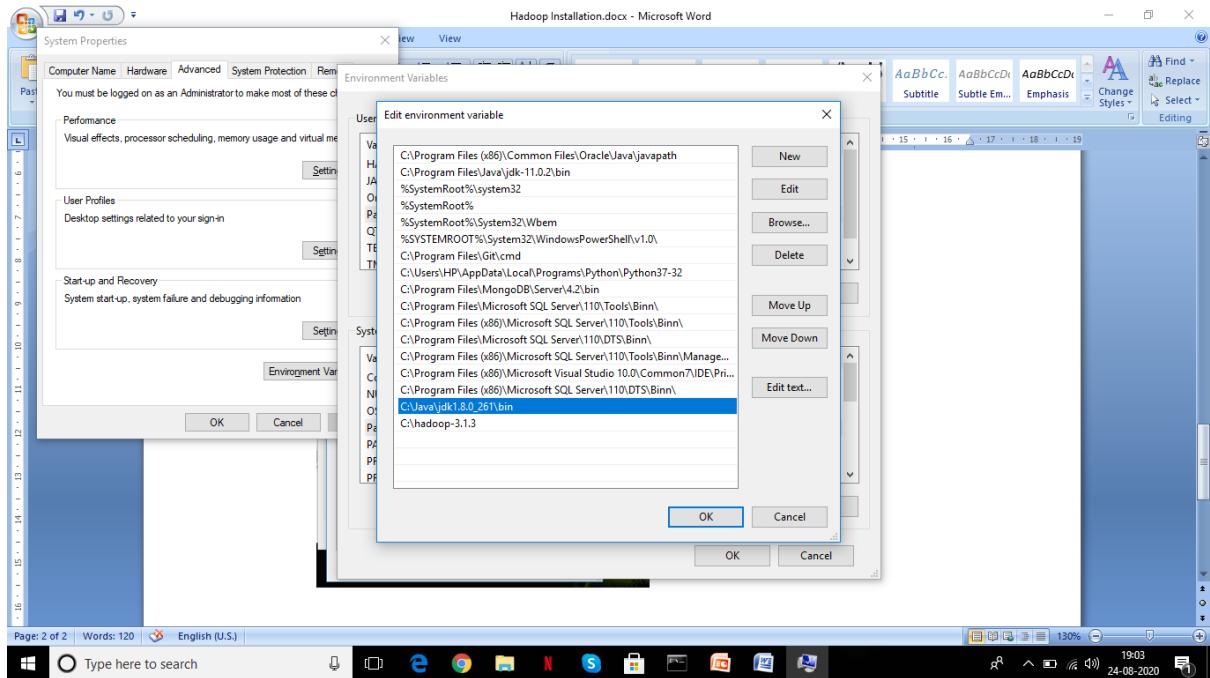


Fig3

Step – 3 : Download binary version for Hadoop 3.1.3 from the following link.

<https://hadoop.apache.org/releases.html>

Or

download the mirror image from this link.

<http://apachemirror.wuchna.com/hadoop/common/hadoop-3.1.3/hadoop-3.1.3.tar.gz>

Extract the tar file into C:/ drive. The path in C:/drive should be like C:\hadoop-3.1.3\bin

Set path for Hadoop as well in environment variable:

- In environment variable create a new user variable name "HADOOP_HOME"=C:\hadoop-3.1.3 and add path C:\hadoop-3.1.3\bin.
- Add the same path into system variables except bin. Path C:\hadoop-3.1.3

Download

Hadoop is released as source code tarballs with corresponding binary tarballs for convenience. The downloads are distributed via mirror sites and should be checked for tampering using GPG or SHA-512.

Version	Release date	Source download	Binary download	Release notes
3.3.0	2020 Jul 14	source (checksum signature)	binary (checksum signature) binary-aarch64 (checksum signature)	Announcement
2.10.0	2019 Oct 29	source (checksum signature)	binary (checksum signature)	Announcement
3.1.3	2019 Oct 21	source (checksum signature)	binary (checksum signature)	Announcement
3.2.1	2019 Sep 22	source (checksum signature)	binary (checksum signature)	Announcement
2.9.2	2018 Nov 19	source (checksum signature)	binary (checksum signature)	Announcement

To verify Hadoop releases using GPG:

1. Download the release `hadoop-X.Y.Z-src.tar.gz` from a [mirror site](#).
2. Download the signature file `hadoop-X.Y.Z-src.tar.gz.asc` from [Apache](#).
3. Download the Hadoop `KEYS` file.
4. `gpg --import KEYS`.

https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.1.3/hadoop-3.1.3.tar.gz

Windows Taskbar: hadoop-3.1.3 (2).tar.gz (3.0/322 MB, Paused) | hadoop-3.1.3 (1).tar.gz (6.8/322 MB, Paused) | Type here to search | Show all | 19:12 24-08-2020

Fig4

System Properties

User variables for HP

Variable	Value
HADOOP_HOME	C:\hadoop-3.1.3

Figure 2.

Hadoop Installation.docx - Microsoft Word

Page: 5 of 5 | Words: 192 | English (U.S.) | Type here to search | 19:20 24-08-2020

Fig5

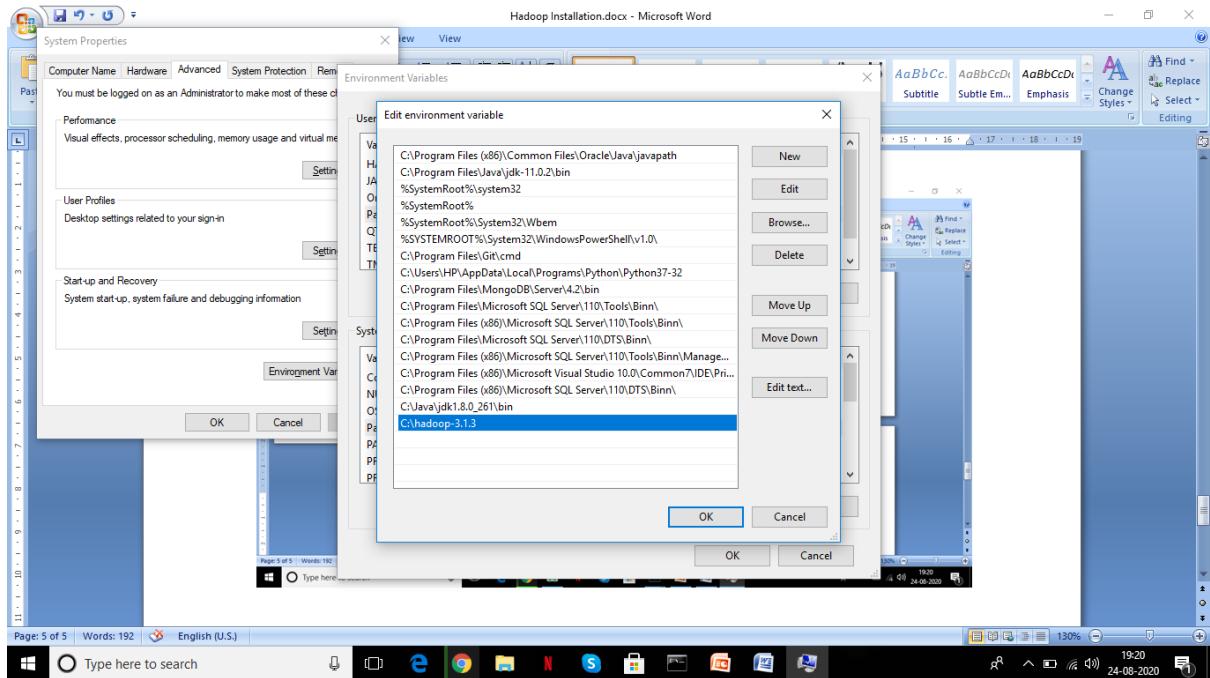


Fig6

Step – 4 : Configure some files located in etc folder under Hadoop-3.1.3

1. core-site.xml
2. 68adoop-env.cmd
3. hdfs-site.xml
4. mapred-site.xml
5. yarn-site.xml
- core-site.xml

```

C:\hadoop-3.1.3\etc\hadoop\core-site.xml - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
01_kmeans.py 02_hc.py core-site.xml mapred-site.xml hdfs-site.xml hadoop-env.cmd yarn-site.xml
4 Licensed under the Apache License, Version 2.0 (the "License");
5 you may not use this file except in compliance with the License.
6 You may obtain a copy of the License at
7
8     http://www.apache.org/licenses/LICENSE-2.0
9
10 Unless required by applicable law or agreed to in writing, software
11 distributed under the License is distributed on an "AS IS" BASIS,
12 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 See the License for the specific language governing permissions and
14 limitations under the License. See accompanying LICENSE file.
15 -->
16
17 <!-- Put site-specific property overrides in this file. -->
18
19 <configuration>
20   <property>
21     <name>fs.defaultFS</name>
22     <value>hdfs://localhost:9000</value>
23   </property>
24 </configuration>
25

```

eXtensible Markup Language file length: 880 lines: 25 Ln:23 Col:15 Sel:0|0 Unix (LF) UTF-8 INS

Type here to search 19:28 24-08-2020

Fig7

- hadoop-env.cmd

```

19 @rem The only required environment variable is JAVA_HOME. All others are
20 @rem optional. When running a distributed configuration it is best to
21 @rem set JAVA_HOME in this file, so that it is correctly defined on
22 @rem remote nodes.
23
24 @rem The java implementation to use. Required.
25 set JAVA_HOME=C:\Java\jdk1.8.0_261
26
27 @rem The jsvc implementation to use. Jsvc is required to run secure datanode

```

Fig8

- hdfs-site.xml

```

C:\hadoop-3.1.3\etc\hadoop\hdfs-site.xml - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
01_kmeans.py 02_hcp.py core-site.xml mapred-site.xml hdfs-site.xml hadoop-env.cmd yarn-site.xml
12 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 See the License for the specific language governing permissions and
14 limitations under the License. See accompanying LICENSE file.
-->
16
17 <!-- Put site-specific property overrides in this file. -->
18
19 <configuration>
20   <property>
21     <name>dfs.replication</name>
22     <value>1</value>
23   </property>
24   <property>
25     <name>dfs.namenode.name.dir</name>
26     <value>/hadoop-3.1.3/data/namenode</value>
27   </property>
28   <property>
29     <name>dfs.datanode.data.dir</name>
30     <value>/hadoop-3.1.3/data/datanode</value>
31   </property>
32 </configuration>
33

```

extensible Markup Language file length : 1,105 lines : 33 Ln : 30 Col : 24 Sel : 0 | 0 Unix (LF) UTF-8 INS

Fig9

- mapred-site.xml

```

C:\hadoop-3.1.3\etc\hadoop\mapred-site.xml - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
01_kmeans.py 02_hcp.py core-site.xml mapred-site.xml hdfs-site.xml hadoop-env.cmd yarn-site.xml
4 Licensed under the Apache License, Version 2.0 (the "License");
5 you may not use this file except in compliance with the License.
6 You may obtain a copy of the License at
7
8   http://www.apache.org/licenses/LICENSE-2.0
9
10 Unless required by applicable law or agreed to in writing, software
11 distributed under the License is distributed on an "AS IS" BASIS,
12 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 See the License for the specific language governing permissions and
14 limitations under the License. See accompanying LICENSE file.
-->
16
17 <!-- Put site-specific property overrides in this file. -->
18
19 <configuration>
20   <property>
21     <name>mapreduce.framework.name</name>
22     <value>yarn</value>
23   </property>
24 </configuration>
25

```

extensible Markup Language file length : 858 lines : 25 Ln : 20 Col : 4 Sel : 0 | 0 Unix (LF) UTF-8 INS

Fig10

- yarn-site.xml

```

6
7     http://www.apache.org/licenses/LICENSE-2.0
8
9     Unless required by applicable law or agreed to in writing, software
10    distributed under the License is distributed on an "AS IS" BASIS,
11    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
12    See the License for the specific language governing permissions and
13    limitations under the License. See accompanying LICENSE file.
14-->
15
16    <!-- Site specific YARN configuration properties -->
17<configuration>
18  <property>
19    <name>yarn.nodemanager.aux-services</name>
20    <value>mapreduce_shuffle</value>
21  </property>
22  <property>
23    <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
24    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
25  </property>
26</configuration>
27

```

eXtensible Markup Language file length: 965 lines: 27 Ln:1 Col:1 Sel:0|0 Unix (LF) UTF-8 INS

Fig11

Step – 5 : Create a new folder ‘data’ under C:\hadoop-3.1.3 and under ‘data’ folder create 2 folders named : ‘namenode’ and ‘datanode’.

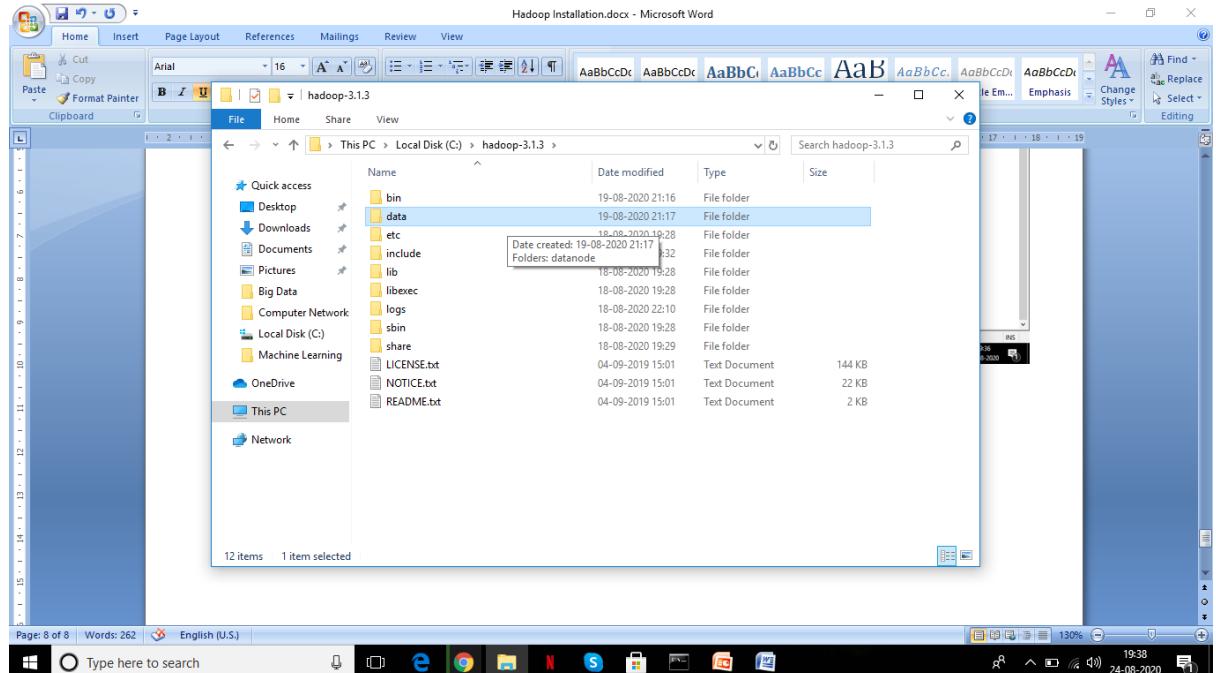


Fig12

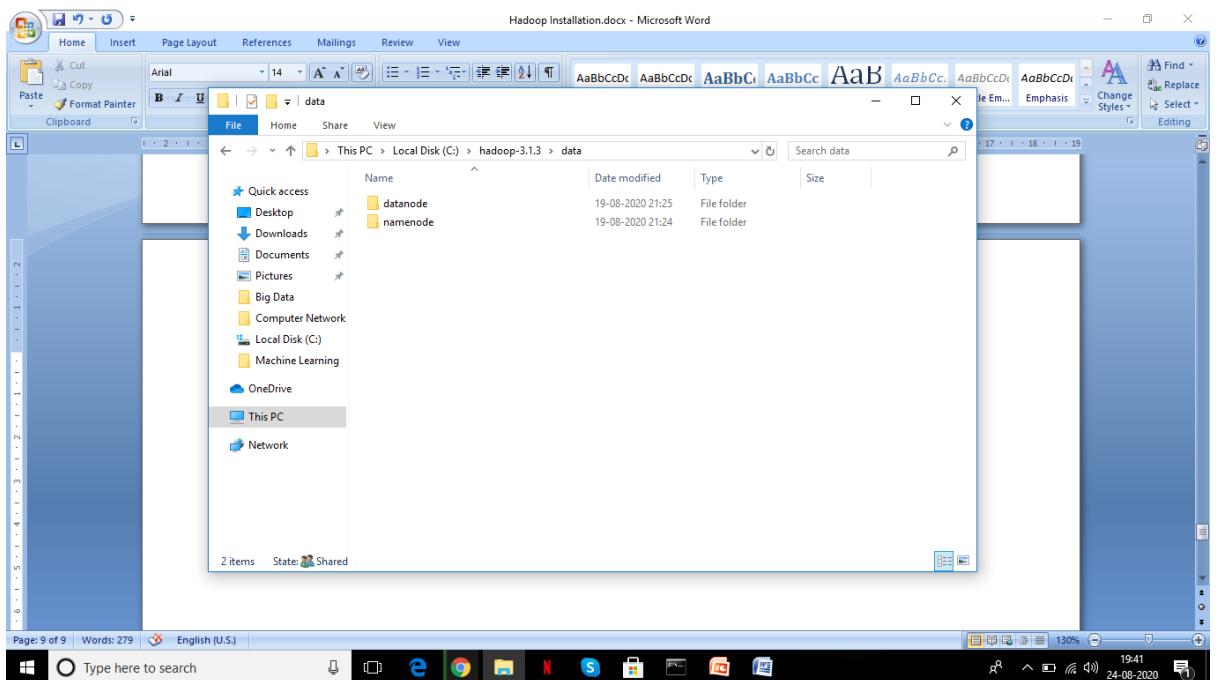


Fig13

Step – 5 : Through the link given below download the zip file 'hadoopconfiguration.zip'

- <https://github.com/s911415/apache-hadoop-3.1.3-winutils>

Extract the file and replaced it with bin folder in C:\hadoop-3.1.3

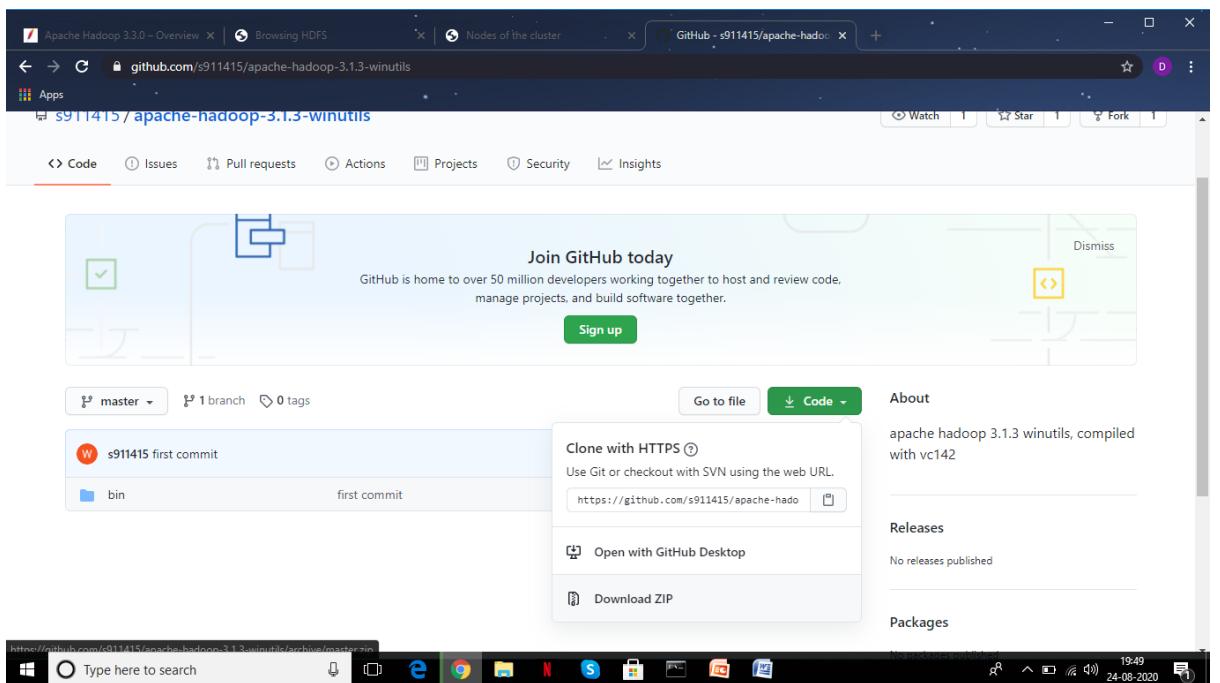


Fig14

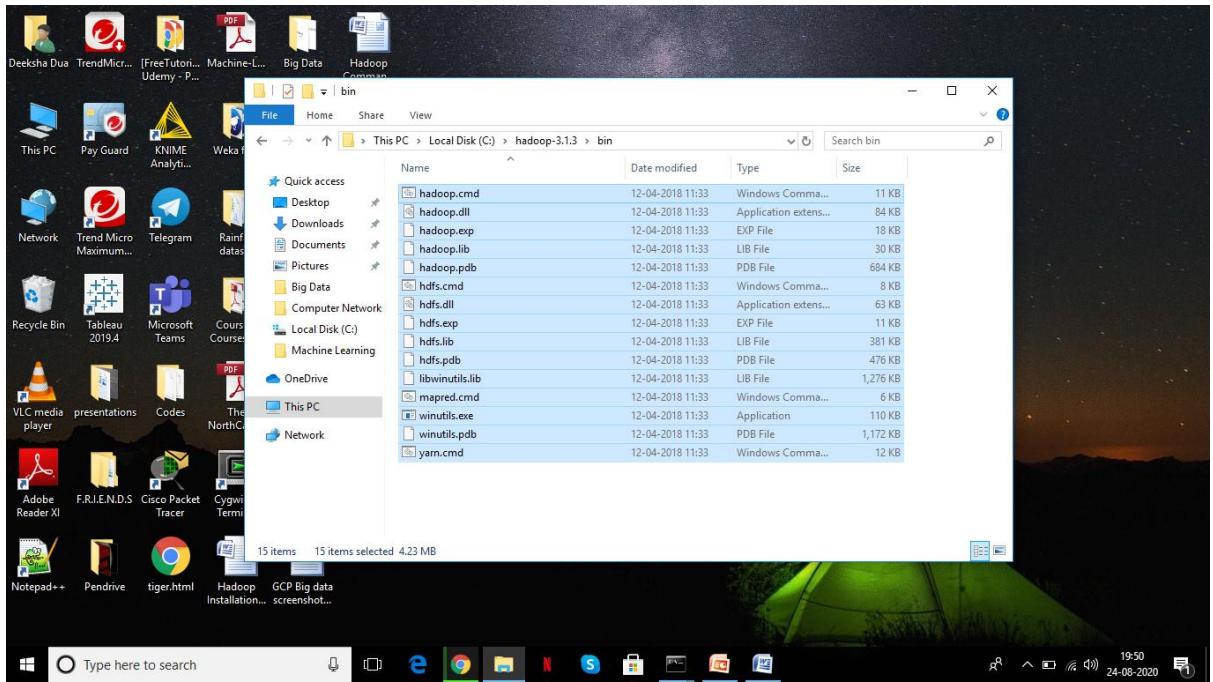


Fig15

Step – 6 : Open cmd and run the following commands to know Hadoop is properly installed or not.

- `$hadoop version`

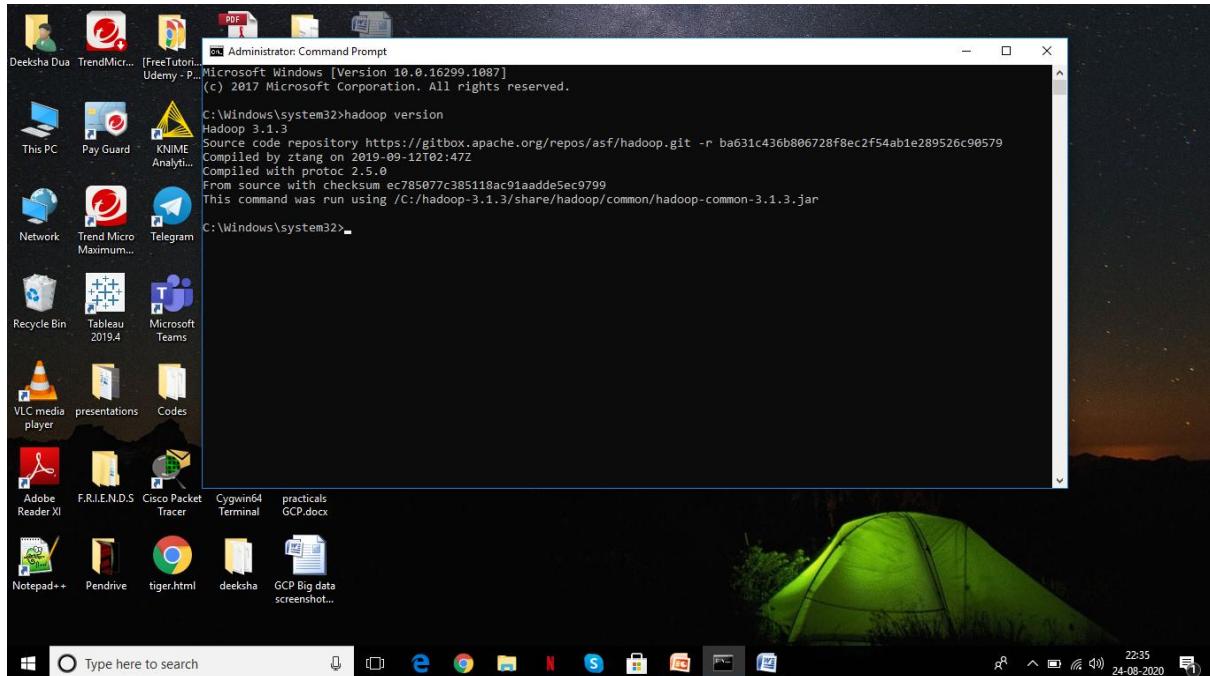


Fig16

Format the namenode

- \$hdfs namenode -format

```
17/07/20 15:38:21 INFO util.GSet: capacity      = 2^18 = 262144 entries
17/07/20 15:38:21 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.9990000128746033
17/07/20 15:38:21 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
17/07/20 15:38:21 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension     = 30000
17/07/20 15:38:21 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
17/07/20 15:38:21 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
17/07/20 15:38:21 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
17/07/20 15:38:21 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
17/07/20 15:38:21 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time
is 600000 millis
17/07/20 15:38:21 INFO util.GSet: Computing capacity for map NameNodeRetryCache
17/07/20 15:38:21 INFO util.GSet: VM type       = 64-bit
17/07/20 15:38:21 INFO util.GSet: 0.029999999329447746% max memory 889 MB = 273.1 KB
17/07/20 15:38:21 INFO util.GSet: capacity      = 2^15 = 32768 entries
17/07/20 15:38:26 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1080504939-192.168.68.1-1500547106632
17/07/20 15:38:26 INFO common.Storage: Storage directory C:\Hadoop-2.8.0\data\namenode has been successfully formatted.
17/07/20 15:38:26 INFO namenode.FSImageFormatProtobuf: Saving image file C:\Hadoop-2.8.0\data\namenode\current\fsimage.c
kpt_00000000000000000000000000000000 using no compression
17/07/20 15:38:26 INFO namenode.FSImageFormatProtobuf: Image file C:\Hadoop-2.8.0\data\namenode\current\fsimage.ckpt_000
0000000000000000 of size 330 bytes saved in 0 seconds.
17/07/20 15:38:26 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
17/07/20 15:38:26 INFO util.ExitUtil: Exiting with status 0
17/07/20 15:38:26 INFO namenode.NameNode: SHUTDOWN_MSG:
*****SHUTDOWN_MSG: Shutting down NameNode at [REDACTED]
*****
```

Fig17

Step – 7 : Change directory to ‘C:\hadoop-3.1.3\sbin’ and type ‘start-all.cmd’ to start the services of hadoop.

- \$start-all.cmd

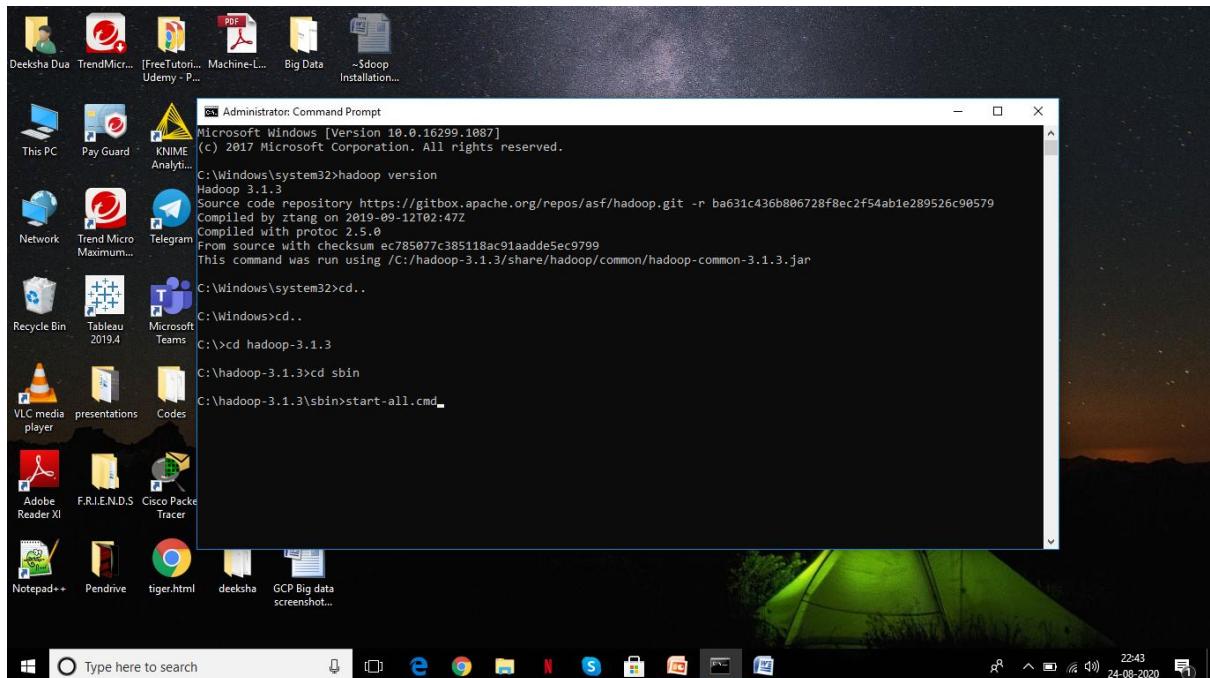


Fig18

After writing the commands 4 new screens pop up as follows :

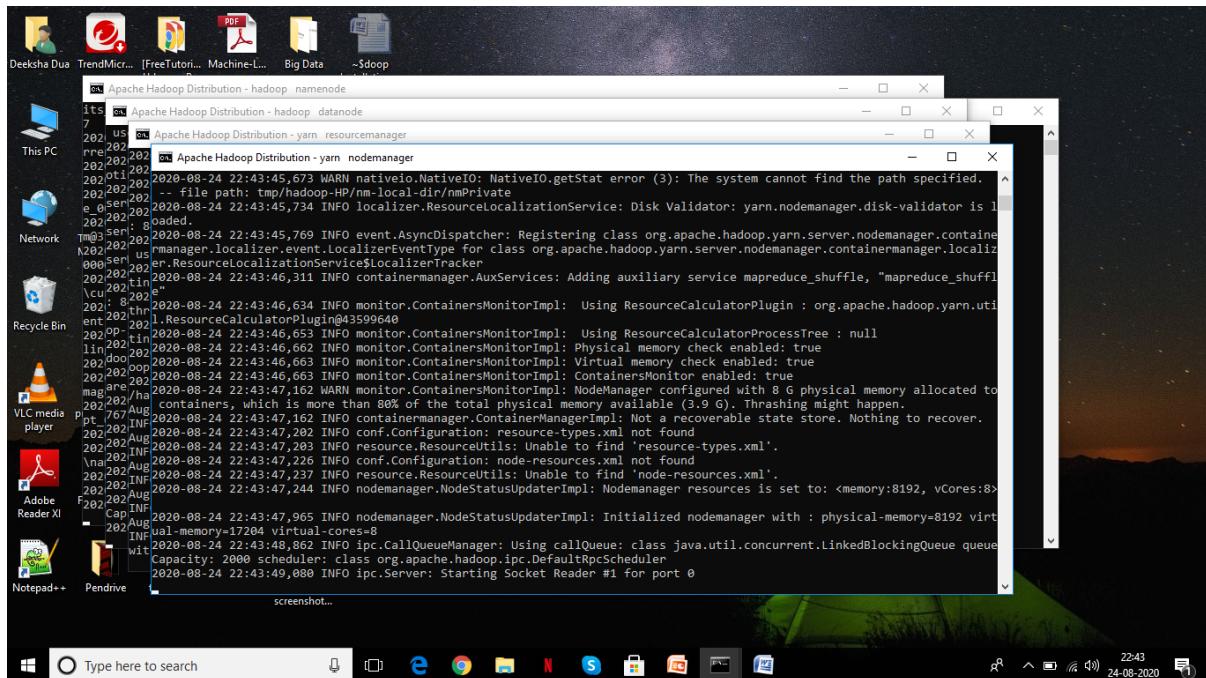


Fig19

Step – 8 : To access information about resource manager current jobs, successful and failed jobs, go to this link in browser-<http://localhost:8088/cluster>

To check the details about the hdfs (namenode and datanode):<http://localhost:9870/>

Step – 9 : As the installation is done, now explore more by running various commands.

Lab – 5

Student Name and Roll Number: Chayan 18csu054

Semester /Section: 5/A

Link to Code: <https://github.com/chayangulati321/Big-Data>

Date: 19/08/20

Faculty Signature:

Remarks:

Hadoop Commands

AIM : To write basic commands for Hadoop.

- First of all check hadoop version with command ‘hadoop version’
- Formate the namenode with the following syntax:

Syntax: \$hdfs namenode -format

Fig1

- Change directory to hadoop\sbin
 - Start all services like namenode, datanode, resource manager and node manager

1. Create directory

- `mkdir` – To create directory in HDFS

- \$hdfs dfs -mkdir [-option] URI

2. Put file in directory

- put – To copy files from local system to HDFS

- \$hdfs dfs -put [-option] <localsrc> URI

Options: -p : Preserves access and modification times, ownership and the permissions. (assuming the permissions can be propagated across filesystems)

-f: Overwrites the destination if it already exists.

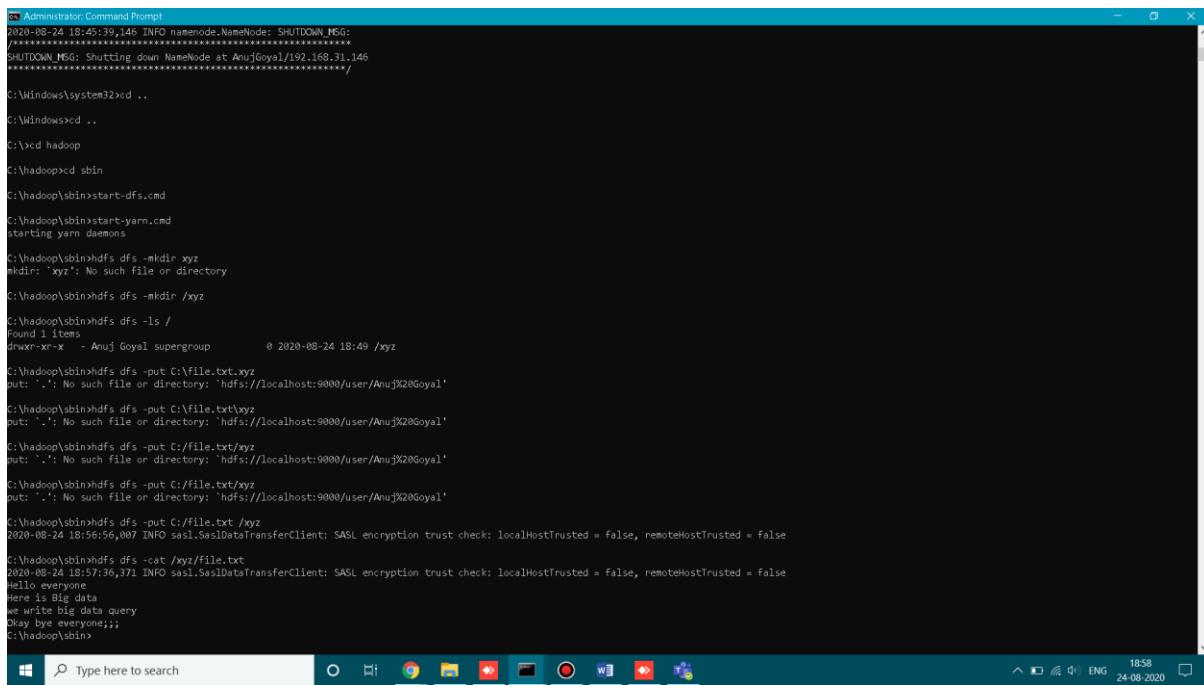
-l: Allow datanode to lazily persist the file to disk, Forces a replication factor of 1. This flag will result in reduced durability. Use with care.

-d: Skip creation of temporary file with the suffix._COPYING_.

3. Read the file

- cat – To read the content of file.

- \$hdfs dfs -cat [-option] URI



```

Administrator: Command Prompt
2020-08-24 18:45:39,146 INFO namenode.NameNode: SHUTDOWN_MSG:
*****SHUTDOWN MSG: Shutting down NameNode at AnujGoyal/192.168.31.146*****
*****SHUTDOWN_MSG: Shutting down NameNode at AnujGoyal/192.168.31.146*****/c:\Windows\system32>c d ..
c:\Windows>c d ..
c:\>cd hadoop
c:\hadoop>cd sbin
c:\hadoop\sbin>start-dfs.cmd
c:\hadoop\sbin>start-yarn.cmd
starting yarn daemons
c:\hadoop\sbin>hdfs dfs -mkdir xyz
mkdir: `xyz': No such file or directory
c:\hadoop\sbin>hdfs dfs -mkdir /xyz
c:\hadoop\sbin>hdfs dfs -ls /
Found 1 items
drwxr-xr-x - Anuj Goyal supergroup          0 2020-08-24 18:49 /xyz
c:\hadoop\sbin>hdfs dfs -put C:\file.txt xyz
put: `.' No such file or directory: `hdfs://localhost:9000/user/Anuj%20Goyal'
c:\hadoop\sbin>hdfs dfs -put C:\file.txt\yz
put: `.' No such file or directory: `hdfs://localhost:9000/user/Anuj%20Goyal'
c:\hadoop\sbin>hdfs dfs -put C:/file.txt\yz
put: `.' No such file or directory: `hdfs://localhost:9000/user/Anuj%20Goyal'
c:\hadoop\sbin>hdfs dfs -put C:/file.txt\yz
put: `.' No such file or directory: `hdfs://localhost:9000/user/Anuj%20Goyal'
c:\hadoop\sbin>hdfs dfs -put C:\file.txt /yz
2020-08-24 18:56:56,007 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
c:\hadoop\sbin>hdfs dfs -cat /xyz/file.txt
2020-08-24 18:57:36,371 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
Hello everyone
here is Big data
we write big data query
play by everyone;;
c:\hadoop\sbin>

```

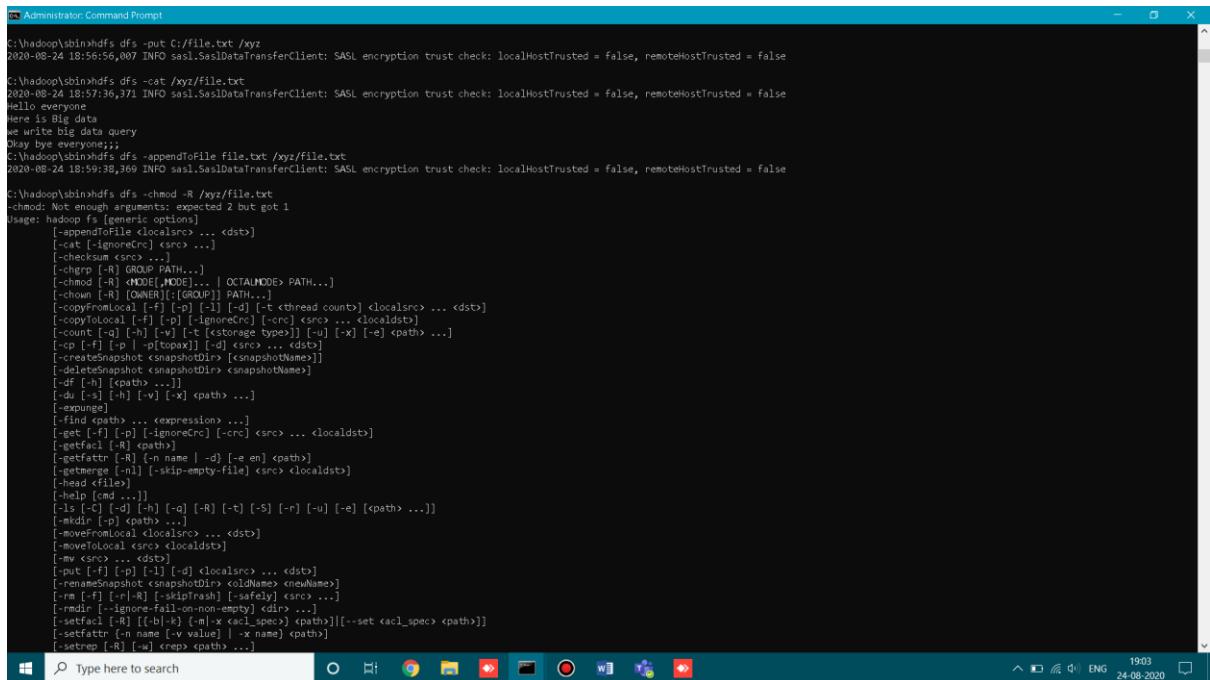
Fig2

4. Append file to directory

- `appendToFile` – To append file from local file system to destination file.
- `$hdfs dfs -appendToFile <srcf><dest>`

5. Change the access mode

- `chmod` – To change the permission of file. (either user or owner or group)
- `$hdfs dfs -chmod [-option] <MODE[,MODE]... | OCTALMODE> URI`
- Options: `-R` : This option will make the change recursively.



The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt". The command entered is:

```
C:\hadoop\bin\nmfs dfs -put C:/file.txt /xyz  
2020-08-24 18:56:56,007 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false  
C:\hadoop\bin\nmfs dfs -cat /xyz/file.txt  
2020-08-24 18:57:36,371 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false  
Hello everyone  
Here is Big data  
we write big data query  
Okay bye everyone;;;  
C:\hadoop\bin\nmfs dfs -appendToFile file.txt /xyz/file.txt  
2020-08-24 18:59:38,369 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
```

The command prompt also displays the usage information for the HDFS dfs command, which includes options for putting files, getting files, and managing snapshots.

Fig3

6. List the files in directory

- **ls** - To display list of files or directories

- \$hdfs dfs -ls [-options] URI

7. Copy a file from local device

- copyFromLocal – To copy files from local device to HDFS
- \$hdfs dfs -copyFromLocal <localsrc> URI

8. Copy a file to local device

- copyToLocal – To copy files from HDFS to local device
- \$hdfs dfs -copyToLocal URI <localdst>

9. Count

- count - To count the no. of files, directories and bytes under the path that match the specified file pattern
- \$hdfs dfs -count [option] path
- Options:
 - -u and -q : These options control what columns the output contains. -q means show quotas, -u limits the output to show quotas and usage only.
 - -h : This option shows sizes in human readable format.
 - -v : This option displays a header line.
 - -x : This option excludes snapshots from the result calculation. Without the -x option, the result is always calculated from all INodes, including all snapshots under the given path. The -x option is ignored if -u or -q option is given.
 - -e : This option shows the erasure coding policy for each file.
 - -q

```
C:\hadoop-3.1.3\sbin>hadoop fs -count -q /
9223372036854775807 9223372036854775782          none          inf         12         13       1013504 /
```

- -q -h

```
C:\hadoop-3.1.3\sbin>hadoop fs -count -q -h /
 8.0 E      8.0 E      none          inf         12         13      989.8 K /
```

```
C:\hadoop-3.1.3\sbin>hadoop fs -count -q -h -v /
 QUOTA      REM_QUOTA      SPACE_QUOTA REM_SPACE_QUOTA   DIR_COUNT   FILE_COUNT    CONTENT_SIZE PATHNAME
 8.0 E      8.0 E      none          inf         12         13      989.8 K /
```

-u

```
C:\hadoop-3.1.3\sbin>hadoop fs -count -u /
9223372036854775807 9223372036854775782          none          inf /
```

-u -h

```
C:\hadoop-3.1.3\sbin>hadoop fs -count -u -h /
 8.0 E      8.0 E      none          inf /
```

- -u -h -v

```
C:\hadoop-3.1.3\sbin>hadoop fs -count -u -h -v /
    QUOTA      REM_QUOTA      SPACE_QUOTA REM_SPACE_QUOTA PATHNAME
    8.0 E        8.0 E          none           inf /
```

- -e

```
C:\hadoop-3.1.3\sbin>hadoop fs -count -e /
    12            13          1013504 EC: /
```

- -v

```
C:\hadoop-3.1.3\sbin>hadoop fs -count -v /
    DIR_COUNT   FILE_COUNT   CONTENT_SIZE PATHNAME
    12           13           1013504 /
```

- -x

```
C:\hadoop-3.1.3\sbin>hadoop fs -count -x /
    12            13          1013504 /
```

10. Copy a file from one place to another

- cp – To copy file from one place to another
- \$hdfs dfs -cp [-option] URI <dst>
- Options:
- -f : It will overwrite the destination if it already exists.
- -p : It will preserve file attributes [topx] (timestamps, ownership, permission, ACL, XAttr). If -p is specified with no *arg*, then preserves timestamps, ownership, permission.

```
C:\hadoop-3.1.3\sbin>hadoop fs -cp /Newd/ /sample/
2020-08-24 19:53:19,186 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2020-08-24 19:53:19,388 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
```

11. To check file or directory information free space

- df – To display the free space available and used space of a file or directory in HDFS
- \$hdfs dfs -df [-option] URI

12. Display size of file or directory

- du – To display the size of file or directory
- \$hdfs dfs -du [-option] URI
- Options:
- -s : This option will result in an aggregate summary of file lengths being displayed, rather than the individual files. Without the -s option, calculation is done by going 1-level deep from the given path.

- **-h** : This option will format file sizes in a “human-readable” fashion (e.g 64.0m instead of 67108864)
- **-v** : This option will display the names of columns as a header line.
- **-x** : This option will exclude snapshots from the result calculation. Without the **-x** option (default), the result is always calculated from all INodes, including all snapshots under the given path.

13. To find a particular file use ‘-find’

- **find** – To find all files that matches the expression
- **\$hdfs dfs -find <path> expression**

```

Administrator: Command Prompt
[truncate [-w] <length> <path> ...]
[-usage [cmd ...]]

Generic options supported are:
-conf <configuration file>           specify an application configuration file
-D <property=value>                  define a value for a given property
-fs <file:///hdfs://namenode:port>    specify default filesystem URL to use, overrides 'fs.defaultFS' property from configurations.
-jr <localresourcemanager:port>       specify a ResourceManager
-fs <file1,...>                     specify a comma-separated list of files to be copied to the map reduce cluster
-jar <jar1,...>                      specify a comma-separated list of jar files to be included in the classpath
-archives <archive1,...>             specify a comma-separated list of archives to be unarchived on the compute machines

The general command line syntax is:
command [genericOptions] [commandOptions]

Usage: hadoop fs [generic options] -chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...
C:\hadoop\bin>hdfs dfs -copyFromLocal C:\file1.txt /xyz
2020-08-24 19:08:53,458 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
C:\hadoop\bin>hdfs dfs -ls /xyz
Found 2 items
-rw-r--r-- 1 Anuj Goyal supergroup 112 2020-08-24 18:59 /xyz/file.txt
-rw-r--r-- 1 Anuj Goyal supergroup 77 2020-08-24 19:06 /xyz/file1.txt
C:\hadoop\bin>hdfs dfs -count -q /xyz/file.txt
none          inf        none      inf     0      1      112 /xyz/file.txt
C:\hadoop\bin>hdfs dfs -cp /xyz/file.txt /xyz/file2.txt
2020-08-24 19:08:31,768 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2020-08-24 19:08:32,014 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
C:\hadoop\bin>hdfs dfs -df /xyz
Filesystem          Size  Used  Available  Use%
hdfs://localhost:9000 108237156352  657 31111180288   0%
C:\hadoop\bin>hdfs dfs -df /xyz/file.txt
Filesystem          Size  Used  Available  Use%
hdfs://localhost:9000 108237156352  657 31109279744   0%
C:\hadoop\bin>hdfs dfs -du /xyz/file.txt
112 112 /xyz/file.txt
C:\hadoop\bin>hdfs dfs -find /xyz/file.
/
/xyz/
/xyz/file.txt
/xyz/file1.txt
/xyz/file2.txt
Find: "xyz/file": No such file or directory

```

Fig4

14. Get the file

- get – To get the files from local file system
- \$hdfs dfs -get [-options] <src> <dst>
- Options:
 - -p : Preserves access and modification times, ownership and the permissions.
 - -f : Overwrites the destination if it already exists.
 - -ignorecrc : Skip CRC checks on the file downloaded.
 - -crc: write CRC checksums for the files downloaded.

```
C:\hadoop-3.1.3\sbin>hadoop fs -get /input_dir7/input /C:/hadoop-3.1.3/sbin/
2020-08-24 22:14:29,087 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false

C:\hadoop-3.1.3\sbin>hadoop fs -ls /
Found 5 items
-rw-r--r-- 1 HP hadoopgroup 1888 2020-08-24 18:34 /Newd
drwxr-xr-x - HP supergroup 0 2020-08-24 19:26 /input_dir7
drwxr-xr-x - HP supergroup 0 2020-08-24 19:53 /sample
drwx----- - HP supergroup 0 2020-08-24 10:19 /tmp
dwx---x--- - HDFS supergroup 0 2020-08-20 12:12 /user

C:\hadoop-3.1.3\sbin>
```

15. Head to check the content of file

- To display the specified no. of lines from beginning of the file in HDFS
- \$hdfs dfs -head URI

16. Help command for any query or help

- help – To display the no. of operations which can be performed
- \$hdfs dfs -help

```

Administrator: Command Prompt
/xyz/file2.txt
find: `xyz/file': No such file or directory
C:\Windows\bin\hadoop\bin>dfs -help
2020-08-24 19:11:00,413 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
Hello everyone
here is Big data
we write big data query
Okay bye everyone;;;hello
okay

Hello everyone
C:\Windows\bin\hadoop\bin>dfs -help
Usage: hadoop fs [generic options]
  [-appendToFile <localsrc> ... <dst>]
  [-cat [-ignorecrc] <src> ...]
  [-checksum <src> ...]
  [-chgrp [-R] <NODE[,MODE]... | OCTALMODE> PATH...]
  [-chmod [-R] <OWNER|[,GROUP] PATH...]
  [-chown [-R] [OWNER][,[GROUP]] PATH...]
  [-copyFromLocal [-f] [-p] [-i] [-d] [-t <thread count>] <localsrc> ... <dst>]
  [-copyToLocal [-f] [-p] [-i] [-d] [-t <thread count>] <src> ... <localdst>]
  [-cp [-R] [-f] [-p] [-i] [-d] [-t <archive type>] [-x] [-e] <path> ...]
  [-cp [-f] [-p] [-i] [-d] <src> <dst>]
  [-createSnapshot <snapshotDir> <snapshotName>]
  [-deleteSnapshot <snapshotDir> <snapshotName>]
  [-df [-n] <path> ...]
  [-du [-s] [-h] [-v] [-x] <path> ...]
  [-expunge]
  [-find <path> ... <expression> ...]
  [-get [-f] [-p] [-i] [-ignorecrc] <src> ... <localdst>]
  [-getfacl [-f] <path> ...]
  [-getfattr [-R] [-n name | -d] [-e en] <path>]
  [-getmerge [-n] [-skip-empty-file] <src> <localdst>]
  [-head <file>]
  [-help [<cmd>...]]
  [-ls [-t] [-d] [-h] [-q] [-R] [-t] [-S] [-r] [-u] [-e] <path> ...]
  [-mkdir [-p] <path> ...]
  [-moveFromLocal <localsrc> ... <dst>]
  [-moveToLocal <src> <localdst>]
  [-put [-f] [-p] [-i] [-d] <localsrc> ... <dst>]
  [-renameSnapshot <snapshotDir> <oldName> <newName>]
  [-rm [-f] [-r|-R] [-skiptrash] [-safely] <src> ...]
  [-rmdir [-ignore-fail-on-non-empty] <dir> ...]
  [-setfacl [-R] [[-b|-k] (#|> <acl_spec>) <path>]|[--set <acl_spec> <path>]]
  [-setfattr [-n name [-v value] | -x name] <path>]
  [-setrep [-R] [-w] <rep> <path> ...]
  [-stat [<format>] <path> ...]
  [-tail [-f] [-s <sleep interval>] <file>]

```

Fig5

17. To display file information

- **ls** – To display the content of files, list of files and directories
- **\$hdfs dfs -ls [-options] URI**
- Options:
 - **-C**: Display the paths of files and directories only.
 - **-d**: Directories are listed as plain files.
 - **-h**: Format file sizes in a human-readable fashion (eg 64.0m instead of 67108864).
 - **-q**: Print ? instead of non-printable characters.
 - **-R**: Recursively list subdirectories encountered.
 - **-t**: Sort output by modification time (most recent first).
 - **-S**: Sort output by file size.
 - **-r**: Reverse the sort order.
 - **-u**: Use access time rather than modification time for display and sorting.
 - **-e**: Display the erasure coding policy of files and directories only

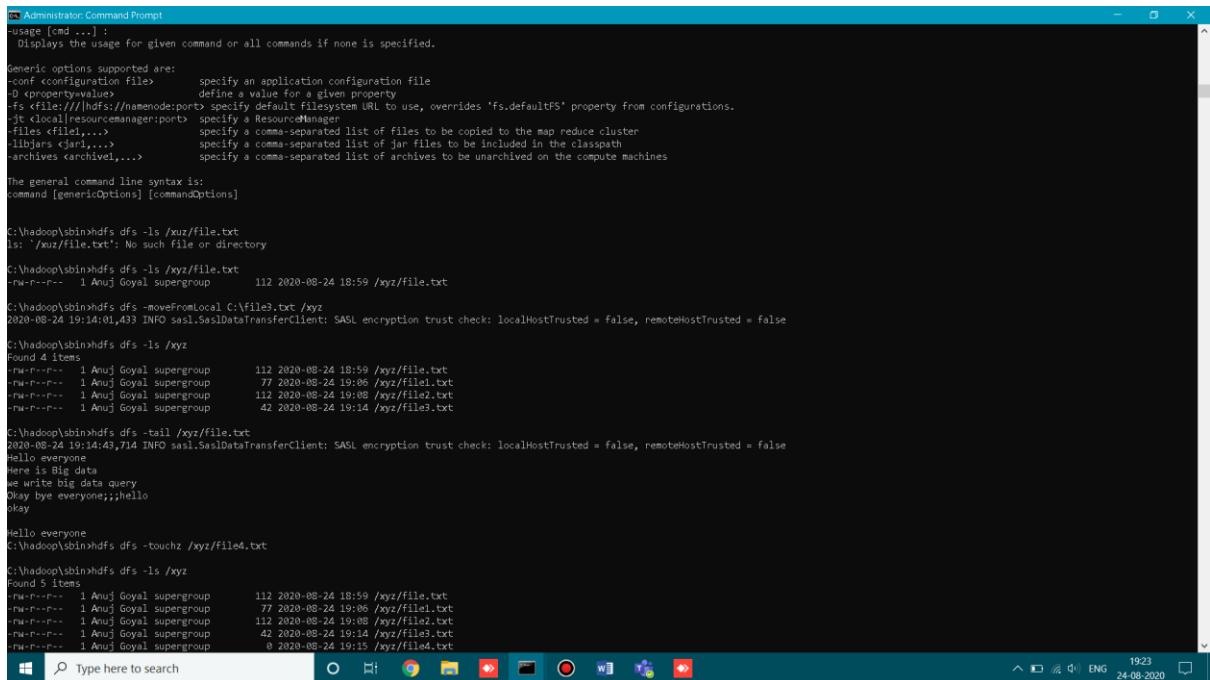
18. Tail to check file content

- To display the specified no. of lines from ending of the file in HDFS
- **\$hdfs dfs -tail URI**

19. Touchz to create a file with zero size

- **touchz** – To create a file of zero length

- **\$hdfs dfs -touchz URI**



```

Administrator: Command Prompt
Usage [cmd ...]:
  Displays the usage for given command or all commands if none is specified.

Generic options supported are:
-conf <configuration file>      specify an application configuration file
-D <property>=<value>          define a value for a given property
-fs <file:///hdfs://namenode:port> specify default filesystem URL to use, overrides 'fs.defaultFS' property from configurations.
-jt <local|resourcemanager>:port specify a ResourceManager port
-files <file1,...>             specify a comma-separated list of files to be copied to the map reduce cluster
-libjars <jar1,...>             specify a comma-separated list of jar files to be included in the classpath
-archives <archive1,...>        specify a comma-separated list of archives to be unarchived on the compute machines

The general command line syntax is:
command [genericOptions] [commandOptions]

C:\hadoop\bin>hdfs dfs -ls /xyz/file.txt
ls: '/xyz/file.txt': No such file or directory

C:\hadoop\bin>hdfs dfs -ls /xyz/file.txt
-rw-r--r-- 1 Anuj Goyal supergroup 112 2020-08-24 18:59 /xyz/file.txt

C:\hadoop\bin>hdfs dfs -moveFromLocal C:\file3.txt /xyz
2020-08-24 19:14:01,439 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false

C:\hadoop\bin>hdfs dfs -ls /xyz
Found 4 items
-rw-r--r-- 1 Anuj Goyal supergroup 112 2020-08-24 18:59 /xyz/file.txt
-rw-r--r-- 1 Anuj Goyal supergroup 77 2020-08-24 19:05 /xyz/file1.txt
-rw-r--r-- 1 Anuj Goyal supergroup 112 2020-08-24 19:08 /xyz/file2.txt
-rw-r--r-- 1 Anuj Goyal supergroup 42 2020-08-24 19:14 /xyz/file3.txt

C:\hadoop\bin>hdfs dfs -tail /xyz/file.txt
2020-08-24 19:14:43,714 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
Hello everyone
Hello everyone
Hello is Big data
we write big data query
Okay bye everyone;;hello
okay

Hello everyone
C:\hadoop\bin>hdfs dfs -touchz /xyz/file4.txt

C:\hadoop\bin>hdfs dfs -ls /xyz
Found 5 items
-rw-r--r-- 1 Anuj Goyal supergroup 112 2020-08-24 18:59 /xyz/file.txt
-rw-r--r-- 1 Anuj Goyal supergroup 77 2020-08-24 19:05 /xyz/file1.txt
-rw-r--r-- 1 Anuj Goyal supergroup 112 2020-08-24 19:08 /xyz/file2.txt
-rw-r--r-- 1 Anuj Goyal supergroup 42 2020-08-24 19:14 /xyz/file3.txt
-rw-r--r-- 1 Anuj Goyal supergroup 0 2020-08-24 19:15 /xyz/file4.txt

```

Fig6

20. Remove file from directory

- **rm – To remove or delete file**
- **\$hdfs dfs -rm URI**

21. Remove directory

- **rmdir – To remove the directory**
- **\$hdfs dfs -rmdir URI**

```
C:\Administrator>Administrator: Command Prompt
C:\hadoop\sbin>hdfs dfs -ls /xyz
Found 4 items
-rw-r--r-- 1 Anuj Goyal supergroup 112 2020-08-24 18:59 /xyz/file.txt
-rw-r--r-- 1 Anuj Goyal supergroup 77 2020-08-24 19:06 /xyz/file1.txt
-rw-r--r-- 1 Anuj Goyal supergroup 112 2020-08-24 19:08 /xyz/file2.txt
-rw-r--r-- 1 Anuj Goyal supergroup 42 2020-08-24 19:14 /xyz/file3.txt

C:\hadoop\sbin>hdfs dfs -tail /xyz/file.txt
2020-08-24 19:14:43,714 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
Hello everyone
Here is Big data
we write big data query
Okay bye everyone;;;hello
okay

Hello everyone
C:\hadoop\sbin>hdfs dfs -touchz /xyz/file4.txt

C:\hadoop\sbin>hdfs dfs -ls /xyz
Found 5 items
-rw-r--r-- 1 Anuj Goyal supergroup 112 2020-08-24 18:59 /xyz/file.txt
-rw-r--r-- 1 Anuj Goyal supergroup 77 2020-08-24 19:06 /xyz/file1.txt
-rw-r--r-- 1 Anuj Goyal supergroup 112 2020-08-24 19:08 /xyz/file2.txt
-rw-r--r-- 1 Anuj Goyal supergroup 42 2020-08-24 19:14 /xyz/file3.txt
-rw-r--r-- 1 Anuj Goyal supergroup 0 2020-08-24 19:15 /xyz/file4.txt

C:\hadoop\sbin>hdfs dfs -rm /xyz/file4.txt
Deleted /xyz/file4.txt

C:\hadoop\sbin>hdfs dfs -ls /xyz
Found 4 items
-rw-r--r-- 1 Anuj Goyal supergroup 112 2020-08-24 18:59 /xyz/file.txt
-rw-r--r-- 1 Anuj Goyal supergroup 77 2020-08-24 19:06 /xyz/file1.txt
-rw-r--r-- 1 Anuj Goyal supergroup 112 2020-08-24 19:08 /xyz/file2.txt
-rw-r--r-- 1 Anuj Goyal supergroup 42 2020-08-24 19:14 /xyz/file3.txt

C:\hadoop\sbin>hdfs dfs -rmdir /xyz
rmdir: '/xyz': Directory is not empty

C:\hadoop\sbin>hdfs dfs -rm /xyz/file3.txt /xyz/file2.txt /xyz/file1.txt
Deleted /xyz/file3.txt
Deleted /xyz/file2.txt
Deleted /xyz/file1.txt
Deleted /xyz/file.txt

C:\hadoop\sbin>hdfs dfs -ls /xyz
C:\hadoop\sbin>hdfs dfs -rmdir /xyz
C:\hadoop\sbin>
```

Fig7

Lab – 6

Student Name and Roll Number: Chayan 18csu054

Semester /Section: 5/A

Link to Code: <https://github.com/chayangulati321/Big-Data>

Date: 20/08/20

Faculty Signature:

Remarks:

Word Count on MapReduce

AIM : MapReduce on a file to count the word appearing in it.

Step – 1 : Open Command Prompt as Administrator and run the following commands:

- \$start-dfs.cmd
- \$start-yarn.cmd
- \$jps

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.18362.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\WINDOWS\system32>jps
18244 Jps
19476 DataNode
23384 NameNode
19692 ResourceManager
21388 NodeManager

C:\WINDOWS\system32>
```

Fig1

Step – 2 : Copy input_file.txt to input_dir7 and check the list of files using -ls

```
C:\WINDOWS\system32>hadoop fs -ls \input_dir7
Found 6 items
-rw-r--r-- 1 HP supergroup          1888 2020-08-24 19:26 /input_dir7/input
-rw-r--r-- 1 HP supergroup          1888 2020-08-24 19:06 /input_dir7/input_copy.txt
-rw-r--r-- 1 HP supergroup          1888 2020-08-24 10:17 /input_dir7/input_file.txt
drwxr-xr-x - HP supergroup          0 2020-08-24 19:51 /input_dir7/move
-rw-r--r-- 1 HP supergroup          1888 2020-08-24 22:35 /input_dir7/move_loc
drwxr-xr-x - HP supergroup          0 2020-08-24 22:44 /input_dir7/put
```

Fig2

Step – 3 : To see the content of file

- \$hadoop dfs -cat /input_dir7/input_file.txt

```
C:\WINDOWS\system32>hadoop dfs -cat /input_dir7/input_file.txt
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
2020-08-24 23:12:33,994 INFO SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
23 23 27 43 24 25 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
C:\WINDOWS\system32>
```

Fig3

Step – 4 : Run the MapReduceClient.jar wordcount using hadoop jar and save it to output directory

- \$hadoop jar C:/MapReduceClient.jar wordcount /input_dir7 /output_dir

```
C:\WINDOWS\system32>hadoop jar C:/MapReduceClient.jar wordcount /input_dir7 /output_dir
2020-08-24 23:27:13,801 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032

FILE: Number of write operations=0
HDFS: Number of bytes read=1999
HDFS: Number of bytes written=120
HDFS: Number of read operations=6
HDFS: Number of large read operations=0
HDFS: Number of write operations=2
Job Counters
Launched map tasks=1
Launched reduce tasks=1
Data-local map tasks=1
Total time spent by all maps in occupied slots (ms)=2180
Total time spent by all reduces in occupied slots (ms)=2442
Total time spent by all map tasks (ms)=2180
Total time spent by all reduce tasks (ms)=2442
Total vcore-milliseconds taken by all map tasks=2180
Total vcore-milliseconds taken by all reduce tasks=2442
Total megabyte-milliseconds taken by all map tasks=2232320
Total megabyte-milliseconds taken by all reduce tasks=2500608
Map-Reduce Framework
Map input records=30
Map output records=390
Map output bytes=2730
Map output materialized bytes=195
Input split bytes=111
Combine input records=390
Combine output records=21
Combine output bytes=111
```

Fig4

Step – 5 : To display the output of generated file

- \$hadoop dfs -cat/output_dir/*

Lab – 7

Student Name and Roll Number: Chayan 18csu054

Semester /Section: 5/A

Link to Code: <https://github.com/chayangulati321/Big-Data>

Date: 24/08/20

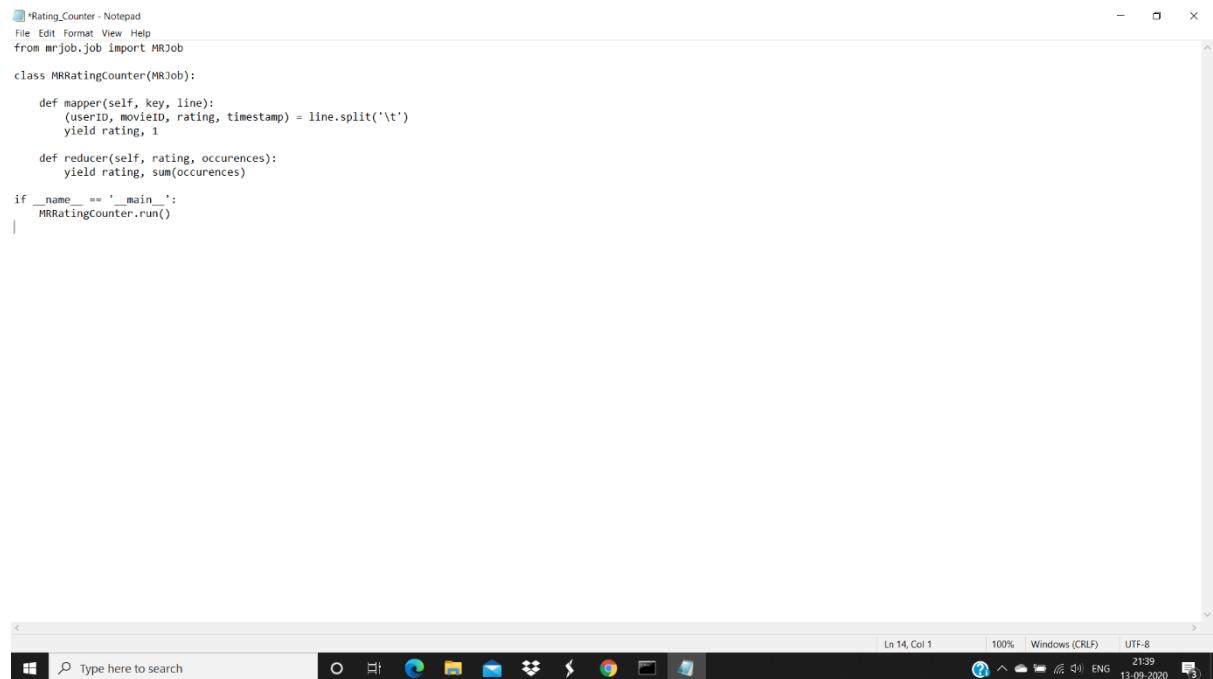
Faculty Signature:

Remarks:

Aim : One Mapper and One Reducer on the local machine

- Find the occurrence of rating

MRJob



```
*Rating_Counter - Notepad
File Edit Format View Help
from mrjob.job import MRJob

class MRRatingCounter(MRJob):

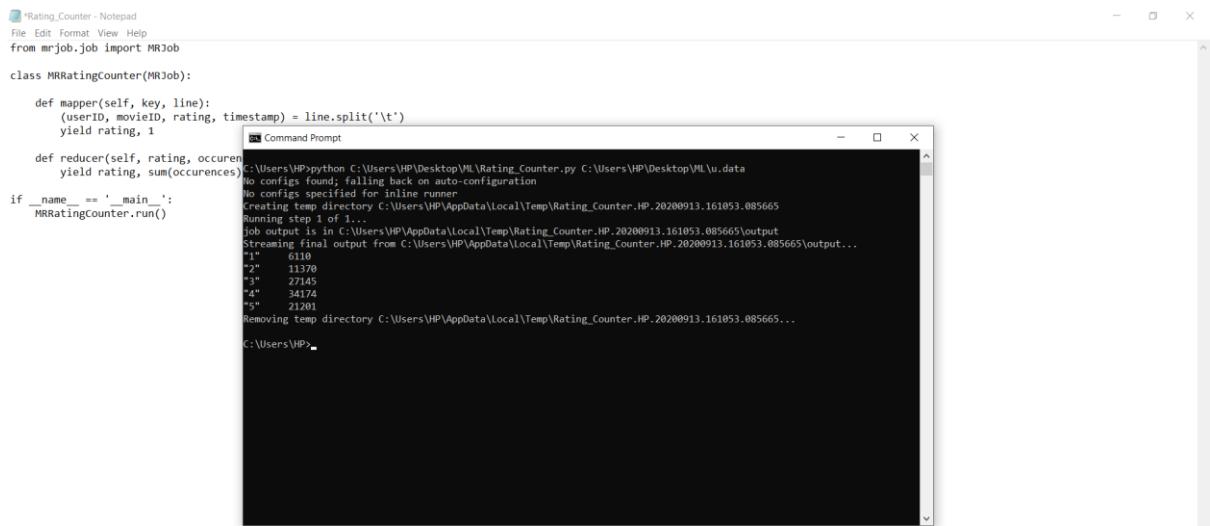
    def mapper(self, key, line):
        (userID, movieID, rating, timestamp) = line.split('\t')
        yield rating, 1

    def reducer(self, rating, occurrences):
        yield rating, sum(occurrences)

if __name__ == '__main__':
    MRRatingCounter.run()
```

Fig1

- To run MRJob run the following commands:-
\$python <path_of_script> <path_of_file>



The screenshot shows a Windows desktop environment. In the foreground, there is a Command Prompt window titled "Command Prompt". The command entered is "\$python C:\Users\HP\Desktop\ML\Rating_Counter.py C:\Users\HP\Desktop\ML\u.data". The output of the command is displayed below the command line, showing the execution of the MRJob script and the resulting rating counts.

```

Rating_Counter - Notepad
File Edit Format View Help
from mrjob.job import MRJob

class MRRatingCounter(MRJob):

    def mapper(self, key, line):
        (userId, movieId, rating, timestamp) = line.split('\t')
        yield rating, 1

    def reducer(self, rating, occurences):
        yield rating, sum(occurences)

if __name__ == '__main__':
    MRRatingCounter.run()

```

```

C:\Users\HP>python C:\Users\HP\Desktop\ML\Rating_Counter.py C:\Users\HP\Desktop\ML\u.data
No configs specified for inline runner
Creating temp directory C:\Users\HP\AppData\Local\Temp\Rating_Counter.HP.20200913.161053.085665
Running step 1 of 1...
job output is in C:\Users\HP\AppData\Local\Temp\Rating_Counter.HP.20200913.161053.085665\output
Streaming final output from C:\Users\HP\AppData\Local\Temp\Rating_Counter.HP.20200913.161053.085665\output...
"1"      6110
"2"      1509
"3"      27145
"4"      34174
"5"      21201
Removing temp directory C:\Users\HP\AppData\Local\Temp\Rating_Counter.HP.20200913.161053.085665...
C:\Users\HP>

```



Fig2

Lab – 8

Student Name and Roll Number: Chayan 18csu054

Semester /Section: 5/A

Link to Code: <https://github.com/chayangulati321/Big-Data>

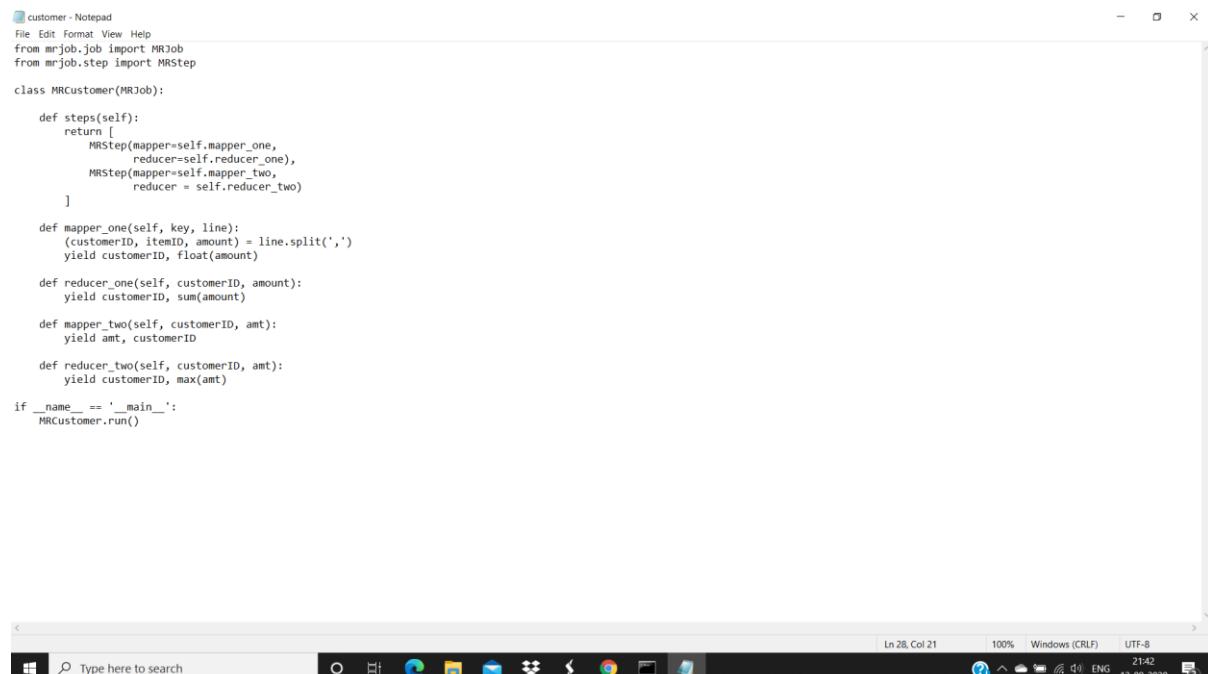
Date: 24/08/20

Faculty Signature:

Remarks:

Aim : Two Mapper and Two Reducer on the local machine

- Find the maximum amount paid by the customer



```
customer - Notepad
File Edit Format View Help
from mrjob.job import MRJob
from mrjob.step import MRStep

class MRCustomer(MRJob):
    def steps(self):
        return [
            MRStep(mapper=self.mapper_one,
                   reducer=self.reducer_one),
            MRStep(mapper=self.mapper_two,
                   reducer = self.reducer_two)
        ]
    def mapper_one(self, key, line):
        (customerID, itemID, amount) = line.split(',')
        yield customerID, float(amount)

    def reducer_one(self, customerID, amounts):
        yield customerID, sum(amounts)

    def mapper_two(self, customerID, amt):
        yield customerID, amt

    def reducer_two(self, customerID, amts):
        yield customerID, max(amts)

if __name__ == '__main__':
    MRCustomer.run()
```

Fig3

- To run MRJob run the following commands:-
\$python <path_of_script> <path_of_file>

The figure shows a Windows desktop environment with two windows open:

- Notepad Window:** The title bar says "customer - Notepad". The content is a Python script named "customer.py" which defines a class "MRCustomer" that extends "MRJob". It includes methods for defining steps, mapping, and reducing data from a CSV file. The code uses the mrjob library.
- Command Prompt Window:** The title bar says "Command Prompt". The command run is "python C:\Users\HP\Desktop\ML\customer.py C:\Users\HP\Desktop\ML\customer-orders.csv". The output shows the job configuration and the start of the job execution process.

```

customer - Notepad
File Edit Format View Help
from mrjob.job import MRJob
from mrjob.step import MRStep

class MRCustomer(MRJob):
    def steps(self):
        return [
            MRStep(mapper=self.mapper,
                   reducer=self.reducer),
            MRStep(mapper=self.mapper,
                   reducer = self.reducer)
        ]
    def mapper_one(self, key, line):
        (customerId, itemId, amount)
        yield customerId, float(amount)
    def mapper_two(self, customerId, itemId, amount):
        yield customerId, max(amount)
    def reducer_one(self, customerId, amounts):
        sum(amount)
        yield customerId, sum(amount)
    def reducer_two(self, customerId, maxAmounts):
        yield customerId, max(maxAmounts)

if __name__ == '__main__':
    MRCustomer.run()
  
```

```

python C:\Users\HP\Desktop\ML\customer.py C:\Users\HP\Desktop\ML\customer-orders.csv
[...]
Creating temp directory: C:\Users\HP\AppData\Local\Temp\customer.HP.20200913.161323.060632
Running step 1 of 2...
Job output is in C:\Users\HP\AppData\Local\Temp\customer.HP.20200913.161323.060632\output
Streaming final output from C:\Users\HP\AppData\Local\Temp\customer.HP.20200913.161323.060632\output...
  
```

Fig4

Lab – 9

Student Name and Roll Number: Chayan 18csu054

Semester /Section: 5/A

Link to Code: <https://github.com/chayangulati321/Big-Data>

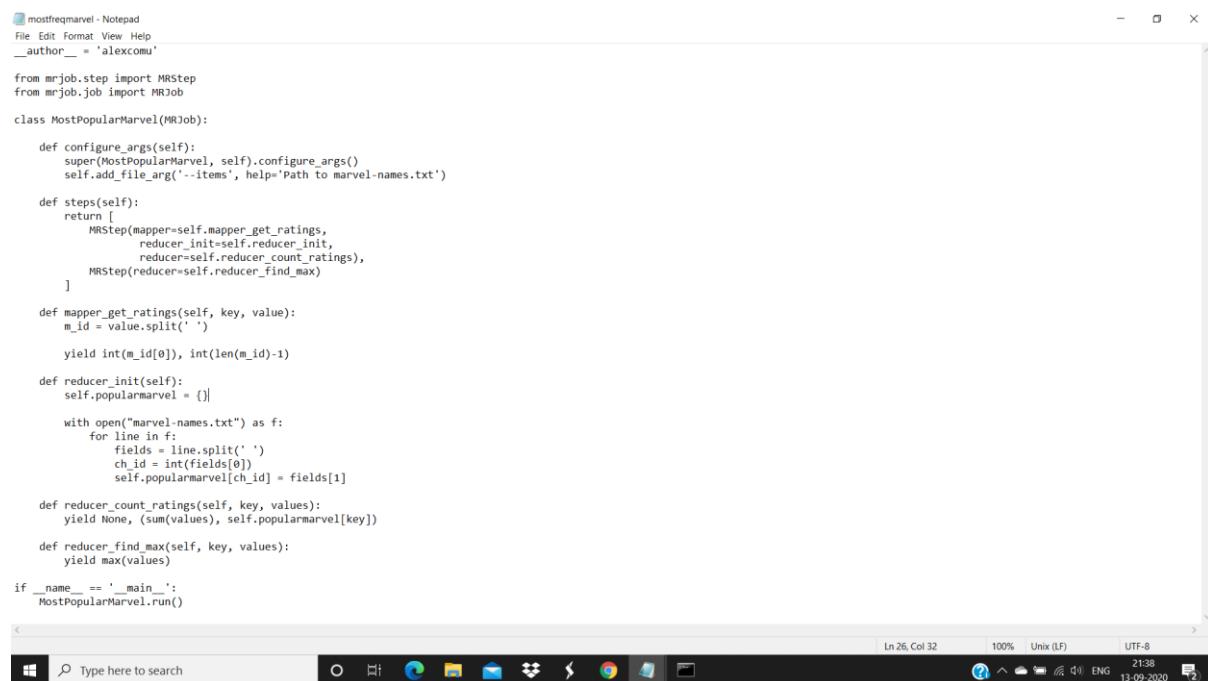
Date: 24/08/20

Faculty Signature:

Remarks:

Aim : Merge two files on the basis of a particular column on the local machine

- Find the most frequent marvel character



```
mostfreqmarvel - Notepad
File Edit Format View Help
__author__ = 'alexcomu'

from mrjob.step import MRStep
from mrjob.job import MRJob

class MostPopularMarvel(MRJob):

    def configure_args(self):
        super(MostPopularMarvel, self).configure_args()
        self.add_file_arg('--items', help='Path to marvel-names.txt')

    def steps(self):
        return [
            MRStep(mapper=self.mapper_get_ratings,
                   reducer_init=self.reducer_init,
                   reducer=self.reducer_count_ratings),
            MRStep(reducer=self.reducer_find_max)
        ]

    def mapper_get_ratings(self, key, value):
        m_id = value.split(' ')
        yield int(m_id[0]), int(len(m_id)-1)

    def reducer_init(self):
        self.popularmarvel = {}

    with open("marvel-names.txt") as f:
        for line in f:
            fields = line.split(' ')
            ch_id = int(fields[0])
            self.popularmarvel[ch_id] = fields[1]

    def reducer_count_ratings(self, key, values):
        yield None, (sum(values), self.popularmarvel[key])

    def reducer_find_max(self, key, values):
        yield max(values)

if __name__ == '__main__':
    MostPopularMarvel.run()
```

Fig5

- To run MRJob run the following commands:-

```
$python <path_of_script> --item=<path_of_file2> <path_of_file1>
```

The screenshot shows a Windows desktop environment. In the foreground, there is a Notepad window titled "mostfreqmarvel - Notepad" containing Python code for an MRJob. The code is a script named "mostfreqmarvel.py" that reads character names from a file and counts their occurrences. A command prompt window titled "Command Prompt" is open, displaying the output of running the script with specific arguments. The desktop taskbar at the bottom shows various pinned icons and the date/time.

```
mostfreqmarvel - Notepad
File Edit Format View Help
__author__ = 'alexcomu'

from mrjob.step import MRStep
from mrjob.job import MRJob

class MostPopularMarvel(MRJob):
    def configure_args(self):
        super(MostPopularMarvel, self)
        self.add_file_arg('--items',
                          'mrjob/marvel-graph.txt')
    def steps(self):
        return [
            MRStep(mapper=self.mapper,
                   reducer_init=self.reducer_init),
            MRStep(reducer=self.reducer)
        ]
    def mapper_get_ratings(self, key, value):
        m_id = value.split(' ')
        yield int(m_id[0]), int(len(m_id))

    def reducer_init(self):
        self.popularmarvel = {}

    with open("marvel-names.txt"):
        for line in f:
            fields = line.split(',')
            ch_id = int(fields[0])
            self.popularmarvel[ch_id] = fields[1]

    def reducer_count_ratings(self, key, values):
        yield None, (sum(values), self.popularmarvel[key])

    def reducer_find_max(self, key, values):
        yield max(values)

if __name__ == '__main__':
    MostPopularMarvel.run()
```

```
C:\Users\HP\Desktop\ML\mostfreqmarvel.py --item=C:\Users\HP\Desktop\ML\marvel-names.txt C:\Users\HP\Desktop\ML\marvel-graph.txt
E:\Users\HP\Python37\python C:\Users\HP\Desktop\ML\mostfreqmarvel.py --item=C:\Users\HP\Desktop\ML\marvel-names.txt C:\Users\HP\Desktop\ML\marvel-graph.txt
No configs specified for inline runner
Creating temp directory C:\Users\HP\AppData\Local\Temp\mostfreqmarvel.HP.20200913.161006.815364...
Running step 1 of 2...
Running step 2 of 2...
Job output is in C:\Users\HP\AppData\Local\Temp\mostfreqmarvel.HP.20200913.161006.815364\output
1937  "\CAPTAIN"
Removing temp directory C:\Users\HP\AppData\Local\Temp\mostfreqmarvel.HP.20200913.161006.815364...
```

Fig6

Lab – 10

Student Name and Roll Number: Chayan 18csu054

Semester /Section: 5/A

Link to Code: <https://github.com/chayangulati321/Big-Data>

Date: 26/08/20

Faculty Signature:

Remarks:

AIM : To execute a MapReduce job using Dataproc service on GCP

Step – 1 : Creating a cluster using Dataproc :

- Select Navigation menu > Big Data > Dataproc > Clusters
- Click on the CREATE CLUSTER
- Now name the cluster and rest set as default
- Then click Create

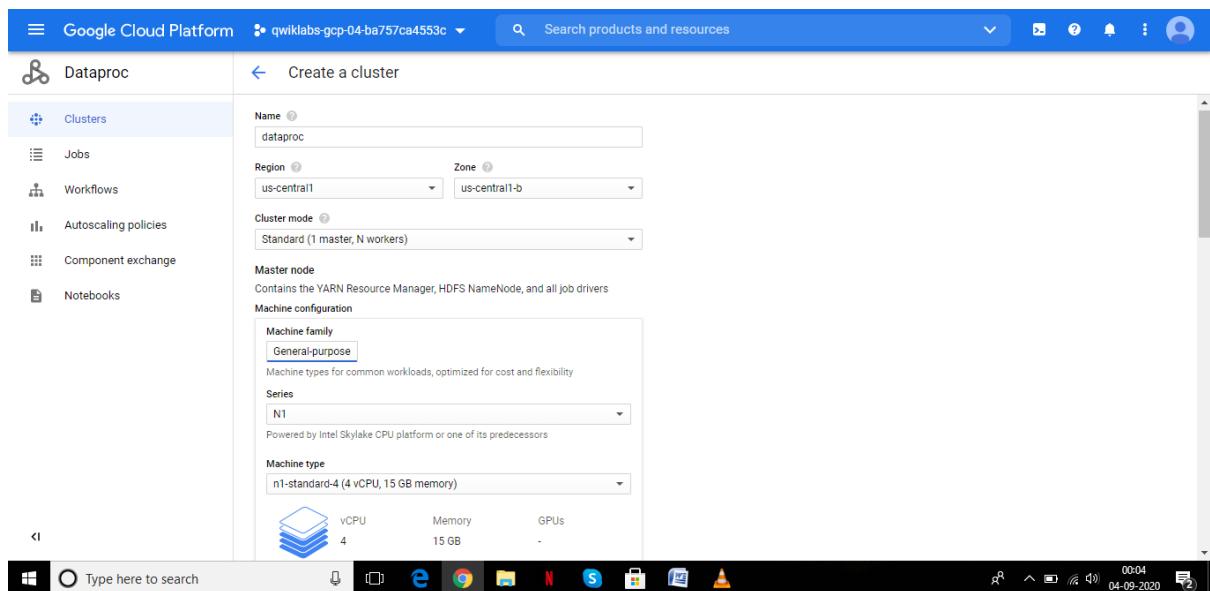


Fig1

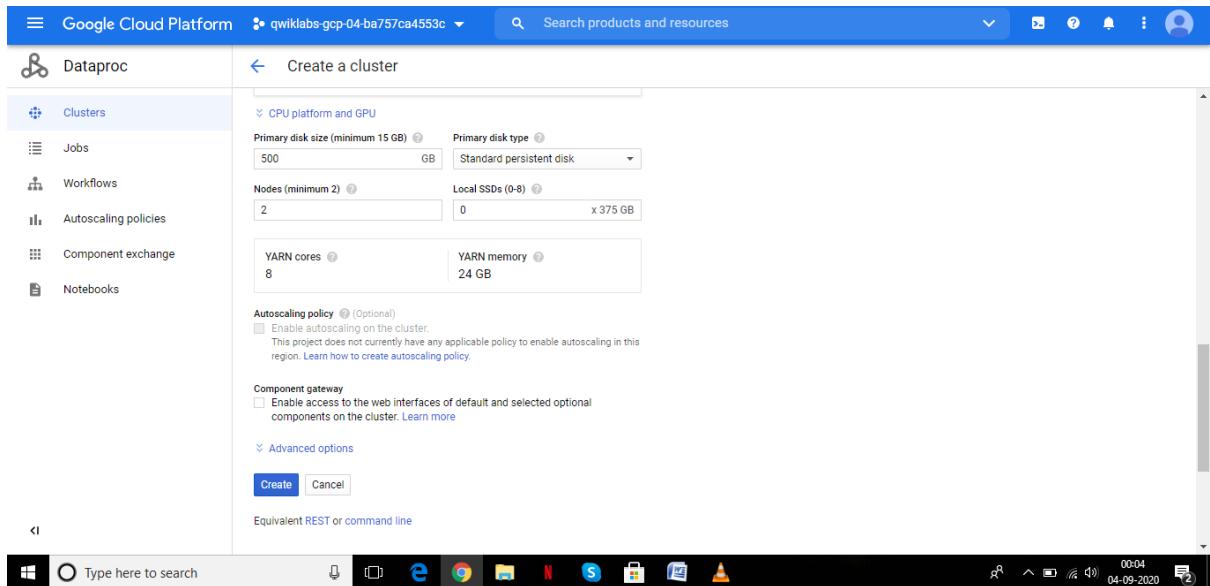


Fig2

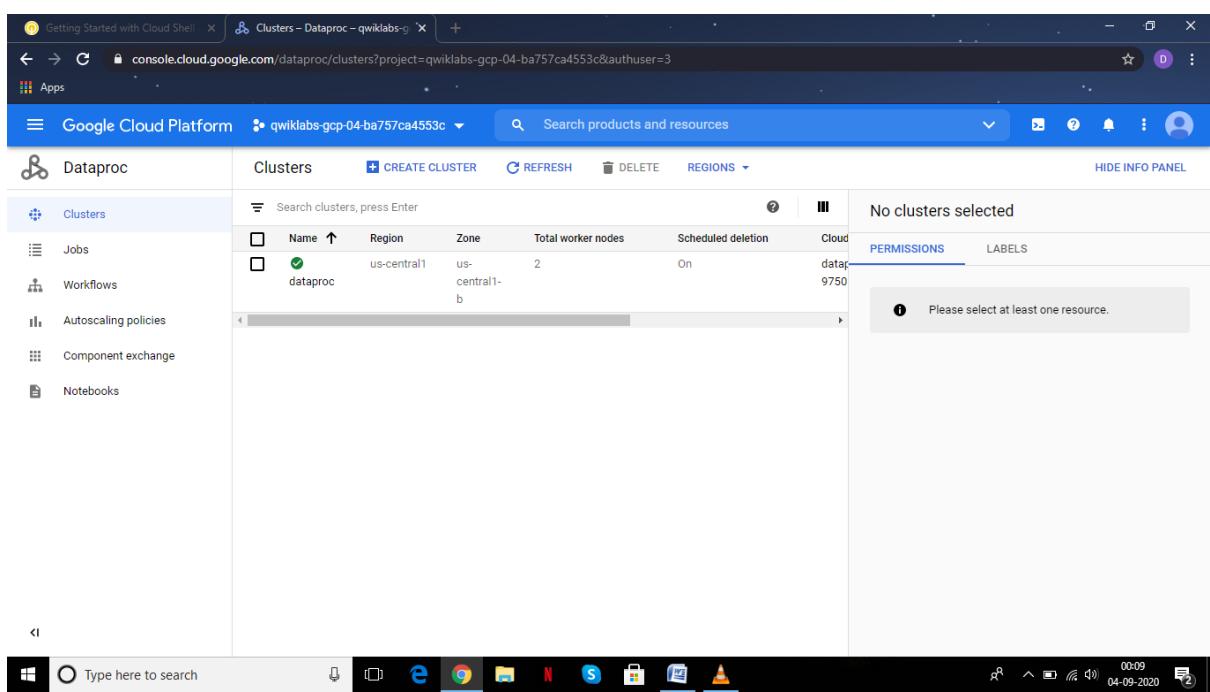


Fig3

Step – 2 : Create a bucket and upload the file containing mapper reducer logic in it

- Select Navigation menu > Storage > Browser, and then click Create bucket
- Enter a unique name to bucket, keep all other settings unchanged, and hit Create

A bucket will be created

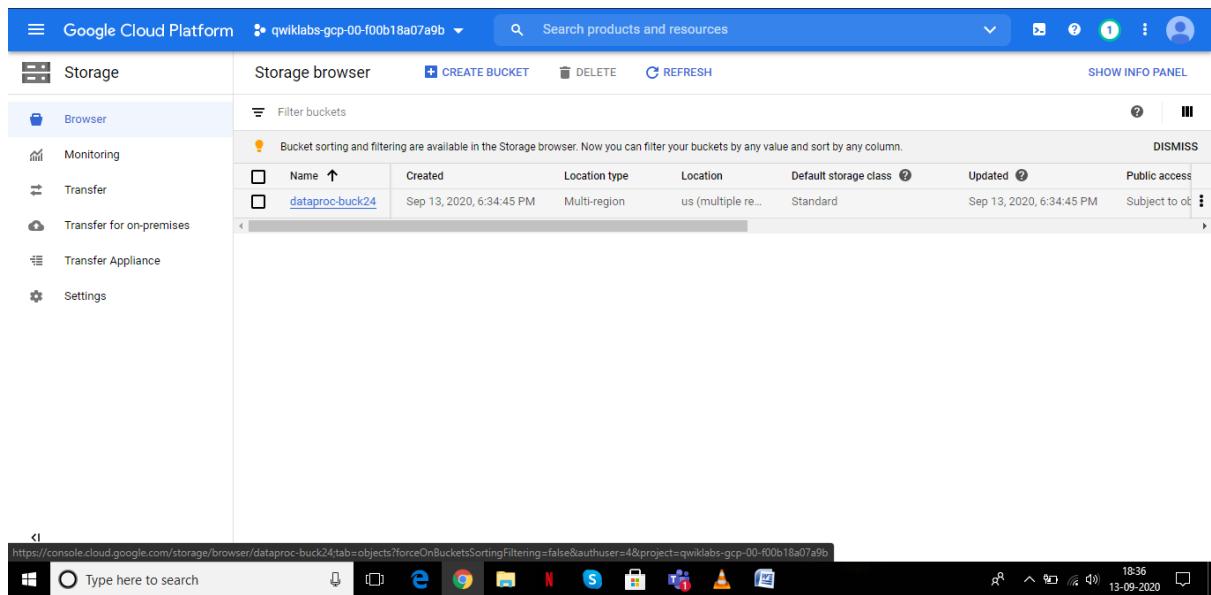


Fig4

- Create 2 folders inside the bucket, named `mapr_input` and `mapr_script`, to store data and other to store the files
- Upload the data file into `mapr_input` and the mapper and reducer code files in the `mapr_script` folders resp.

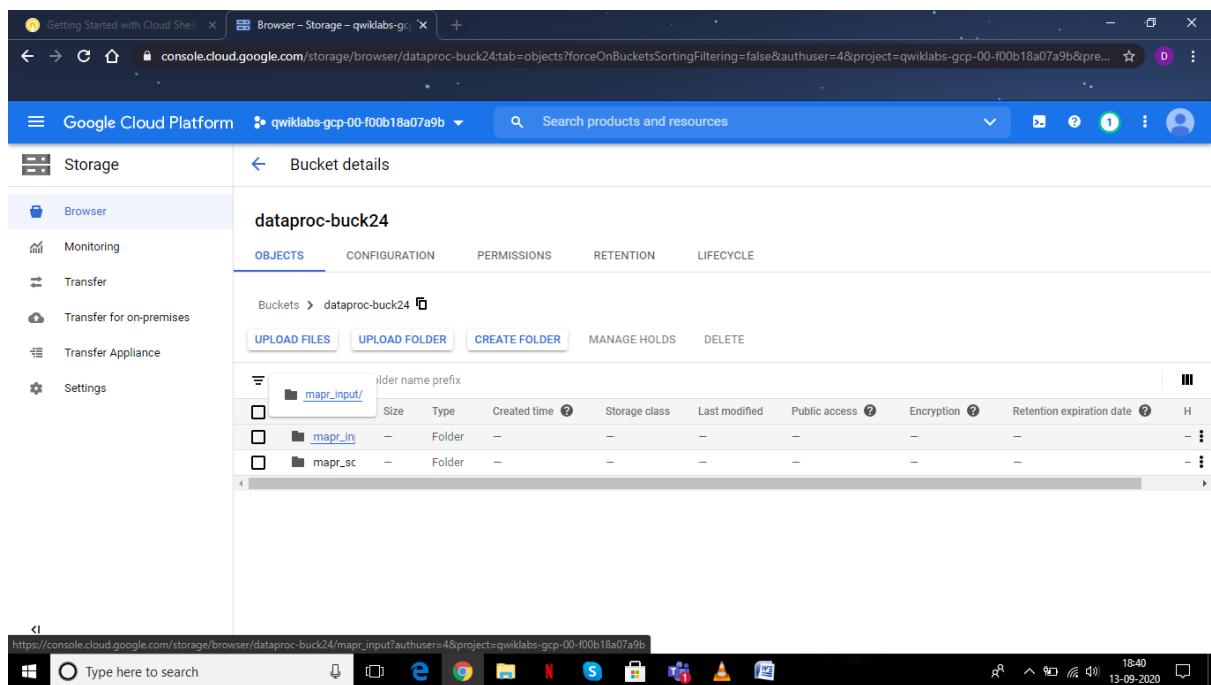


Fig5

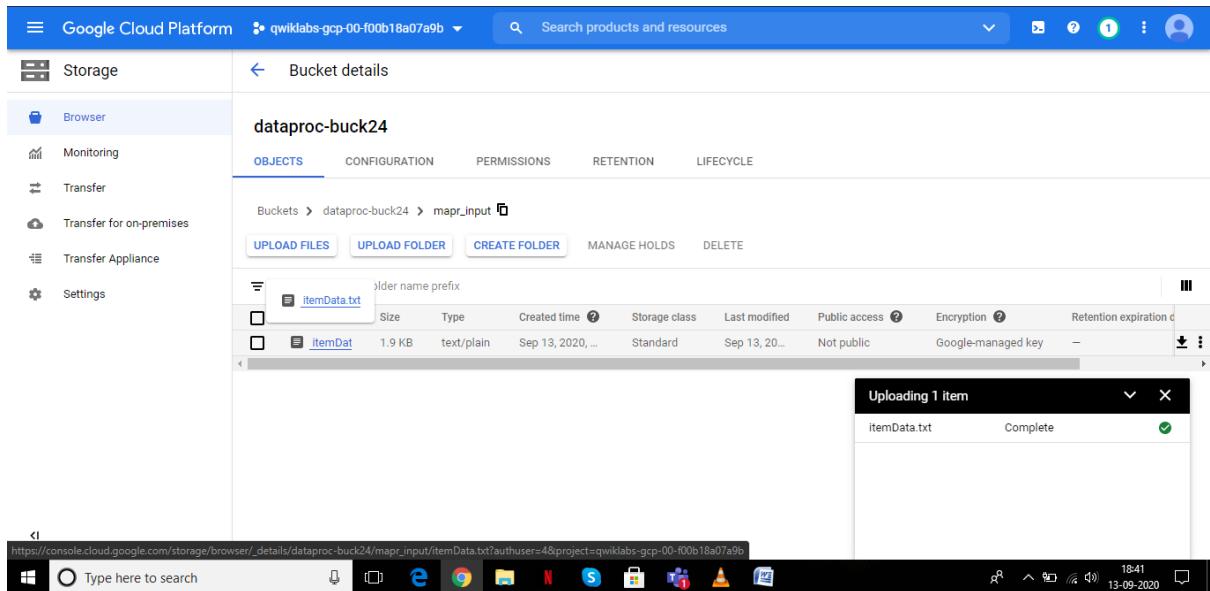


Fig6

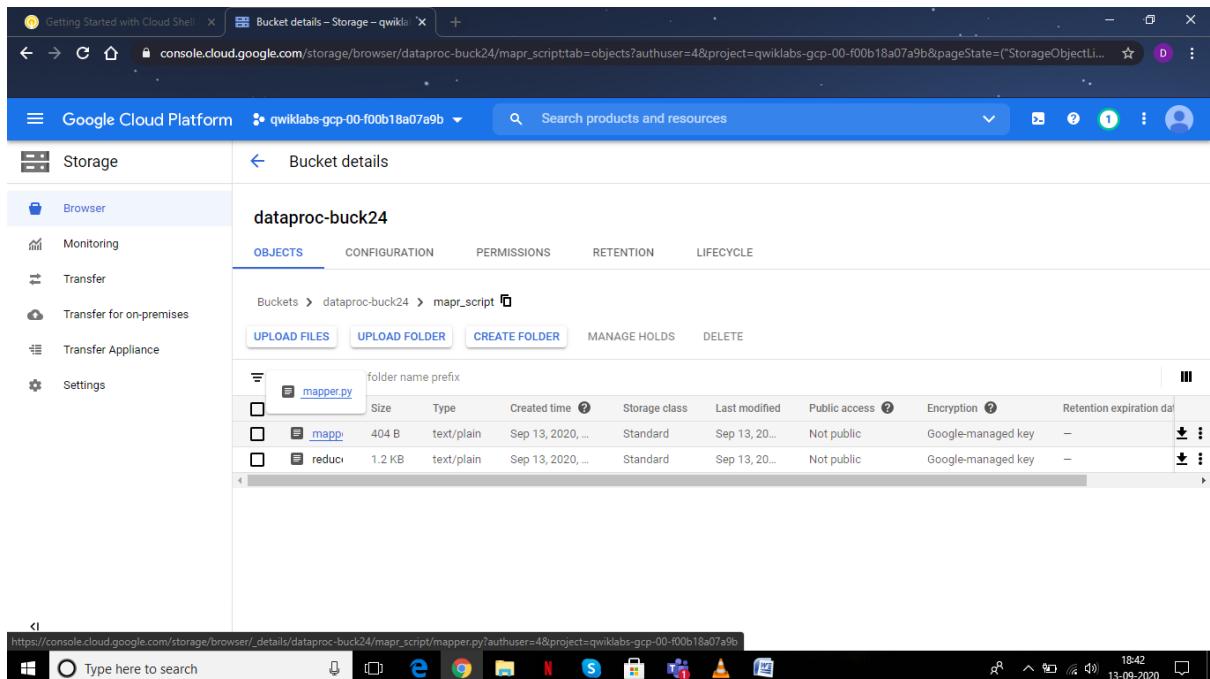


Fig7

Running the jobs using SSH

Step-3: To submit the jobs using the master node SSH

1. Select Navigation menu > Big Data > Dataproc > Clusters > dataproc > VM INSTANCES
2. And then click on the SSH of the master node (on the right)

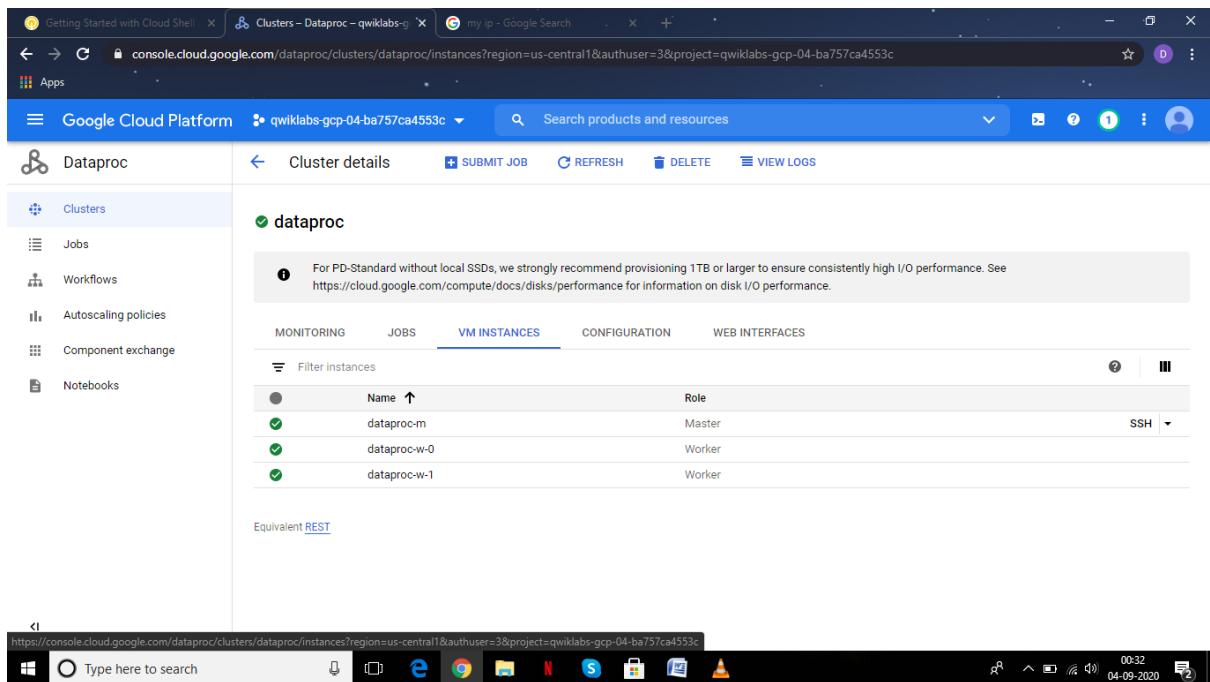


Fig8

Type command in SSH as follows :

- To see the data that is in the directory

```
$ Hadoop fs -ls /
```

- To execute the syntax of mapper and reducer (which is on the bucket) onto the cluster created

```
$hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar \
```

-files

```
gs://dataproc-buck24/mapr_script/mapper.py,gs://dataproc-
buck24/mapr_script/reducer.py \
```

```
-mapper 'python mapper.py' \
```

```
-reducer 'python reducer.py' \
```

```
-input gs://dataproc-buck24/mapr_input/itemData.txt \
```

```
-output gs://dataproc-buck24/mapr_output
```

```

student-00-a2406d51d71@dataproc-m: ~ - Google Chrome
ssh.cloud.google.com/projects/qwiklabs-gcp-00-b0b18a07a9b/zones/us-central1-a/instances/dataproc-m?authuser=4&hl=en_US&pr...
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Creating directory '/home/student-00-a2406d51d71'.
student-00-a2406d51d71@dataproc-m:~$ hadoop fs -ls /
Found 3 items
drwxr-xr-x - mapred hadoop 0 2020-09-13 13:00 /hadoop
drwxrwxrwt - hdfs hadoop 0 2020-09-13 13:00 /tmp
drwxrwxrwt - hdfs hadoop 0 2020-09-13 13:00 /user
student-00-a2406d51d71@dataproc-m:~$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar \
> -files gs://dataproc-buck24/mapr_script/mapper.py,gs://dataproc-buck24/mapr_script/reducer.py \
> -mapper 'python mapper.py' \
> -reducer 'python reducer.py' \
> -input gs://dataproc-buck24/mapr_input/itemData.txt \
> -output gs://dataproc-buck24/mapr_output
packageJobJar: [] [/usr/lib/hadoop-mapreduce/hadoop-streaming-2.9.2.jar] /tmp/streamjob6779814696019763547.jar tmpDir null
20/09/13 13:29:15 INFO client.RMProxy: Connecting to ResourceManager at dataproc-m/10.128.0.2:8032
20/09/13 13:29:15 INFO client.RMProxy: Connecting to Application History server at dataproc-m/10.128.0.2:10200
20/09/13 13:29:15 INFO client.RMProxy: Connecting to Application History server at dataproc-m/10.128.0.2:8032
20/09/13 13:29:16 INFO mapred.FileInputFormat: Total input files to process : 1
20/09/13 13:29:16 INFO mapreduce.JobSubmitter: number of splits:2
20/09/13 13:29:16 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
20/09/13 13:29:17 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1600002013452_0001
20/09/13 13:29:17 INFO impl.YarnClientImpl: Submitted application application_1600002013452_0001
20/09/13 13:29:17 INFO mapreduce.Job: The url to track the job: http://dataproc-m:8088/proxy/application_1600002013452_0001/
20/09/13 13:29:17 INFO mapreduce.Job: Running job: job_1600002013452_0001
20/09/13 13:29:25 INFO mapreduce.Job: Job job_1600002013452_0001 running in uber mode : false
20/09/13 13:29:25 INFO mapreduce.Job: map 0% reduce 0%
20/09/13 13:29:34 INFO mapreduce.Job: map 14% reduce 0%
20/09/13 13:29:37 INFO mapreduce.Job: map 33% reduce 0%
20/09/13 13:29:42 INFO mapreduce.Job: map 48% reduce 0%
20/09/13 13:29:46 INFO mapreduce.Job: map 67% reduce 0%
20/09/13 13:29:50 INFO mapreduce.Job: map 81% reduce 0%
20/09/13 13:29:56 INFO mapreduce.Job: map 86% reduce 0%
20/09/13 13:29:57 INFO mapreduce.Job: map 100% reduce 0%
20/09/13 13:30:06 INFO mapreduce.Job: map 100% reduce 29%
20/09/13 13:30:08 INFO mapreduce.Job: map 100% reduce 71%
20/09/13 13:30:09 INFO mapreduce.Job: map 100% reduce 86%

```

Type here to search 19:03 13-09-2020

Fig9

- To see the content of the output in the part files

\$hdfs dfs -cat gs://dataproc-buck24/mapr_output/part-00000

\$hdfs dfs -cat gs://dataproc-buck24/mapr_output/part-00001

```

student-00-a2406d51d71@dataproc-m: ~ - Google Chrome
ssh.cloud.google.com/projects/qwiklabs-gcp-00-b0b18a07a9b/zones/us-central1-a/instances/dataproc-m?authuser=4&hl=en_US&pr...
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Creating directory '/home/student-00-a2406d51d71'.
student-00-a2406d51d71@dataproc-m:~$ hadoop fs -ls /
Found 3 items
drwxr-xr-x - mapred hadoop 0 2020-09-13 13:00 /hadoop
drwxrwxrwt - hdfs hadoop 0 2020-09-13 13:00 /tmp
drwxrwxrwt - hdfs hadoop 0 2020-09-13 13:00 /user
student-00-a2406d51d71@dataproc-m:~$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar \
> -files gs://dataproc-buck24/mapr_script/mapper.py,gs://dataproc-buck24/mapr_script/reducer.py \
> -mapper 'python mapper.py' \
> -reducer 'python reducer.py' \
> -input gs://dataproc-buck24/mapr_input/itemData.txt \
> -output gs://dataproc-buck24/mapr_output
packageJobJar: [] [/usr/lib/hadoop-mapreduce/hadoop-streaming-2.9.2.jar] /tmp/streamjob6779814696019763547.jar tmpDir null
20/09/13 13:29:17 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1600002013452_0001
20/09/13 13:29:17 INFO impl.YarnClientImpl: Submitted application application_1600002013452_0001
20/09/13 13:29:17 INFO mapreduce.Job: The url to track the job: http://dataproc-m:8088/proxy/application_1600002013452_0001/
20/09/13 13:29:17 INFO mapreduce.Job: Running job: job_1600002013452_0001
20/09/13 13:29:25 INFO mapreduce.Job: Job job_1600002013452_0001 running in uber mode : false
20/09/13 13:29:25 INFO mapreduce.Job: map 0% reduce 0%
20/09/13 13:29:34 INFO mapreduce.Job: map 14% reduce 0%
20/09/13 13:29:37 INFO mapreduce.Job: map 33% reduce 0%
20/09/13 13:29:42 INFO mapreduce.Job: map 48% reduce 0%
20/09/13 13:29:46 INFO mapreduce.Job: map 67% reduce 0%
20/09/13 13:29:50 INFO mapreduce.Job: map 81% reduce 0%
20/09/13 13:29:56 INFO mapreduce.Job: map 86% reduce 0%
20/09/13 13:29:57 INFO mapreduce.Job: map 100% reduce 0%
20/09/13 13:30:06 INFO mapreduce.Job: map 100% reduce 29%
20/09/13 13:30:08 INFO mapreduce.Job: map 100% reduce 71%
20/09/13 13:30:09 INFO mapreduce.Job: map 100% reduce 86%
20/09/13 13:30:10 INFO mapreduce.Job: map 100% reduce 100%
20/09/13 13:30:11 INFO mapreduce.Job: Job job_1600002013452_0001 completed successfully
20/09/13 13:30:11 INFO mapreduce.Job: Counters: 55
File System Counters
FILE: Number of bytes read=771
FILE: Number of bytes written=5934340
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
GS: Number of bytes read=21336
GS: Number of bytes written=275
GS: Number of read operations=0
GS: Number of large read operations=0
GS: Number of write operations=0
HDFS: Number of bytes read=2016
HDFS: Number of bytes written=0
HDFS: Number of read operations=21
HDFS: Number of large read operations=0
HDFS: Number of write operations=0
Job Counters
Killed map tasks=1
Launched map tasks=21
Launched reduce tasks=7
Rack-local map tasks=21
Total time spent by all maps in occupied slots (ms)=499950
Total time spent by all reduces in occupied slots (ms)=150150

```

Type here to search 19:03 13-09-2020

Fig10

The screenshot shows the Google Cloud Platform interface. On the left, the navigation menu includes Storage, Browser, Monitoring, Transfer, Transfer for on-premises, Transfer Appliance, and Settings. The main content area displays a StreamJob output directory: gs://dataproc-buck24/mapr_output/part-00000. A Cloud Shell terminal window is open, showing the following command and its output:

```

Merged Map outputs=147
GC time elapsed (ms)=5049
CPU time spent (ms)=30080
Physical memory (bytes) snapshot=13842792448
Virtual memory (bytes) snapshot=122296541184
Total committed heap usage (bytes)=11923681984

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=21336
File Output Format Counters
  Bytes Written=275

20/09/13 13:30:11 INFO streaming.StreamJob: Output directory: gs://dataproc-buck24/mapr_output
student-00-a82406d51d71@dataproc-m:~$ hdfs dfs -cat gs://dataproc-buck24/mapr_output/part-00000
Cameras      485.71
DVDs        492.8
Music       213.64
student-00-a82406d51d71@dataproc-m:~$ hdfs dfs -cat gs://dataproc-buck24/mapr_output/part-00001
Crafts      489.93
Garden      386.56
student-00-a82406d51d71@dataproc-m:~$ hdfs dfs -cat gs://dataproc-buck24/mapr_output/part-00002
ConsumerElectronics   410.37
HealthandBeauty     464.36
HealthandBeauty     157.91
PetSupplies      164.5
student-00-a82406d51d71@dataproc-m:~$ hdfs dfs -cat gs://dataproc-buck24/mapr_output/part-00003
Books         498.29
Computers      288.32
Men'sClothing    388.3
Toys          13.79
Women'sClothing  481.31
student-00-a82406d51d71@dataproc-m:~$ hdfs dfs -cat gs://dataproc-buck24/mapr_output/part-00005
student-00-a82406d51d71@dataproc-m:~$ hdfs dfs -cat gs://dataproc-buck24/mapr_output/part-00006
VideoGames     349.41
student-00-a82406d51d71@dataproc-m:~$ 

```

The taskbar at the bottom shows various application icons.

Fig11

Running the jobs using Command Line or Cloud Shell

Step-4: Running the jobs from the Command Line

- Select Navigation menu > Big Data > Dataproc > Jobs
- Now activate the Cloud Shell by clicking Continue and type the following command :-


```
$gcloud dataproc jobs submit hadoop --cluster dataproc --region us-central1 \
--jar file:///usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples-2.9.2.jar \
-- wordcount gs://dataproc-buck24/mapr_input/itemData.txt gs://dataproc-
buck24/mapr_java_output
```

Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to **qwiklabs-gcp-01-c1695821db93**.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
student 01 46da61a6c8@cloudshell: ~(qwiklabs-gcp-01-c1695821db93)\$ gcloud dataproc jobs submit hadoop --cluster dataproc --region us-central1 \> --jar file:///usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples-2.9.2.jar \> --wordcount gs://dataproc-buck24/mapr_input/itemData.txt gs://dataproc-buck24/mapr_java_output
Job [54aea774bf6e46e0b3e26576f2925efe] submitted.
Waiting for job output...
20/09/13 14:50:04 INFO client.RMProxy: Connecting to ResourceManager at dataproc-m/10.128.0.3:8032
20/09/13 14:50:04 INFO client.AHSProxy: Connecting to Application History server at dataproc-m/10.128.0.3:10200
20/09/13 14:50:05 INFO input.FileInputFormat: Total input files to process : 1
20/09/13 14:50:05 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
20/09/13 14:50:05 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1600008508803_0001
20/09/13 14:50:06 INFO impl.YarnClientImpl: Submitted application application_1600008508803_0001
20/09/13 14:50:06 INFO mapreduce.Job: The url to track the job: http://dataproc-m:8088/proxy/application_1600008508803_0001/
20/09/13 14:50:06 INFO mapreduce.Job: Running job: job_1600008508803_0001
20/09/13 14:50:14 INFO mapreduce.Job: map 0% reduce 0%
20/09/13 14:50:21 INFO mapreduce.Job: map 100% reduce 0%
20/09/13 14:50:30 INFO mapreduce.Job: map 100% reduce 29%
20/09/13 14:50:32 INFO mapreduce.Job: map 100% reduce 57%

Fig1

- Now a job will be created under the Job section as shown:

Job ID	Region	Type	Cluster	Start time	Elapsed time	Status
54aea774bf6e46e0b3e26576f2925efe	us-central1	Hadoop	dataproc	Sep 13, 2020, 8:20:01 PM	36 sec	Succeeded

```
reference:  

jobId: 54aea774bf6e46e0b3e26576f2925efe  

projectId: qwiklabs-gcp-01-c1695821db93  

status:  

state: DONE  

stateStartTime: '2020-09-13T14:50:37.404Z'  

statusHistory:  

- state: PENDING  

stateStartTime: '2020-09-13T14:50:01.391Z'  

- state: SETTING_UP  

stateStartTime: '2020-09-13T14:50:01.425Z'  

- details: Agent reported job success  

state: RUNNING  

stateStartTime: '2020-09-13T14:50:01.627Z'  

yarnApplications:  

- name: word count  

progress: 1.0  

state: FINISHED  

trackingUrl: http://dataproc-m:8088/proxy/application_1600008508803_0001/  

student 01 46da61a6c8@cloudshell: ~(qwiklabs-gcp-01-c1695821db93)$ ls  

https://console.cloud.google.com/dataproc/jobs/54aea774bf6e46e0b3e26576f2925efe/project=qwiklabs-gcp-01-c1695821db93&region=us-central1
```

Fig2

- Now run the following command to see the content of the output part files
- gsutil cat gs://dataproc-buck24/mapr_java_output/part-r-00000

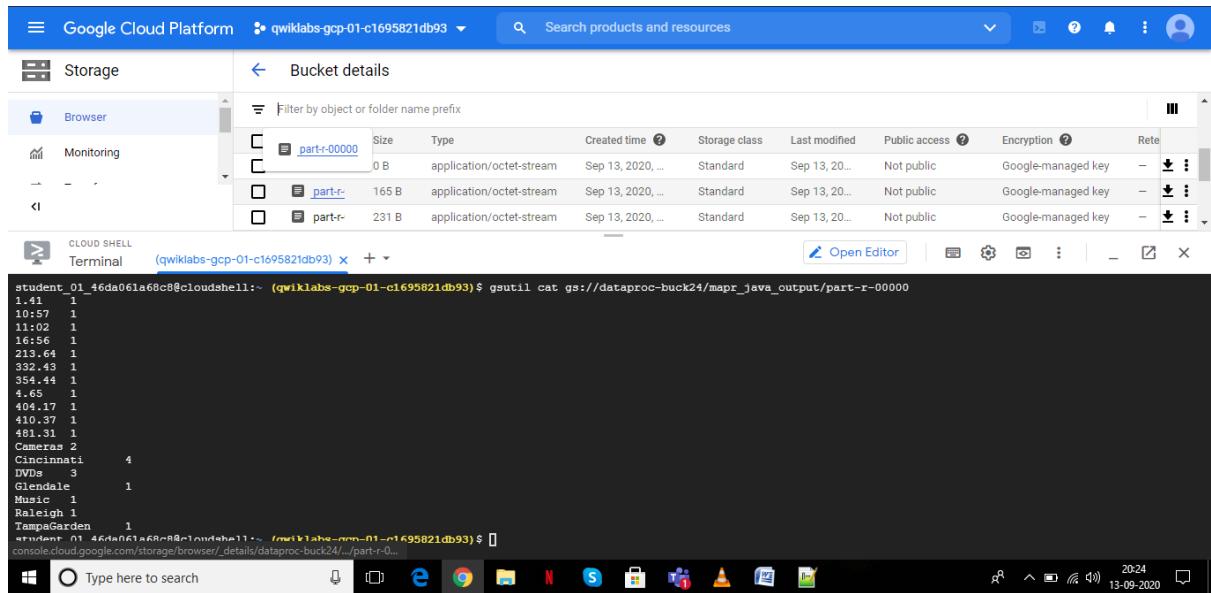


Fig3

Lab – 11

Student Name and Roll Number: Chayan 18csu054

Semester /Section: 5/A

Link to Code: <https://github.com/chayangulati321/Big-Data>

Date: 26/08/20

Faculty Signature:

Remarks:

AIM : To execute a MapReduce job using Dataproc service on GCP

Step – 1 : Creating a cluster using Dataproc :

- Select Navigation menu > Big Data > Dataproc > Clusters
- Click on the CREATE CLUSTER
- Now name the cluster and rest set as default
- Then click Create

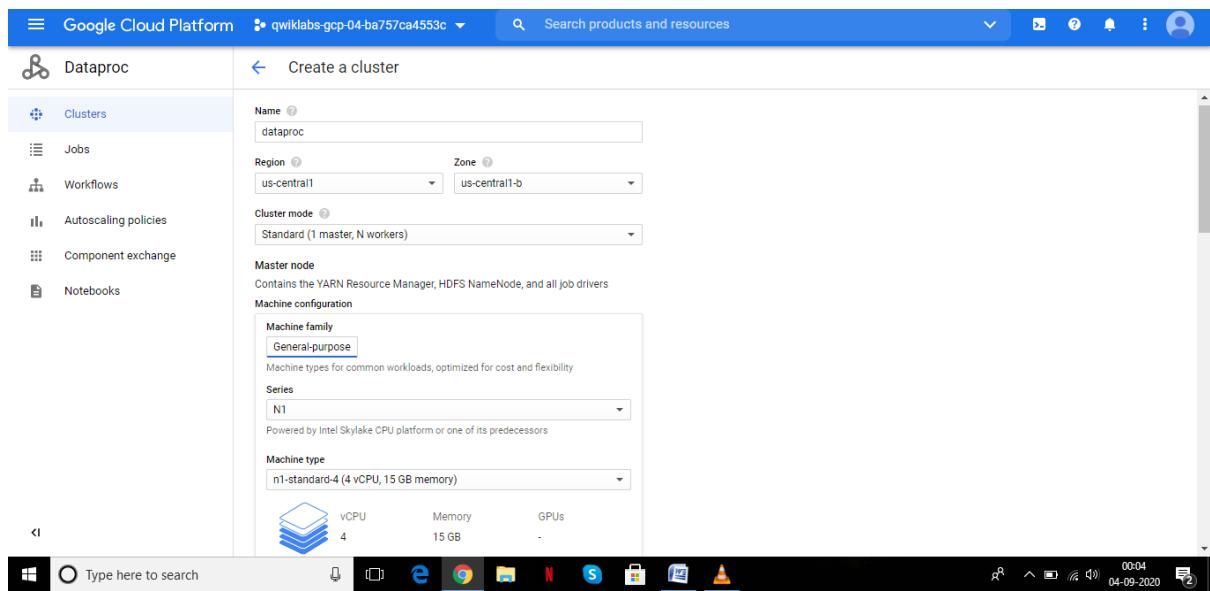


Fig1

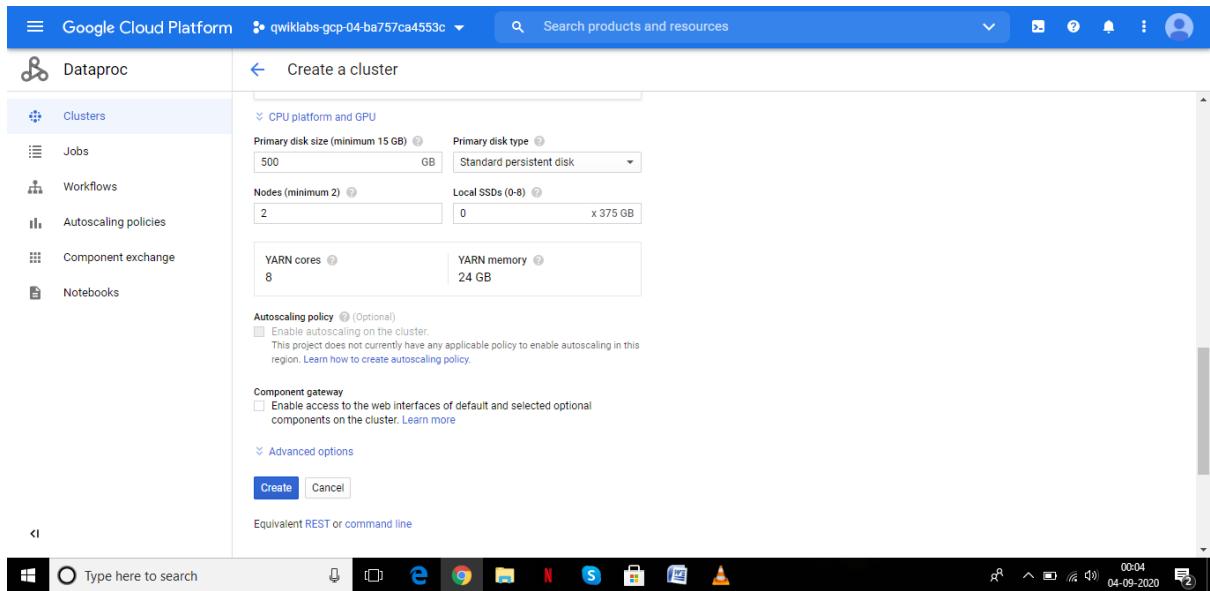


Fig2

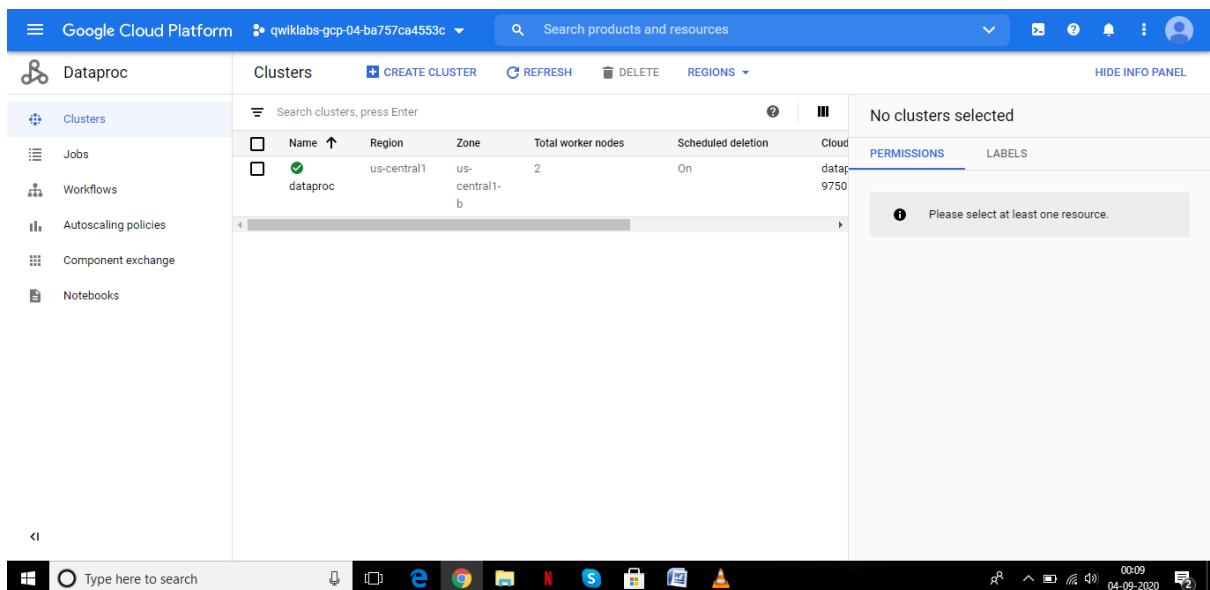


Fig3

Step – 2 : Downloading the JSON file which is required for the execution of the MRJob onto the cloud and creating key for app engine default service account

- Select Navigation menu > IAM and Admin > Service Accounts
- Go to the first link associated with the project id of the qwiklabs account and go to Actions and click Create Key
- Click on JSON and a file would be downloaded on the local system

The screenshot shows the Google Cloud Platform IAM & Admin Service accounts page for project "qwiklabs-gcp-03-687f6e75a79d". The left sidebar has "Service Accounts" selected. The main area displays three service accounts:

Email	Status	Name	Description	Key ID
qwiklabs-gcp-03-687f6e75a79d@appspot.gserviceaccount.com	Green checkmark	App Engine default service account	No keys	
327056227779-compute@developer.gserviceaccount.com	Green checkmark	Compute Engine default service account	No keys	
qwiklabs-gcp-03-687f6e75a79d@qwiklabs-gcp-03-687f6e75a79d.iam.gserviceaccount.com	Green checkmark	QwikLabs User Service account	2dbe77f0b2303a509c1dd5e9aaa7f83db32feae	

A context menu is open over the third service account, listing options: Edit, Disable, Create key, and Delete.

Fig4

The screenshot shows the Google Cloud Platform IAM & Admin Service accounts page for project "qwiklabs-gcp-03-687f6e75a79d". A modal dialog is displayed in the center of the screen with the message "Private key saved to your computer". Below the message, a warning icon states: "qwiklabs-gcp-03-687f6e75a79d-8d9248b0c295.json allows access to your cloud resources, so store it securely." At the bottom right of the modal is a "CLOSE" button.

Fig5

Step – 3 :To submit the jobs using the master node's SSH

1. Select Navigation menu > Big Data > Dataproc > Clusters > dataproc > VM INSTANCES
2. And then click on the SSH of the master node (on the right)

The screenshot shows the Google Cloud Platform interface for managing a Dataproc cluster named 'dataproc'. The left sidebar has 'Dataproc' selected under 'Clusters'. The main area shows 'Cluster details' with tabs for 'MONITORING', 'JOBS', 'VM INSTANCES' (which is active and highlighted in blue), 'CONFIGURATION', and 'WEB INTERFACES'. Below these tabs is a table with columns 'Name', 'Role', and 'SSH'. The table contains three rows: 'dataproc-m' (Role: Master), 'dataproc-w-0' (Role: Worker), and 'dataproc-w-1' (Role: Worker). At the bottom of the table is a link 'Equivalent REST'. The browser address bar shows the URL: <https://console.cloud.google.com/dataproc/clusters/dataproc/instances?region=us-central1&authuser=2&project=qwiklabs-gcp-04-ba757ca4552c>. The taskbar at the bottom includes icons for File, Home, Task View, Start, Edge, Google Chrome, File Explorer, File History, Netflix, OneDrive, and VLC.

Fig6

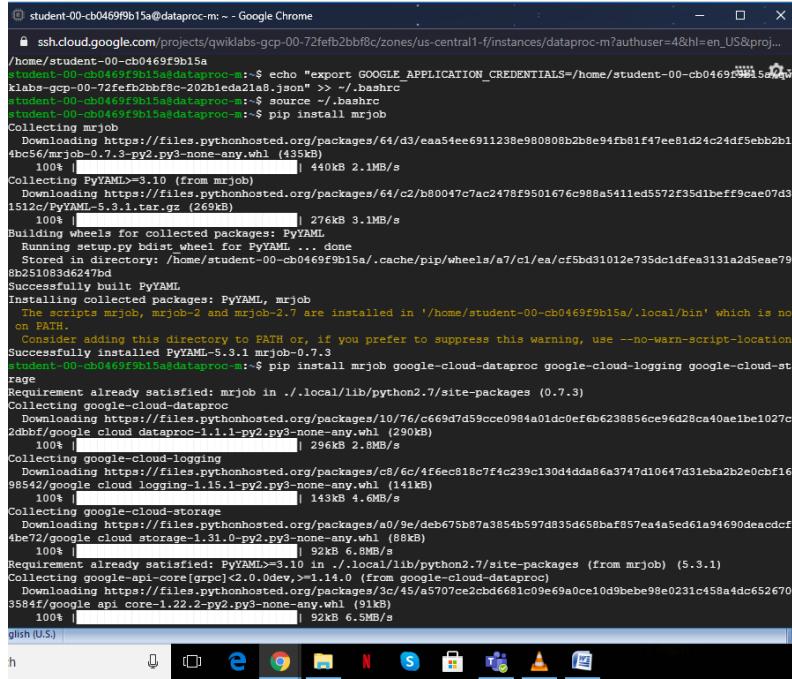
Go to settings icon and click on upload a file

The screenshot shows an SSH session in Google Chrome. The terminal window displays a login prompt for 'student-03-443cdf6b834f@dataproc-m'. The user has run several commands, including 'ssh.cloud.google.com/projects/qwiklabs-gcp-03-687f6e75a79d/zones/us-central1-a/instances/dataproc-m?authuser=4&hl=en_US&proto=3.0&xauthuser=student-03-443cdf6b834f'. A context menu is open over the terminal window, listing options like 'Color Themes', 'Text Size', 'Font', etc. A tooltip message from the terminal window reads: 'Please consider adding the IAP-secured Tunnel User IAM role to start using Cloud IAP for TCP forwarding for better performance. Learn more Dismiss'. The taskbar at the bottom includes icons for File, Home, Task View, Start, Edge, Google Chrome, File Explorer, File History, Netflix, OneDrive, and VLC.

Fig7

3. Now write the following commands in SSH:

- \$echo "export GOOGLE_APPLICATION_CREDENTIALS=/home/student-00-cb0469f9b15a/qwiklabs-gcp-00-72fefb2bbf8c-202b1eda21a8.json" >> ~/.bashrc
- \$source ~/.bashrc
- \$pip install mrjob
- \$pip install mrjob google-cloud-dataproc google-cloud-logging google-cloud-storage



```

student-00-cb0469f9b15a@dataproc-m: ~ - Google Chrome
ssh.cloud.google.com/projects/qwiklabs-gcp-00-72fefb2bbf8c/zones/us-central1-f/instances/dataproc-m?authuser=4&hl=en_US&proj...
/home/student-00-cb0469f9b15a
student-00-cb0469f9b15a$ dataproc-m:~$ echo "export GOOGLE_APPLICATION_CREDENTIALS=/home/student-00-cb0469f9b15a/qwik...
Klabs-gcp-00-72fefb2bbf8c-202b1eda21a8.json" >> ~/.bashrc
student-00-cb0469f9b15a$ source ~/.bashrc
student-00-cb0469f9b15a$ dataproc-m:~$ pip install mrjob
Collecting mrjob
  Downloading https://files.pythonhosted.org/packages/64/d3/ea54ee6911238e980808b2b8e94fb81f47ee81d24c24df5eb2b11
4bc56/mrjob-0.7.3-py2.py3-none-any.whl (435kB)
    100% |██████████| 440kB 2.1MB/s
Collecting PyYAML<3.10 (from mrjob)
  Downloading https://files.pythonhosted.org/packages/64/c2/b80047c7ac2478f9501676c988a5411ed5572f35d1beff9cae07d32
1512c/PyYAML-5.3.1.tar.gz (269kB)
    100% |██████████| 276kB 3.1MB/s
Building wheels for collected packages: PyYAML
  Running setup.py bdist_wheel for PyYAML ... done
    Stored in directory: /home/student-00-cb0469f9b15a/.cache/pip/wheels/a7/c1/ea/cf5bd31012e735dcldfea3131a2d5eae797
8b251083d6247bd
Successfully built PyYAML
Installing collected packages: PyYAML, mrjob
  The scripts mrjob, mrjob-2, and mrjob-2.7 are installed in '/home/student-00-cb0469f9b15a/.local/bin' which is not
  on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed PyYAML-5.3.1 mrjob-0.7.3
student-00-cb0469f9b15a$ dataproc-m:~$ pip install mrjob google-cloud-dataproc google-cloud-logging google-cloud-sto
rage
Requirement already satisfied: mrjob in ./local/lib/python2.7/site-packages (0.7.3)
Collecting google-cloud-dataproc
  Downloading https://files.pythonhosted.org/packages/10/76/c669d7d59cce0984a01dc0ef6b6238856ce96d28ca40aelbe1027c7
2dbbf/google cloud dataproc-1.1.1-py2.py3-none-any.whl (290kB)
    100% |██████████| 296kB 2.8MB/s
Collecting google-cloud-logging
  Downloading https://files.pythonhosted.org/packages/c8/6c/4f6ec818c7f4c239c130d4ddaa86a3747d10647d31eba2b2e0cbf160
98542/google cloud logging-1.15.1-py2.py3-none-any.whl (141kB)
    100% |██████████| 143kB 4.6MB/s
Collecting google-cloud-storage
  Downloading https://files.pythonhosted.org/packages/a0/se/deb675bb7a3854b597d835d658ba857ea4a5ed61a94690deacd0fc
4be72/google cloud storage-1.31.0-py2.py3-none-any.whl (88kB)
    100% |██████████| 92kB 6.6MB/s
Requirement already satisfied: PyYAML<3.10 in ./local/lib/python2.7/site-packages (from mrjob) (5.3.1)
Collecting google-api-core[grpc]<2.0.0.dev,>=1.14.0 (from google-cloud-dataproc)
  Downloading https://files.pythonhosted.org/packages/3d/45/a5707ce2cd6681c09e69a0ce10d9beb98e0231c458a4dc652670f
3584f/google api core-1.22.2-py2.py3-none-any.whl (91kB)
    100% |██████████| 92kB 6.5MB/s
gish (U.S.) | h

```

Fig8

- Now upload the file and dataset similarly using the upload file option and type 'ls' to see the content

```

CS 123: mrjob in a cluster
student-00-cb0469f9b15a@dataproc-m: ~ - Google Chrome
ssh.cloud.google.com/projects/qwiklabs-gcp-00-72febf2bbf8c/zones/us-central1-f/instances/dataproc-m?authuser=4&hl=en_US&proj...
proc)
  Downloading https://files.pythonhosted.org/packages/0e/5f/eeb402746a65839acdec78b7e757635f5e446138cc1d68989d12...
ba593/grpcio-1.32.0.tar.gz (20.8MB)
    100% [██████████] 20.8MB 55kB/s
  Downloading https://files.pythonhosted.org/packages/26/f8/8127fdd0294f044121d20aac7785eb810e159098447967a6103de...
df996/rsa-4.5-py2.py3-none-any.whl
Collecting pyasn1-modules>=0.2.1 (from google-auth<2.0.dev,>=1.11.0->google-cloud-storage)
  Downloading https://files.pythonhosted.org/packages/95/de/214830a981892a3e286c3794f41ae67a4495df1108c3da8a9f62159...
b9e9d/pyasn1_modules-0.2.8-py2.py3-none-any.whl (155kB)
    100% [██████████] 163kB 6.9MB/s
Collecting cachetools<5.0,>=2.0.0 (from google-auth<2.0.dev,>=1.11.0->google-cloud-storage)
  Downloading https://files.pythonhosted.org/packages/2f/a6/30b0a0bef12283e3e58cidee/b5aabc7acf4110df81a4471655d3...
3e704/cachetools-3.1.1-py2.py3-none-any.whl
Collecting crcmod<1.7,>=1.6 (from google-resumable-media<2.0.dev,>=1.1.0->google-cloud-storage)
  Downloading https://files.pythonhosted.org/packages/6b/b0/e595ce2a2527e169c3bcd633d2473c1918e0b7f6626a045ca1245d...
d4e5b/crcmod-1.7.tar.gz (89kB)
    100% [██████████] 92kB 9.9MB/s
Requirement already satisfied: enum34>=1.0.4 in /usr/lib/python2.7/dist-packages (from grpcio<2.0.dev,>=1.29.0; extr...
e == "grpc">google-api-core[grpc]<2.0.0dev,>=1.14.0->google-cloud-dataproc) (1.1.6)
Collecting pyasn1>=0.1.3 (from rsa<4.6; python_version < "3.5"->google-auth<2.0.dev,>=1.11.0->google-cloud-storage)
  Downloading https://files.pythonhosted.org/packages/62/1e/a94a8d635fa3ce4cf07f506000548d0a2447ae76fd5ca53932970fe...
3053f/pyasn1-0.4.8-py2.py3-none-any.whl (77kB)
    100% [██████████] 81kB 9.4MB/s
Building wheels for collected packages: grpcio, crcmod
  Running setup.py bdist wheel for grpcio ... done
  Stored in directory: /home/student-00-cb0469f9b15a/.cache/pip/wheels/a9/71/06/2a56cccc8cd6db7b495515b14c9008abb...
cf156f9551b0992
  Running setup.py bdist_wheel for crcmod ... done
  Stored in directory: /home/student-00-cb0469f9b15...
a0798e4aa8dac
Successfully built grpcio crcmod
Installing collected packages: protobuf, futures, p...
epis-common-protos, grpcio, google-api-core, goog...
d, google-resumable-media, google-cloud-storag...
  Running setup.py install for grpcio ... error
  error: subprocess-exited-with-error
    × grpcio setup.py install --no-binary ...
      in '/home/student-00-cb0469f9b15a/.local/bin' while
        Consider adding this directory to PATH or if you
  Successfully installed cachetools-3.1.1 crcmod-1.7
loud-core-1.4.1 google-cloud-dataproc-1.1.1 google-
e-media-1.0.0 googleapis-common-protos-1.52.0 grpci...
2020.1 rsa-4.5
student-00-cb0469f9b15a@dataproc-m:~$
```

File Transfer

simple_word_count.py Finished

File upload destination: /home/student-00-cb0469f9b15a

Fig9

5. To execute the script type the following command:

- `python simple_word_count.py -r dataproc adventures.txt`

```

student-00-cb0469f9b15a@dataproc-m ~ - Google Chrome
ssh.cloud.google.com/projects/quiklabs-gcp-00-72febf2bbf8c/zones/us-central1-f/instances/dataproc-m?authuser=4&hl=en_US&proj...
Collecting cachetools<5.0,>=2.0.0 (from google-auth<2.0dev,>=1.11.0->google-cloud-storage) ...
  Downloading https://files.pythonhosted.org/packages/2f/a6/30b0a0bef12283e83e58c1d6e7b5aab7acf7fc4110df81a4471655d3...
Collecting crcmod<1.7: python version == "2.7" (from google-resumable-media<2.0dev,>=1.0.0->google-cloud-storage) ...
  Downloading https://files.pythonhosted.org/packages/e6/b0/e595ce2a527e169c3bcd6c33d2473c1918e0b7f6826a43ca1245d...
d4e5b/crcmod-1.7.tar.gz (89kB)
  100% |██████████| 92kB 9.9MB/s
Requirement already satisfied: enum34>=1.0.4 in /usr/lib/python2.7/dist-packages (from grpcio<2.0dev,>=1.29.0; extr...
a == "grpc">=google-api-core[grpc]<2.0.0dev,>=1.14.0->google-cloud-storage) (1.1.6)
Collecting pyasn1>=0.1.3 (from rsa<4.6; python version < "3.5"->google-auth<2.0dev,>=1.11.0->google-cloud-storage) ...
  Downloading https://files.pythonhosted.org/packages/62/1e/a94a8d635fa3ce4fcf7f506003548d0a2447ae76fd5ca53932970fe...
3053f/pyasn1-0.4.8-py2.py3-none-any.whl (77kB)
  100% |██████████| 81kB 9.4MB/s
Building wheels for collected packages: grpcio, crcmod
  Running setup.py bdist_wheel for grpcio ... done
    Stored in directory: /home/student-00-cb0469f9b15a/.cache/pip/wheels/a9/71/06/2a56cccc8cd6db67b495515b44c9008abbb...
cf455f9551b0992
  Running setup.py bdist_wheel for crcmod ... done
    Stored in directory: /home/student-00-cb0469f9b15a/.cache/pip/wheels/50/24/4d/4580ca4a299f1ad6fd63443e6e584cb21e9...
a07988e4aa8daac
Successfully built grpcio crcmod
Installing collected packages: protobuf, futures, pytz, pyasn1, rsa, pyasn1-modules, cachetools, google-auth, google-...
apis-common-protos, grpcio, google-api-core, google-cloud-dataproc, google-cloud-core, google-cloud-logging, crcmo...
d, google-resumable-media, google-cloud-storage
  The scripts pyrsa-decrypt, pyrsa-encrypt, pyrsa-keygen, pyrsa-prv2pub, pyrsa-sign and pyrsa-verify are installed...
in '/home/student-00-cb0469f9b15a/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed cachetools-3.1.1 crcmod-1.7 futures-3.3.0 google-api-core-1.22.2 google-auth-1.21.1 google-...
cloud-core-1.4.1 google-cloud-dataproc-1.1.1 google-cloud-logging-1.15.1 google-cloud-storage-1.31.0 google-resum...
able-media-1.0.0 googleapis-common-protos-1.52.0 grpcio-1.32.0 protobuf-3.13.0 pyasn1-0.4.8 pyasn1-modules-0.2.8 pytz-...
2020.1 rsa-4.5
/home/student-00-cb0469f9b15a:~$ python simple_word_count.py -r dataproc adventures.txt
No configs found, falling back to auto-configuration
No config specified for dataproc
Creating temp directory /tmp/simple_word_count.student-00-cb0469f9b15a.20200914.190024.512205
writing master bootstrap script to /tmp/simple_word_count.student-00-cb0469f9b15a.20200914.190024.512205/b.sh
uploading working dir files to gs://mrjob-us-west1-l38e4801e0b63d68/tmp/simple_word_count.student-00-cb0469f9b15a.2...
0200914.190024.512205/files...
Copying other local files to gs://mrjob-us-west1-l38e4801e0b63d68/tmp/simple_word_count.student-00-cb0469f9b15a.202...
00914.190024.512205/files/
Waiting for GCS sync (eventual consistency) - sleeping 5.0 second(s)

```

Fig10

Step – 4 : To view the content

- Select Navigation menu > Storage > Browser
- And a bucket would be created containing the ‘tmp’ folder which contains files and output and the output folder contains the part files which has the output of the MRJob executed above
- And type the following command to view the data of part files

```
$hdfs dfs -cat gs://mrjob-us-west1-138e4801e0b63d68/tmp/simple_word_count.student-00-cb0469f9b15a.20200914.190024.512205/output/part-00000
```

The screenshot shows the Google Cloud Platform Storage interface. On the left, a sidebar lists 'Storage' (selected), 'Browser', 'Monitoring', 'Transfer', 'Transfer for on-premises', 'Transfer Appliance', and 'Settings'. The main area is titled 'Bucket details' for 'mrjob-us-west1-138e4801e0b63d68'. Under the 'OBJECTS' tab, it shows the contents of the 'output' directory, which contains four files: '_SUCCESS' (0 B), '_SUCC' (0 B), 'part-01' (85.9 KB), and 'part-02' (87.3 KB). The files are listed with columns for Size, Type, Created time, Storage class, Last modified, Public access, and Encryption.

Fig11

The screenshot shows a terminal window on a Windows 10 desktop. The command entered was '\$ hdfs dfs -cat gs://mrjob-us-west1-138e4801e0b63d68/tmp/simple_word_count.student-00-cb0469f9b15a.20200914.190024.512205/output/part-00000'. The output displays a large list of words and their counts, such as 'you.' (7), 'you' (3), 'you' (3), 'you\'' (16), 'young' (75), 'young.' (4), 'young.' (2), 'youngster' (2), 'youngs' (10), 'yours' (4), 'yours' (2), 'yours\'' (2), 'yourself!' (1), 'yourself?' (1), 'youth\'' (1), 'zero,' (1), 'zest' (1), and 'zigzag' (1). The terminal also shows the command 'hdfs dfs -rm -r /tmp/simple_word_count.student-00-cb0469f9b15a.20200914.190024.512205' followed by a success message.

Fig12

Lab – 12

Student Name and Roll Number: Chayan 18csu054

Semester /Section: 5/A

Link to Code: <https://github.com/chayangulati321/Big-Data>

Date: 16/09/20

Faculty Signature:

Remarks:

AIM : To execute Pig Latin queries on dataset on Gcloud using Dataproc services.

Step – 1 : Select Navigation menu > Storage > Browser, and then click Create bucket

Enter a unique name for your bucket, keep all other settings, and hit **Create**

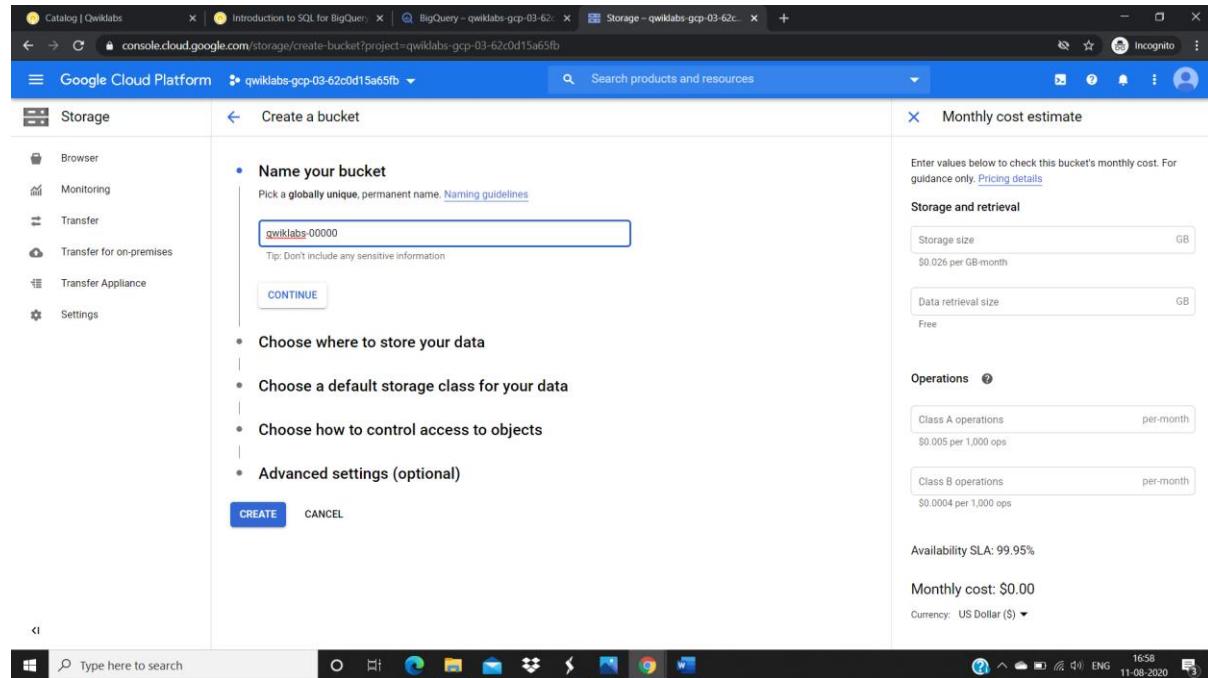


Fig1

After creating a cluster, Click on the cluster name

The screenshot shows the Google Cloud Platform interface for the Dataproc service. The left sidebar has 'Clusters' selected. The main area displays a table of clusters with one entry:

Name	Region	Zone	Total worker nodes	Scheduled deletion	Cloud Storage staging bucket
cluster2881	us-central1	us-central1-c	2	Off	dataproc-staging-us-central1-114702095928-invyr4mr

Below the table, there are tabs for 'PERMISSIONS' and 'LABELS'. A message says 'Please select at least one resource.' The status bar at the bottom indicates it's an Incognito window from 08-11-2020 at 19:27.

Fig2

Click on the SSH icon, right to the master node.

The screenshot shows the Google Cloud Platform interface for the Dataproc service, specifically for the 'cluster2881' cluster. The left sidebar has 'Clusters' selected. The main area shows cluster details and a list of VM instances.

Cluster Details:

- Name: cluster2881
- Cluster UUID: cd27792c-79f4-430a-bc42-c9755191131e
- Type: Dataproc Cluster
- Status: Running

VM Instances:

Name	Role	SSH
cluster2881-m	Master	SSH
cluster2881-w-0	Worker	
cluster2881-w-1	Worker	

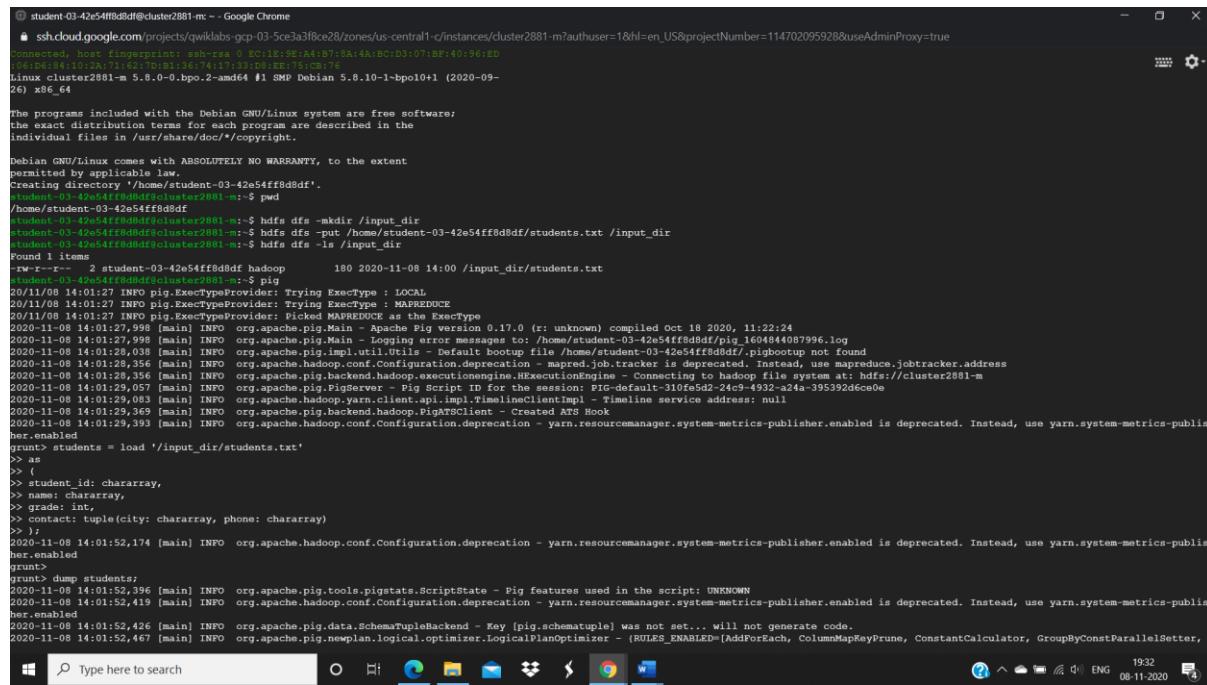
The status bar at the bottom indicates it's an Incognito window from 08-11-2020 at 19:27.

Fig3

Step – 2 : Upload the file in SSH (student.txt) and follow the following commands :-

- pwd
- hdfs dfs -mkdir /input_dir
- hdfs dfs -ls
- hdfs dfs -put <pwdPATH>/students.txt /input_dir
- students = load '/input_dir/students.txt'
as
(
student_id: chararray,
name: chararray,
grade: int,
contact: tuple(city: chararray, phone: chararray)
);

dump students;



```

student-03-42e54ff8d8df@cluster2881: ~ - Google Chrome
● ssh.cloud.google.com/projects/quicklabs-gcp-03-5ce3a3f8ce28/zones/us-central1-c/instances/cluster2881-m?authuser=1&hl=en_US&projectNumber=114702095928&useAdminProxy=true
Connected, host fingerprint: ssh-rsa 0 EC1B:9E:A4:B7:8A:4A:CD:D7:BF:40:96:ED
192.168.10.128:216237048136174117133100:BE751CB76
Linux cluster2881-m 5.8.0-0.bpo.2-amd64 #1 SMP Debian 5.8.10-1-bpo10+1 (2020-09-26) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Creating directory '/home/student-03-42e54ff8d8df'.
student-03-42e54ff8d8df@cluster2881-m:~$ pwd
/home/student-03-42e54ff8d8df
student-03-42e54ff8d8df@cluster2881-m:~$ hdfs dfs -mkdir /input_dir
student-03-42e54ff8d8df@cluster2881-m:~$ hdfs dfs -put /home/student-03-42e54ff8d8df/students.txt /input_dir
student-03-42e54ff8d8df@cluster2881-m:~$ hdfs dfs -ls /input_dir
Found 1 items
-rw-r--r-- 2 student-03-42e54ff8d8df hadoop 180 2020-11-08 14:00 /input_dir/students.txt
student-03-42e54ff8d8df@cluster2881-m:~$ pig
20/11/08 14:01:27 INFO pig.ExeTypeProvider: Trying ExeType : LOCAL
20/11/08 14:01:27 INFO pig.ExeTypeProvider: Trying ExeType : MAPREDUCE
20/11/08 14:01:27 INFO pig.ExeTypeProvider: Trying ExeType : HDFS
20/11/08 14:01:27 INFO org.apache.pig.Main : Apache Pig version 0.17.0 (r: unknown) compiled Oct 18 2020, 11:22:24
20/11/08 14:01:27,998 [main] INFO org.apache.pig.Main : Logging error messages to: /home/student-03-42e54ff8d8df/pig_1604844007996.log
20/11/08 14:01:28,038 [main] INFO org.apache.pig.impl.util.Utils : Default bootstrap file /home/student-03-42e54ff8d8df/.pigbootstrap not found
20/11/08 14:01:28,356 [main] INFO org.apache.hadoop.conf.Configuration.deprecation : mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
20/11/08 14:01:28,356 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine : Connecting to hadoop file system at: hdfs://cluster2881-m
20/11/08 14:01:28,356 [main] INFO org.apache.pig.PigServer : Pig Script for the session PIG-default-310fe5d2-24c9-4932-a24a-395392d6ce0e
20/11/08 14:01:29,083 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReducePlanner : Timeline client: null
20/11/08 14:01:29,369 [main] INFO org.apache.pig.backend.hadoop.PigStorageClient : Created ATS Hook
20/11/08 14:01:29,393 [main] INFO org.apache.hadoop.conf.Configuration.deprecation : yarn.resourcemanager.systems-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-public.enabled
grunt> students = load '/input_dir/students.txt'
>> as
>> student_id: chararray,
>> name: chararray,
>> grade: int,
>> contact: tuple(city: chararray, phone: chararray)
>> ;
20/11/08 14:01:52,174 [main] INFO org.apache.hadoop.conf.Configuration.deprecation : yarn.resourcemanager.systems-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-public.enabled
grunt> dump students;
20/11/08 14:01:52,396 [main] INFO org.apache.pig.tools.pigstats.ScriptState : Pig features used in the script: UNKNOWN
20/11/08 14:01:52,419 [main] INFO org.apache.hadoop.conf.Configuration.deprecation : yarn.resourcemanager.systems-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-public.enabled
20/11/08 14:01:52,426 [main] INFO org.apache.pig.data.SchemaTupleHackend : Key {pig.schematuple} was not set... will not generate code.
20/11/08 14:01:52,467 [main] INFO org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer : {RULES_ENABLED=(AddForEach, ColumnMapKeyPrune, ConstantCalculator, GroupByConstParallelSetter,

```

Fig4

- student_info = foreach students generate \$1, \$3.\$0, \$3.\$1;
- dump student_info;

```

student-03-42e54ff8d@cluster2881-m: ~ - Google Chrome
ssh.cloud.google.com/projects/quickstarts-gcp-03-5ce3a3f8ce28/zones/us-central1-c/instances/cluster2881-m?authuser=1&hl=en_US&projectNumber=114702095928&useAdminProxy=true
2020-11-08 14:04:04,821 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 100% complete
2020-11-08 14:04:04,823 [main] INFO org.apache.pig.tools.pigstats.mapreduce.SimplePigStats - Script Statistics:
HadoopVersion PigVersion Userid StartedAt FinishedAt Features
2.9.2 0.17.0 student-03-42e54ff8d@cluster2881-m 2020-11-08 14:03:48 2020-11-08 14:04:04 UNKNOWN
Success!
Job Stats (time in seconds):
JobId Maps Reduces MaxMapTime MinMapTime AvgMapTime MedianMapTime MaxReduceTime MinReduceTime AvgReduceTime MedianReduceTime Alias Feature Outputs
job_1604843773818_0002 1 0 2 2 0 0 0 0 0 MAP_ONLY hdfs://cluster2881-m/tmp/temp-67398674/tmp-1733889417,
Input(s):
Successfully read 5 records (550 bytes) from: "/input_dir/students.txt"
Output(s):
Successfully stored 5 records (235 bytes) in: "hdfs://cluster2881-m/tmp/temp-67398674/tmp-1733889417"
Counters:
Total records written : 5
Total bytes written : 235
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
Job DAG:
job_1604843773818_0002

2020-11-08 14:04:04,825 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at cluster2881-m/10.128.0.3:8032
2020-11-08 14:04:04,825 [main] INFO org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at cluster2881-m/10.128.0.3:10200
2020-11-08 14:04:04,829 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2020-11-08 14:04:04,831 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at cluster2881-m/10.128.0.3:8032
2020-11-08 14:04:04,847 [main] INFO org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at cluster2881-m/10.128.0.3:10200
2020-11-08 14:04:04,850 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2020-11-08 14:04:04,852 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at cluster2881-m/10.128.0.3:8032
2020-11-08 14:04:04,852 [main] INFO org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at cluster2881-m/10.128.0.3:10200
2020-11-08 14:04:04,853 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2020-11-08 14:04:04,853 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at cluster2881-m/10.128.0.3:8032
2020-11-08 14:04:04,854 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2020-11-08 14:04:04,854 [main] INFO org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at cluster2881-m/10.128.0.3:10200
2020-11-08 14:04:04,855 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Configuration - yarn.resourcemanager.systems-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-public.enabled
2020-11-08 14:04:04,900 [main] INFO org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2020-11-08 14:04:04,921 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2020-11-08 14:04:04,921 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapReduceUtil - Total input paths to process : 1
{("s1234", "John", 8, {"Seattle", 2068541229}), ("s1234", "Emily", 8, {"Samannah", 4253445435}), ("s1234", "Nina", 8, {"Kirkland", 2523216437}), ("s1234", "Mike", 8, {"Issaquah", 4254366721}), ("s1234", "Nita", 8, {"Redmond", 4253268720})
grunt> student_info = foreach students generate $1, $3.$0, $3.$1;

```

Fig5

```

student-03-42e54ff8d@cluster2881-m: ~ - Google Chrome
ssh.cloud.google.com/projects/quickstarts-gcp-03-5ce3a3f8ce28/zones/us-central1-c/instances/cluster2881-m?authuser=1&hl=en_US&projectNumber=114702095928&useAdminProxy=true
2020-11-08 14:04:50,639 [main] INFO org.apache.pig.tools.pigstats.mapreduce.SimplePigStats - Script Statistics:
HadoopVersion PigVersion Userid StartedAt FinishedAt Features
2.9.2 0.17.0 student-03-42e54ff8d@cluster2881-m 2020-11-08 14:04:34 2020-11-08 14:04:50 UNKNOWN
Success!
Job Stats (time in seconds):
JobId Maps Reduces MaxMapTime MinMapTime AvgMapTime MedianMapTime MaxReduceTime MinReduceTime AvgReduceTime MedianReduceTime Alias Feature Outputs
job_1604843773818_0003 1 0 2 2 2 2 0 0 0 student_info,students MAP_ONLY hdfs://cluster2881-m/tmp/temp-67398674/tmp-1557620279,
Input(s):
Successfully read 5 records (550 bytes) from: "/input_dir/students.txt"
Output(s):
Successfully stored 5 records (180 bytes) in: "hdfs://cluster2881-m/tmp/temp-67398674/tmp-1557620279"
Counters:
Total records written : 5
Total bytes written : 180
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
Job DAG:
job_1604843773818_0003

2020-11-08 14:04:50,642 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at cluster2881-m/10.128.0.3:8032
2020-11-08 14:04:50,642 [main] INFO org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at cluster2881-m/10.128.0.3:10200
2020-11-08 14:04:50,646 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2020-11-08 14:04:50,669 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at cluster2881-m/10.128.0.3:8032
2020-11-08 14:04:50,669 [main] INFO org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at cluster2881-m/10.128.0.3:10200
2020-11-08 14:04:50,674 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2020-11-08 14:04:50,680 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at cluster2881-m/10.128.0.3:8032
2020-11-08 14:04:50,689 [main] INFO org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at cluster2881-m/10.128.0.3:10200
2020-11-08 14:04:50,693 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2020-11-08 14:04:50,710 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2020-11-08 14:04:50,711 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.systems-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-public.enabled
2020-11-08 14:04:50,711 [main] INFO org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2020-11-08 14:04:50,719 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2020-11-08 14:04:50,719 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapReduceUtil - Total input paths to process : 1
{("John", "Seattle", 2068541229), ("Emily", "Samannah", 4253445435), ("Nina", "Kirkland", 2523216437), ("Mike", "Issaquah", 4254366721), ("Nita", "Redmond", 4253268720)}
grunt> 

```

Fig6

- `students_no_schema = load '/input_dir/students.txt'`
- `dump students_no_schema;`
- `students_no_schema_info = foreach students generate $1, $3;`
- `dump students_no_schema_info;`

```

student-03-42e54ff8d@cluster2881-m: ~ - Google Chrome
ssh.cloud.google.com/projects/quickstarts-gcp-03-5ce3a3f8ce28/zones/us-central1-c/instances/cluster2881-m?authuser=1&hl=en_US&projectNumber=114702095928&useAdminProxy=true
2.9.2 0.17.0 student-03-42e54ff8d@cluster2881-m 2020-11-08 14:04:34 2020-11-08 14:04:50 UNKNOWN
Success!
Job Stats (time in seconds):
JobId Maps Reduces MaxMapTime MinMapTime AvgMapTime MedianMapTime MaxReduceTime MinReduceTime AvgReduceTime MedianReduceTime Alias Feature Outputs
job_1604843773818_0003 1 0 2 2 0 0 0 0 0 student_info,students MAP_ONLY hdfs://cluster2881-m/tmp/temp-67398674/tmp-1557620279

Input(s):
Successfully read 5 records (550 bytes) from: "/input_dir/students.txt"

Output(s):
Successfully stored 5 records (180 bytes) in: "hdfs://cluster2881-m/tmp/temp-67398674/tmp-1557620279"

Counters:
Total records written : 5
Total bytes written : 180
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1604843773818_0003

2020-11-08 14:04:50,642 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at cluster2881-m/10.128.0.3:8032
2020-11-08 14:04:50,642 [main] INFO org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at cluster2881-m/10.128.0.3:10200
2020-11-08 14:04:50,646 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2020-11-08 14:04:50,669 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at cluster2881-m/10.128.0.3:8032
2020-11-08 14:04:50,674 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2020-11-08 14:04:50,689 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at cluster2881-m/10.128.0.3:8032
2020-11-08 14:04:50,693 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2020-11-08 14:04:50,711 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2020-11-08 14:04:50,711 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - Key (pig.schematuple) was not set... will not generate code.
2020-11-08 14:04:50,719 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2020-11-08 14:04:50,719 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(John,Seattle, 2068541229)
(Emily,Sammamish, 4253445435)
(Nina,Kirkland, 2532316437)
(Mike,Issaquah, 4254366721)
(Nita,Redmond, 4253268720)
grants> foreach students generate TOTUPLE($0, $1, $2, $3);
grants> dump student_info bag;
2020-11-08 14:06:42,702 [main] INFO org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: UNKNOWN
2020-11-08 14:06:42,713 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled

```

Fig7

```

student-03-42e54ff8d@cluster2881-m: ~ - Google Chrome
ssh.cloud.google.com/projects/quickstarts-gcp-03-5ce3a3f8ce28/zones/us-central1-c/instances/cluster2881-m?authuser=1&hl=en_US&projectNumber=114702095928&useAdminProxy=true
2020-11-08 14:06:58,681 [main] INFO org.apache.pig.tools.pigstats.SimplePigStats - Script Statistics:
HadoopVersion PigVersion Userid StartedAt FinishedAt Features
2.9.2 0.17.0 student-03-42e54ff8d@cluster2881-m 2020-11-08 14:06:42 2020-11-08 14:06:58 UNKNOWN
Success!
Job Stats (time in seconds):
JobId Maps Reduces MaxMapTime MinMapTime AvgMapTime MedianMapTime MaxReduceTime MinReduceTime AvgReduceTime MedianReduceTime Alias Feature Outputs
job_1604843773818_0004 1 0 2 2 0 0 0 0 0 student_info_bag,students MAP_ONLY hdfs://cluster2881-m/tmp/temp-67398674/tmp-105739191

Input(s):
Successfully read 5 records (550 bytes) from: "/input_dir/students.txt"

Output(s):
Successfully stored 5 records (240 bytes) in: "hdfs://cluster2881-m/tmp/temp-67398674/tmp-105739191"

Counters:
Total records written : 5
Total bytes written : 240
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1604843773818_0004

2020-11-08 14:06:58,682 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at cluster2881-m/10.128.0.3:8032
2020-11-08 14:06:58,683 [main] INFO org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at cluster2881-m/10.128.0.3:10200
2020-11-08 14:06:58,686 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2020-11-08 14:06:58,704 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at cluster2881-m/10.128.0.3:8032
2020-11-08 14:06:58,707 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2020-11-08 14:06:58,721 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at cluster2881-m/10.128.0.3:8032
2020-11-08 14:06:58,722 [main] INFO org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at cluster2881-m/10.128.0.3:10200
2020-11-08 14:06:58,725 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2020-11-08 14:06:58,744 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapreduceLayer.MapReduceLauncher - Success!
2020-11-08 14:06:58,745 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2020-11-08 14:06:58,746 [main] INFO org.apache.pig.data.SchemaTupleBackend - Key (pig.schematuple) was not set... will not generate code.
2020-11-08 14:06:58,757 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2020-11-08 14:06:58,757 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
((s1234,John,8),(Seattle, 2068541229))
((s1245,Emily,8),(Sammamish, 4253445435))
((s1234,Nina,8),(Kirkland, 2532316437))
((s1234,Mike,8),(Issaquah, 4254366721))
((s1254,Nita,8),(Redmond, 4253268720))
grants> 

```

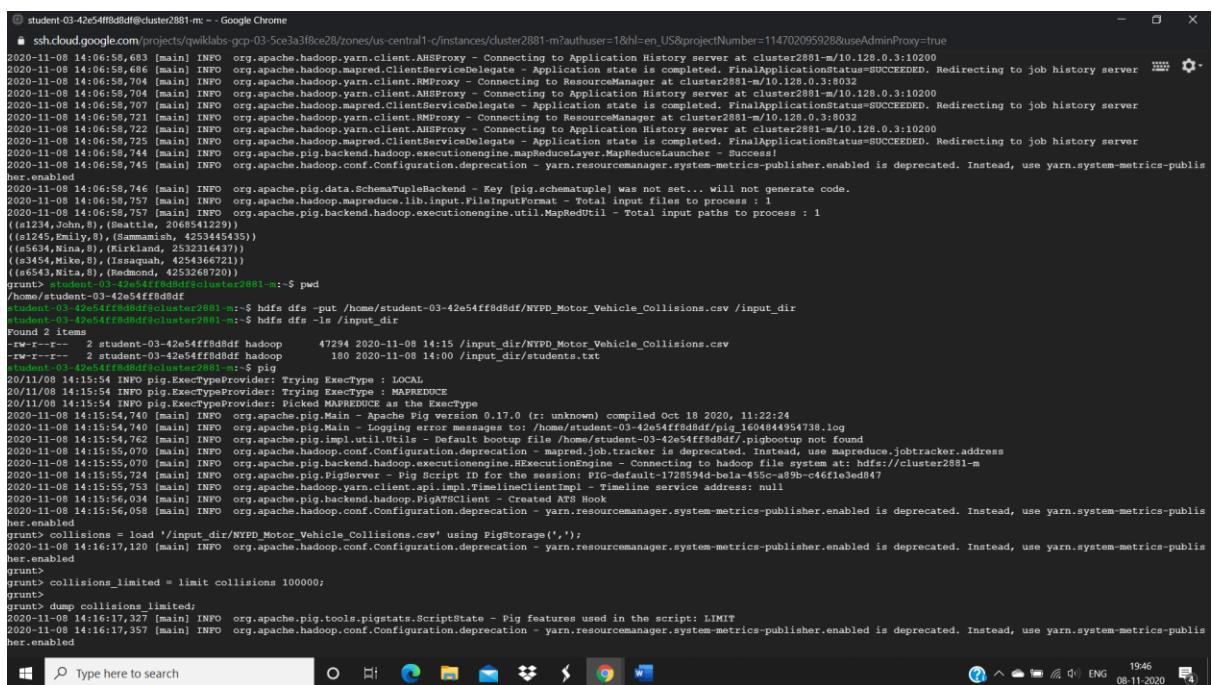
Fig8

Step – 3 : Upload the file in SSH (NYPD_Motor_Vehicle_Collisions.csv) and follow the following commands :-

- pwd
- hdfs dfs -ls
- hdfs dfs -put <pwdPATH>/students.txt /input_dir
- hdfs dfs -ls /input_dir
- collisions = load '/input_dir/NYPD_Motor_Vehicle_Collisions.csv' using PigStorage(',');

collisions_limited = limit collisions 100000;

dump collisions_limited;



```

student-03-42e54ff8d8df@cluster2881-m: ~ - Google Chrome
ssh.cloud.google.com/projects/qlkllabs-gcp-03-5ce3a3f8e28/zones/us-central1-c/instances/cluster2881-m@authuser-1&hl=en_US&projectNumber=114702095928&useAdminProxy=true
2020-11-08 14:06:58,683 [main] INFO org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at cluster2881-m/10.128.0.3:10200
2020-11-08 14:06:58,683 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2020-11-08 14:06:58,704 [main] INFO org.apache.hadoop.yarn.client.AHSProxy - Connecting to ResourceManager at cluster2881-m/10.128.0.3:8032
2020-11-08 14:06:58,704 [main] INFO org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at cluster2881-m/10.128.0.3:10200
2020-11-08 14:06:58,707 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2020-11-08 14:06:58,721 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at cluster2881-m/10.128.0.3:8032
2020-11-08 14:06:58,722 [main] INFO org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at cluster2881-m/10.128.0.3:10200
2020-11-08 14:06:58,725 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2020-11-08 14:06:58,744 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2020-11-08 14:06:58,745 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2020-11-08 14:06:58,746 [main] INFO org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2020-11-08 14:06:58,757 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2020-11-08 14:06:58,757 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapReduceUtil - Total input paths to process : 1
((s12345,John,8), (Seattle, 206954193))
((s12345,Emily,8), (Seattle, 206954193))
((s12345,Nina,8), (Kirkland, 2532316437))
((s12345,Mike,8), (Issaquah, 4254366721))
((s12345,Nita,8), (Redmond, 425326720))
grunt> student-03-42e54ff8d8df@cluster2881-m:~$ pwd
/home/student-03-42e54ff8d8df
student-03-42e54ff8d8df@cluster2881-m:~$ hdfs dfs -put /home/student-03-42e54ff8d8df/NYPD_Motor_Vehicle_Collisions.csv /input_dir
student-03-42e54ff8d8df@cluster2881-m:~$ hdfs dfs -ls /input_dir
Found 2 items
-rw-r--r-- 2 student-03-42e54ff8d8df hadoop 47294 2020-11-08 14:15 /input_dir/NYPD_Motor_Vehicle_Collisions.csv
-rw-r--r-- 2 student-03-42e54ff8d8df hadoop 180 2020-11-08 14:00 /input_dir/students.txt
student-03-42e54ff8d8df@cluster2881-m:~$ pig
20/11/08 14:15:54 INFO pig.ExcuteTypeProvider: Trying ExcuteType : LOCAL
20/11/08 14:15:54 INFO pig.ExcuteTypeProvider: Trying ExcuteType : MAPREDUCE
20/11/08 14:15:54 INFO org.apache.pig.impl.mapreduce.PigMapReduce as the ExcuteType
2020-11-08 14:15:54,740 [main] INFO org.apache.pig.Main - Apache Pig version 0.17.0 (r: unknown) compiled Oct 18 2020, 11:22:24
2020-11-08 14:15:54,740 [main] INFO org.apache.pig.Main - Logging error messages to: /home/student-03-42e54ff8d8df/pig_1604844954738.log
2020-11-08 14:15:54,762 [main] INFO org.apache.pig.Impl.Util - Default bootstrap file /home/student-03-42e54ff8d8df/.pigbootup not found
2020-11-08 14:15:55,070 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2020-11-08 14:15:55,070 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker.address=cluster2881-m:8032
2020-11-08 14:15:55,753 [main] INFO org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Created ATS Hook
2020-11-08 14:15:55,034 [main] INFO org.apache.pig.backend.hadoop.PigTSCClient - Timeline service address: null
2020-11-08 14:15:56,058 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2020-11-08 14:15:56,058 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
grunt> collisions_limited = limit collisions 100000;
grunt> dump collisions_limited;
2020-11-08 14:16:17,327 [main] INFO org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: LIMIT
2020-11-08 14:16:17,357 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled

```

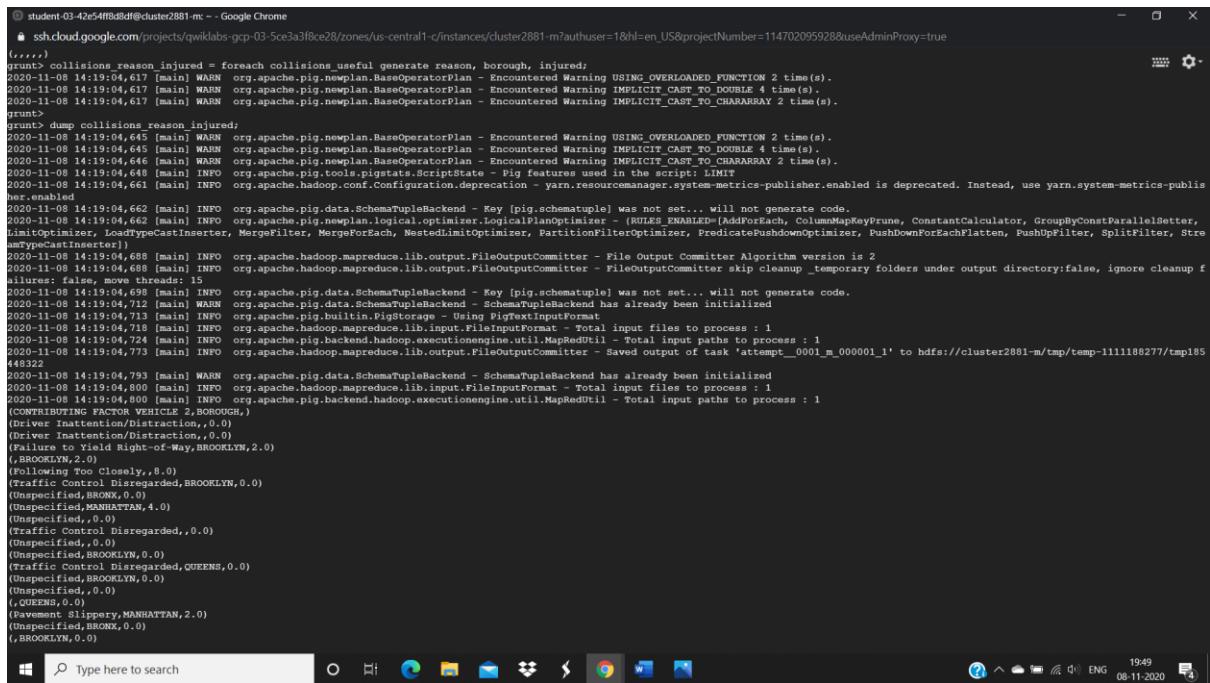
Fig9

Fig10

- `collisions_header = limit collisions_limited 1;`
 - `dump collisions_header;`
 - `collisions_useful = foreach collisions_limited generate $0 as date, $2 as borough, $3 as zipcode, TRIM($8) as location, $11 + $13 + $15 + $17 as injured, TRIM($19) as reason;`
 - `dump collisions_useful;`

Fig11

- `collisions_reason_injured = foreach collisions_useful generate reason, borough, injured;`
- `dump collisions_reason_injured;`



```
student-03-42e54ff8d@cluster2881-m: ~ - Google Chrome
ssh.cloud.google.com/projects/qwiklabs-gcp-03-5ce3a3f8ce28/zones/us-central1-c/instances/cluster2881-m?authuser=1&hl=en_US&projectNumber=114702095928&useAdminProxy=true
(.,.,.)
grunt> collisions reason injured = foreach collisions_useful generate reason, borough, injured;
2020-11-08 14:19:04,617 [main] WARN org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning USING_OVERLOADED_FUNCTION 2 time(s).
2020-11-08 14:19:04,617 [main] WARN org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 4 time(s).
2020-11-08 14:19:04,617 [main] WARN org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_CHARARRAY 2 time(s).
parent>
parent> grunt> dump collisions reason injured;
2020-11-08 14:19:04,645 [main] WARN org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning USING_OVERLOADED_FUNCTION 2 time(s).
2020-11-08 14:19:04,645 [main] WARN org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 4 time(s).
2020-11-08 14:19:04,646 [main] WARN org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_CHARARRAY 2 time(s).
2020-11-08 14:19:04,648 [main] INFO org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: LINIT
2020-11-08 14:19:04,651 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-public.enabled
2020-11-08 14:19:04,662 [main] INFO org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple1] was not set... will not generate code.
2020-11-08 14:19:04,662 [main] INFO org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - (RULES ENABLED=AddForEach, ColumnMapKeyPrune, ConstantCalculator, GroupByConstParallelSetter, LimitOptimizer, LoadTypeCastInserter, MergeForEach, NestedLimitOptimizer, PartitionFilterOptimizer, PredicatePushdownOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StringTypeCastInserter)
2020-11-08 14:19:04,686 [main] INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter - File Output Committer Algorithm version is 2
2020-11-08 14:19:04,688 [main] INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter - skip cleanup _temporary folders under output directory:false, ignore cleanup failure: false, move threads: 15
2020-11-08 14:19:04,698 [main] INFO org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple1] was not set... will not generate code.
2020-11-08 14:19:04,712 [main] WARN org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2020-11-08 14:19:04,713 [main] INFO org.apache.pig.builtin.PigStorage - Using PigTextInputFormat
2020-11-08 14:19:04,718 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapredUtil - Total input files to process : 1
2020-11-08 14:19:04,724 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapredUtil - Total input paths to process : 1
2020-11-08 14:19:04,773 [main] INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter - Saved output of task 'attempt_0001_m_000001_1' to hdfs://cluster2881-m/tmp/temp-1111180277/tmp18544832
2020-11-08 14:19:04,793 [main] WARN org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2020-11-08 14:19:04,800 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2020-11-08 14:19:04,800 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapredUtil - Total input paths to process : 1
(ATTRIBUTING FACTURE VEHICLE 2,BOROUGH,
(Driver Inattention/Distracted,0.0)
(Driver Inattentive/Distracted,0.0)
(Failure to Yield Right of-Way,BROOKLYN,2.0)
(,BROOKLYN,2.0)
(Following Too Closely,,0.0)
(Traffic Control Disregarded,BROOKLYN,0.0)
(Unspecified,MANHATTAN,0.0)
(Unspecified,MANHATTAN,4.0)
(Unspecified,,0.0)
(Traffic Control Disregarded,,0.0)
(Unspecified,BROOKLYN,0.0)
(Traffic Control Disregarded,QUEENS,0.0)
(Unspecified,BROOKLYN,0.0)
(Unspecified,,0.0)
(,QUEENS,0.0)
(Pavement Slippery,MANHATTAN,2.0)
(Unspecified,BRONX,0.0)
(,BROOKLYN,0.0)
```

Fig12

Lab – 13

Student Name and Roll Number: Chayan 18csu054

Semester /Section: 5/A

Link to Code: <https://github.com/chayangulati321/Big-Data>

Date: 23/09/20

Faculty Signature:

Remarks:

AIM : To execute Hive queries on item details data on Gcloud using Dataproc services.

Step – 1 : Select Navigation menu > Storage > Browser, and then click Create bucket

Enter a unique name for your bucket, keep all other settings, and hit **Create**

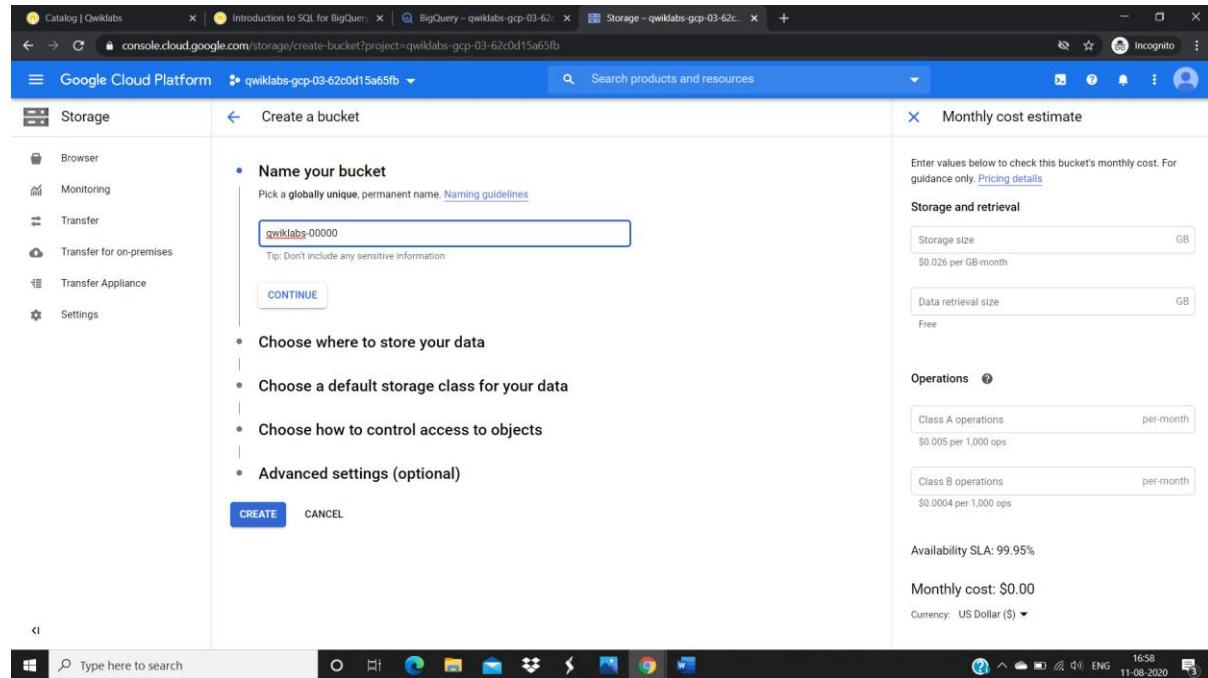


Fig1

After creating a cluster, Click on the cluster name

The screenshot shows the Google Cloud Platform interface for the Dataproc service. The left sidebar has 'Clusters' selected. The main area displays a table of clusters with one entry:

Name	Region	Zone	Total worker nodes	Scheduled deletion	Cloud Storage staging bucket
cluster2881	us-central1	us-central1-c	2	Off	dataproc-staging-us-central1-114702095928-invyr4mr

The right panel shows 'No clusters selected' and a message: 'Please select at least one resource.' Below the table, there are tabs for 'PERMISSIONS' and 'LABELS'. The status bar at the bottom indicates it's an Incognito window from 08-11-2020 at 19:27.

Fig2

Click on the SSH icon, right to the master node.

The screenshot shows the Google Cloud Platform interface for the Dataproc service, specifically for the 'cluster2881' cluster. The left sidebar has 'Clusters' selected. The main area shows cluster details and a table of VM instances.

Cluster Details:

- Name: cluster2881
- Cluster UUID: cd27792c-79f4-430a-bc42-c9755191131e
- Type: Dataproc Cluster
- Status: Running

VM Instances:

Name	Role
cluster2881-m	Master
cluster2881-w-0	Worker
cluster2881-w-1	Worker

The status bar at the bottom indicates it's an Incognito window from 08-11-2020 at 19:27.

Fig3

Step – 2 : Upload the file in SSH (emp.txt) and follow the following commands :-

- pwd
- hdfs dfs -mkdir /input_dir
- hdfs dfs -ls
- hdfs dfs -put <pwdPATH>/emp.txt /input_dir
- CREATE TABLE employee (
 name string,
 work_place ARRAY<string>,
 gender_age STRUCT<gender:string,age:int>,
 skills_score MAP<string,int>,
 depart_title MAP<STRING,ARRAY<STRING>>
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
COLLECTION ITEMS TERMINATED BY ','
MAP KEYS TERMINATED BY ':'
STORED AS TEXTFILE;
- Load data
- LOAD DATA INPATH 'gs://pig-bucket1/emp.txt' OVERWRITE INTO TABLE employee;
- Query the whole table
- SELECT * FROM employee;

```
student-03-42e54ff8d8df@cluster2881-m: ~ - Google Chrome
sh.cloud.google.com/projects/qwiklabs-gcp-03-5ce3a3fce28/zones/us-central1-c/instances/cluster2881-m?authuser=1&hl=en_US&projectNumber=114702095928&useAdminProxy=true
(/, )
student-03-42e54ff8d8df@cluster2881-m:~$ pwd
/home/student-03-42e54ff8d8df
student-03-42e54ff8d8df@cluster2881-m:~$ hdfs dfs -put /home/student-03-42e54ff8d8df/emp.txt /input_dir
put: '/home/student-03-42e54ff8d8df/emp.txt': No such file or directory
student-03-42e54ff8d8df@cluster2881-m:~$ hdfs dfs -put /home/student-03-42e54ff8d8df/emp.txt /input_dir
student-03-42e54ff8d8df@cluster2881-m:~$ hdfs dfs -ls /input_dir
Found 3 items
-rw-r--r-- 2 student-03-42e54ff8d8df hadoop 47294 2020-11-08 14:15 /input_dir/NYPD_Motor_Vehicle_Collisions.csv
-rw-r--r-- 2 student-03-42e54ff8d8df hadoop 234 2020-11-08 14:24 /input_dir/emp.txt
-rw-r--r-- 2 student-03-42e54ff8d8df hadoop 180 2020-11-08 14:00 /input_dir/students.txt
student-03-42e54ff8d8df@cluster2881-m:~$ hive
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async:
true
hive> CREATE TABLE employee (
    >     name string,
    >     work_place ARRAY<string>,
    >     gender_age STRUCT<gender:string,age:int>,
    >     skills_score MAP<string,int>,
    >     depart_title MAP<STRING,ARRAY<STRING>>
    > )
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
COLLECTION ITEMS TERMINATED BY ','  

MAP KEYS TERMINATED BY ':'
STORED AS TEXTFILE;
OK
Time taken: 1.202 seconds
hive> LOAD DATA INPATH '/home/student-03-42e54ff8d8df/input_dir/emp.txt' OVERWRITE INTO TABLE employee;
FAILED: SemanticException line 1:17 Invalid path ''/home/student-03-42e54ff8d8df/input_dir/emp.txt
!': No files matching path hdfs://cluster2881-m:/home/student-03-42e54ff8d8df/input_dir/emp.txt
hive> LOAD DATA INPATH '/input_dir/emp.txt' OVERWRITE INTO TABLE employee;
Loading data to table default.employee
OK
Time taken: 0.62 seconds
hive> SELECT * FROM employee;
OK
Maddy ["Montreal","Toronto"] {"gender":"Male","age":30} {"DB":80} {"Product":["Developer","Lead"]}
Will ["Montreal"] {"gender":"Male","age":35} {"Perl":85} {"Product":["Lead"],"Test":1}
Shelley ["New York"] {"gender":"Female","age":27} {"Python":80} {"Test":("Lead"),"COE":("Architect")}
Lucy ["Vancouver"] {"gender":"Female","age":57} {"Sales":89,"HR":94} {"Sales":("Lead")}

Windows Type here to search 0:11:2020 19:58 ENG
```

Fig4

--Query the ARRAY in the table

- SELECT work_place FROM employee;
- SELECT work_place[0] AS col_1, work_place[1] AS col_2, work_place[2] AS col_3
FROM employee;

--Query the STRUCT in the table

- SELECT gender_age FROM employee;
- SELECT gender_age.gender, gender_age.age FROM employee;

--Query the MAP in the table

- SELECT skills_score FROM employee;

The screenshot shows a Windows desktop environment. At the top, there's a taskbar with icons for File Explorer, Task View, Start, Edge browser, File Explorer, Task View, and Task View again. The system tray shows the date (08-11-2020), time (19:58), battery level (ENG), and signal strength. A OneDrive notification is visible in the bottom right corner, stating "Screenshot saved. The screenshot was added to your OneDrive." The main window is a terminal session titled "student-03-42e54f8d@cluster2881:m ~ - Google Chrome". It displays several Hive queries and their results:

```
ssh.cloud.google.com/projects/qwiklabs-gcp-03-5ce3a3f8ce28/zones/us-central1-c/instances/cluster2881-m?authuser=1&hl=en_US&projectNumber=114702095928&useAdminProxy=true
hive> SELECT work_place FROM employee;
OK
+-----+
| [Vancouver,"Female",57] | {"Sales":89,"HR":94} | ["Sales":["Lead"]]
+-----+
Time taken: 0.601 seconds, Fetched: 6 row(s)
hive> SELECT work_place[0] AS col_1, work_place[1] AS col_2, work_place[2] AS col_3 FROM employee;
OK
+-----+
| ["Montreal","Toronto"] | ["Montreal"] | ["New York"]
| ["Montreal"] | ["Montreal"] | ["Vancouver"]
| ["Vancouver"] | ["Vancouver"] | [NULL]
+-----+
Time taken: 0.19 seconds, Fetched: 6 row(s)
hive> SELECT work_place[0] AS col_1, work_place[1] AS col_2, work_place[2] AS col_3 FROM employee;
OK
+-----+
| Montreal | Toronto | NULL
| Montreal | NULL | NULL
| New York | NULL | NULL
| Vancouver | NULL | NULL
| NULL | NULL | NULL
| NULL | NULL | NULL
+-----+
Time taken: 0.434 seconds, Fetched: 6 row(s)
hive> SELECT gender_age FROM employee;
OK
+-----+
| {"gender": "Male", "age": 30} |
| {"gender": "Male", "age": 35} |
| {"gender": "Female", "age": 27} |
| {"gender": "Female", "age": 57} |
+-----+
Time taken: 0.195 seconds, Fetched: 6 row(s)
hive> SELECT gender.gender, gender.age FROM employee;
OK
+-----+
| Male   30 |
| Male   35 |
| Female 27 |
| Female 57 |
+-----+
Time taken: 0.184 seconds, Fetched: 6 row(s)
hive> SELECT skills_score FROM employee;
OK
+-----+
| {"DB":80} |
| {"Perl":85} |
| {"Python":80} |
| {"Sales":89,"HR":94} |
+-----+
Time taken: 0.184 seconds, Fetched: 6 row(s)
```

Fig5

- SELECT name, skills_score['DB'] AS DB,
- skills_score['Perl'] AS Perl, skills_score['Python'] AS Python,
- skills_score['Sales'] as Sales, skills_score['HR'] as HR FROM employee;
- SELECT depart_title FROM employee;
- SELECT name, depart_title['Product'] AS Product, depart_title['Test'] AS Test,
- depart_title['COE'] AS COE, depart_title['Sales'] AS Sales
- FROM employee;
- SELECT name,
- depart_title['Product'][0] AS product_col0,

- depart_title['Test'][0] AS test_col0
- FROM employee;

```

student-03-42e54ff8d@cluster2881: ~ - Google Chrome
sh.cloud.google.com/projects/qwiklabs-gcp-03-5ce3a3f8ce28/zones/us-central1-c/instances/cluster2881-m?authuser=1&hl=en_US&projectNumber=114702095928&useAdminProxy=true
("python":80)
("Sales":89,"HR":94)
NULL
NULL
Time taken: 0.179 seconds, Fetched: 6 row(s)
hive> SELECT name, skills_score['DB'] AS DB,
    > skills_score['Perl'] AS Perl, skills_score['Python'] AS Python,
    > skills_score['Sales'] as Sales, skills_score['HR'] as HR FROM employee;
OK
Maddy      80      NULL      NULL      NULL      NULL
Will       NULL     85      NULL      NULL      NULL
Shelley    NULL     80      NULL      NULL      NULL
Lucy       NULL     NULL     90      95      NULL
NULL      NULL     NULL     NULL      NULL      NULL
NULL      NULL     NULL     NULL      NULL      NULL
Time taken: 0.208 seconds, Fetched: 6 row(s)
hive> 
    > SELECT depart_title FROM employee;
OK
("product": "Developer"|"Lead")
("product": "Lead", "Test": "Lead")
("Test": "Lead", "COE": "Architect")
("Sales": "Lead")
NULL
NULL
Time taken: 0.18 seconds, Fetched: 6 row(s)
hive> SELECT name, depart_title['Product'] AS Product, depart_title['Test'] AS Test,
    > depart_title['COE'] AS COE, depart_title['Sales'] AS Sales
    > FROM employee;
OK
Maddy      ["Developer"|"Lead"]      NULL      NULL      NULL
Will       ["Lead"]      ["Lead"]      NULL      NULL
Shelley    NULL      ["Lead"]      ["Architect"]      NULL
Lucy       NULL      NULL      ["Lead"]
NULL      NULL      NULL      NULL
NULL      NULL      NULL      NULL
Time taken: 0.177 seconds, Fetched: 6 row(s)
hive> 
    > SELECT name,
    > depart_title['Product'][0] AS product_col0,
    > depart_title['Test'][0] AS test_col0
    > FROM employee;
OK
Maddy      Developer"|"Lead NULL
Will       Lead      Lead
Shelley    NULL      Lead
Lucy       NULL      NULL
NULL      NULL      NULL
NULL      NULL      NULL
Time taken: 0.179 seconds, Fetched: 6 row(s)
hive> 

```

Fig6

Lab – 14

Student Name and Roll Number: Chayan 18csu054

Semester /Section: 5/A

Link to Code: <https://github.com/chayangulati321/Big-Data>

Date: 07/10/20

Faculty Signature:

Remarks:

AIM : To execute Spark program (word count) on GCP using pyspark.

Step – 1 : Select Navigation menu > Storage > Browser, and then click Create bucket

Enter a unique name for your bucket, keep all other settings, and hit **Create**

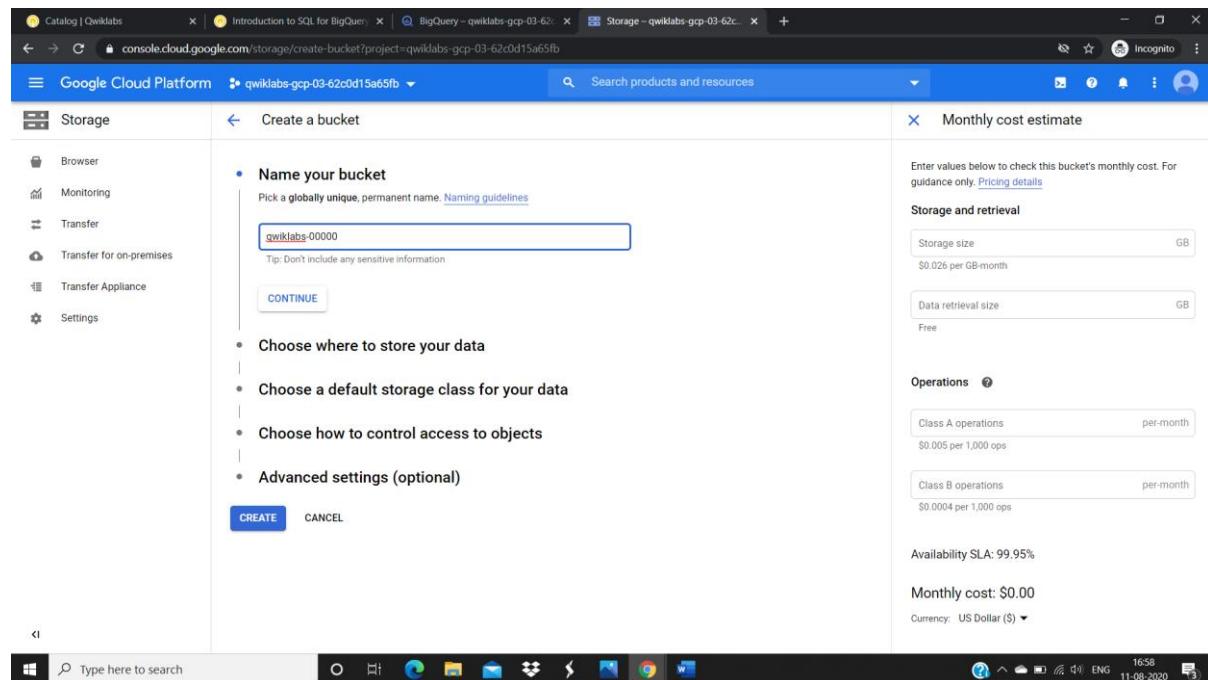


Fig1

After creating a cluster, Click on the cluster name

The screenshot shows the Google Cloud Platform interface for the DataProc service. In the left sidebar, under the 'Clusters' section, there is a single entry: 'cluster2881'. This entry includes columns for Name, Region, Zone, Total worker nodes, Scheduled deletion, and Cloud Storage staging bucket. The 'cluster2881' row is highlighted with a green checkmark. To the right of the table, a message says 'No clusters selected'. Below the table, there are tabs for 'PERMISSIONS' and 'LABELS'. A note at the bottom says 'Please select at least one resource.' The browser address bar shows the URL: `console.cloud.google.com/dataproc/clusters?region=us-central1&authuser=1&project=qwiklabs-gcp-03-5ce3a3f8ce28`. The system tray at the bottom indicates the date as 08-11-2020 and the time as 19:27.

Fig2

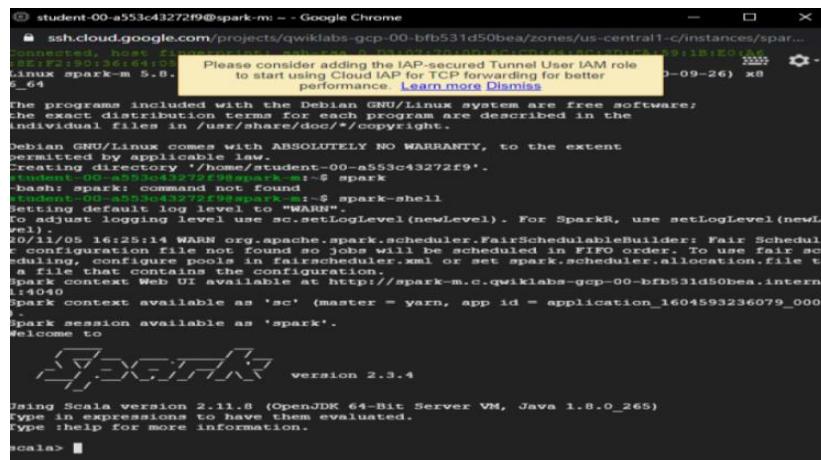
Click on the SSH icon, right to the master node.

The screenshot shows the detailed view of the 'cluster2881' cluster. At the top, it displays basic information: Name (cluster2881), Cluster UUID (cd27792c-79f4-430a-bc42-c9755191131e), Type (DataProc Cluster), and Status (Running). Below this, there are tabs for MONITORING, JOBS, VM INSTANCES, CONFIGURATION, and WEB INTERFACES. The VM INSTANCES tab is selected, showing a table of instances. The first instance listed is 'cluster2881-m', which is identified as the Master node. To the right of its row, there is an 'SSH' icon. The table also lists 'cluster2881-w-0' and 'cluster2881-w-1' as Worker nodes. At the bottom of the page, there is a note about provisioning SSDs for PD-Standard disk types. The browser address bar shows the URL: `console.cloud.google.com/dataproc/clusters/cluster2881/instances?region=us-central1&authuser=1&project=qwiklabs-gcp-03-5ce3a3f8ce28`. The system tray at the bottom indicates the date as 08-11-2020 and the time as 19:27.

Fig3

Step – 2 : Upload the file in SSH (test.txt) and follow the following commands :-

- pwd
- hdfs dfs -mkdir /input_dir
- hdfs dfs -ls
- hdfs dfs -put <pwdPATH>/test.txt /input_dir
- spark-shell



```
student-00-a553c43272f9@spark-m: ~ - Google Chrome
ssh.cloud.google.com/projects/quiklabs-gcp-00-bfb531d50bea/zones/us-central1-c/instances/spark...
Connected host: spark-m.c.quiklabs-gcp-00-bfb531d50bea.internal IP: 10.128.0.105
Please consider adding the IAP-secured Tunnel User IAM role to start using Cloud IAP for TCP forwarding for better performance. Learn more Dismiss
Linux spark-m 5.8.0-64-generic #1-Mon Jul 23 10:59:18 UTC 2018 x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

student-00-a553c43272f9:~$ spark
-bash: spark: command not found
student-00-a553c43272f9:~$ spark-shell
WARN: Default logging level 'WARN'.
Set SPARK_DEFAULT_LOGGING_LEVEL or SPARK_WARN to adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
WARN: 16:25:14 WARN org.apache.spark.scheduler.FairSchedulableBuilder: Fair Scheduler configuration file not found so jobs will be scheduled in FIFO order. To use fair scheduling, configure pools in fairscheduler.xml or set spark.scheduler.allocation.file to a file that contains the configuration.
spark context Web UI available at http://spark-m.c.quiklabs-gcp-00-bfb531d50bea.internal:4040
Spark context available as 'sc' (master = yarn, app id = application_1604593236079_0001).
spark session available as 'spark'.
Welcome to
    \_____/ .-.-\_\_/\_\_/\_\_/\_\_
     \_\_/\_\_/\_\_/\_\_/\_\_/\_\_/\_\_
version 2.3.4

Using Scala version 2.11.8 (OpenJDK 64-Bit Server VM, Java 1.8.0_265)
Type in expressions to have them evaluated.
Type help for more information.
scala>
```

Fig4

- var test = sc.textFile("gs://bucket_name/test.txt"); test.collect();
- var map =
sc.textFile("gs://bucket_name/test.txt").flatMap(line=>line.split(""))).map(word
(word,1));
- map.collect();
- var counts1 = map.reduceByKey(_+_);
- counts1.collect();

```

scala> var test = sc.textFile("gs://word_count_spark/test.txt");
test: org.apache.spark.rdd.RDD[String] = gs://word_count_spark/test.txt MapPartitionsRDD[1] at textFile at <console>:24
scala> test.collect();
res0: Array[String] = Array(I love python I love coding, I love python I love coding, I
love python I love coding, I love python I love coding, I
love python I love coding, I love python I love coding, I
love python I love coding)
scala> var map = sc.textFile("gs://word_count_spark/test.txt").flatMap(line => line.split(" ")).map(word => (word,1));
map: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[5] at map at <console>:24
scala> map.collect();
res1: Array[(String, Int)] = Array((I,1), (love,1), (python,1), (I,1), (love,1), (codin
g,1), (I,1), (love,1), (python,1), (I,1), (love,1), (pytho
n,1), (I,1), (love,1), (coding,1), (I,1), (love,1), (pytho
n,1), (I,1), (love,1), (python,1), (I,1), (love,1), (codin
g,1), (I,1), (love,1), (python,1), (I,1), (love,1), (pytho
n,1), (I,1), (love,1), (coding,1), (I,1), (love,1), (pytho
n,1), (I,1), (love,1), (python,1), (I,1), (love,1), (codin
g,1), (I,1), (love,1), (coding,1), (I,1), (love,1), (pytho
n,1))
scala> var counts1 = map.reduceByKey(_+_);
counts1: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[6] at reduceByKey at <co
nsole>:25
scala> counts1.collect();
res2: Array[(String, Int)] = Array((coding,9), (love,18), (python,9), (I,18))

```

Fig5

To sort in ascending order based on the frequencies of the words appear, sortBy function is used and it shows the final output.

- sort_result = counts1.sortBy(_.value,true);
- Sort_result.collect();

```

scala> val sort_result = counts1.sortBy(_.value,true);
sort_result: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[24] at sortBy at <console>:25
scala> sort_result.collect();
res7: Array[(String, Int)] = Array((coding,9), (python,9), (love,18), (I,18))

```

Fig6

Lab – 15

Student Name and Roll Number: Chayan 18csu054

Semester /Section: 5/A

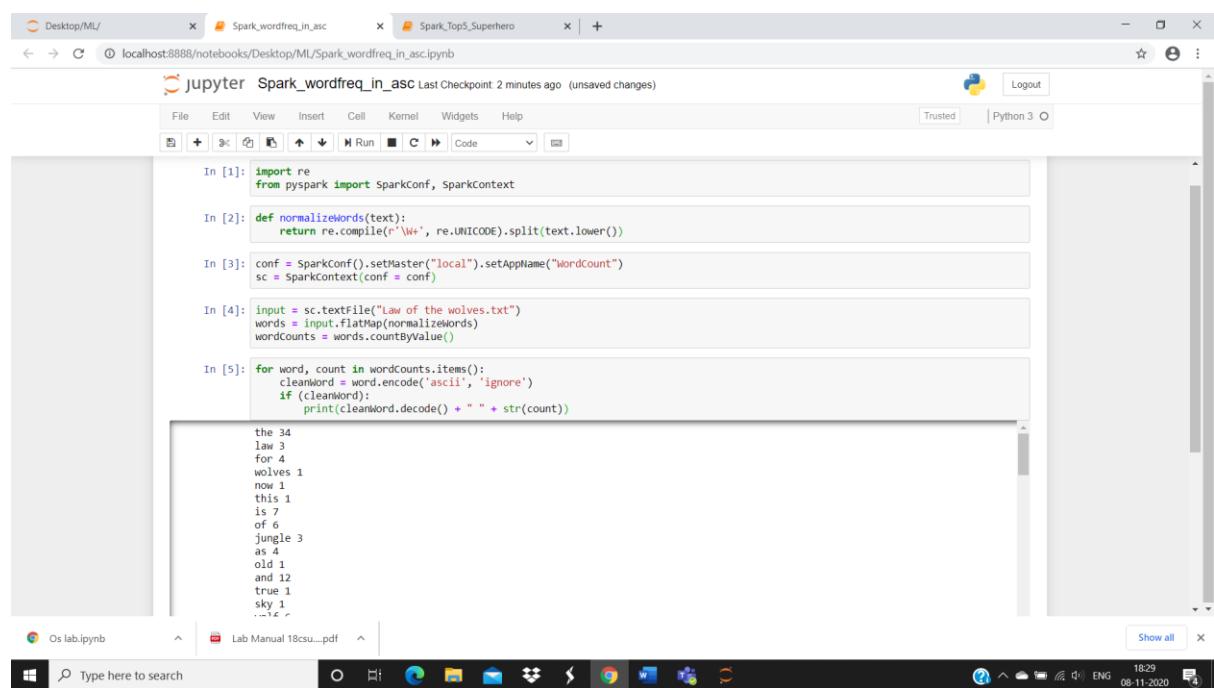
Link to Code: <https://github.com/chayangulati321/Big-Data>

Date: 22/10/20

Faculty Signature:

Remarks:

AIM : To count the words frequency in a file in ascending order.



The screenshot shows a Jupyter Notebook interface with three tabs at the top: "Desktop/ML/" (closed), "Spark_wordfreq_in_asc" (active), and "Spark_Top5_Superhero" (closed). The notebook title is "jupyter Spark_wordfreq_in_asc Last Checkpoint 2 minutes ago (unsaved changes)". The toolbar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, and Python 3. The code cell In [5] contains the following Python code:

```
import re
from pyspark import SparkConf, SparkContext

def normalizeWords(text):
    return re.compile(r'\W+', re.UNICODE).split(text.lower())

conf = SparkConf().setMaster("local").setAppName("WordCount")
sc = SparkContext(conf = conf)

input = sc.textfile("law of the wolves.txt")
words = input.flatMap(normalizeWords)
wordCounts = words.countByValue()

for word, count in wordCounts.items():
    cleanword = word.encode('ascii', 'ignore')
    if (cleanword):
        print(cleanword.decode() + " " + str(count))
```

The output pane shows the results of the word count operation:

```
the 34
law 3
for 4
wolves 1
now 1
this 1
is 7
of 6
jungle 3
as 4
old 1
and 12
true 1
sky 1
... 1
```

At the bottom, the taskbar shows "Os lab.ipynb" and "Lab Manual 18csu...pdf". The system tray indicates the date as 08-11-2020 and the time as 18:29.

Fig1

Lab – 16

Student Name and Roll Number: Chayan 18csu054

Semester /Section: 5/A

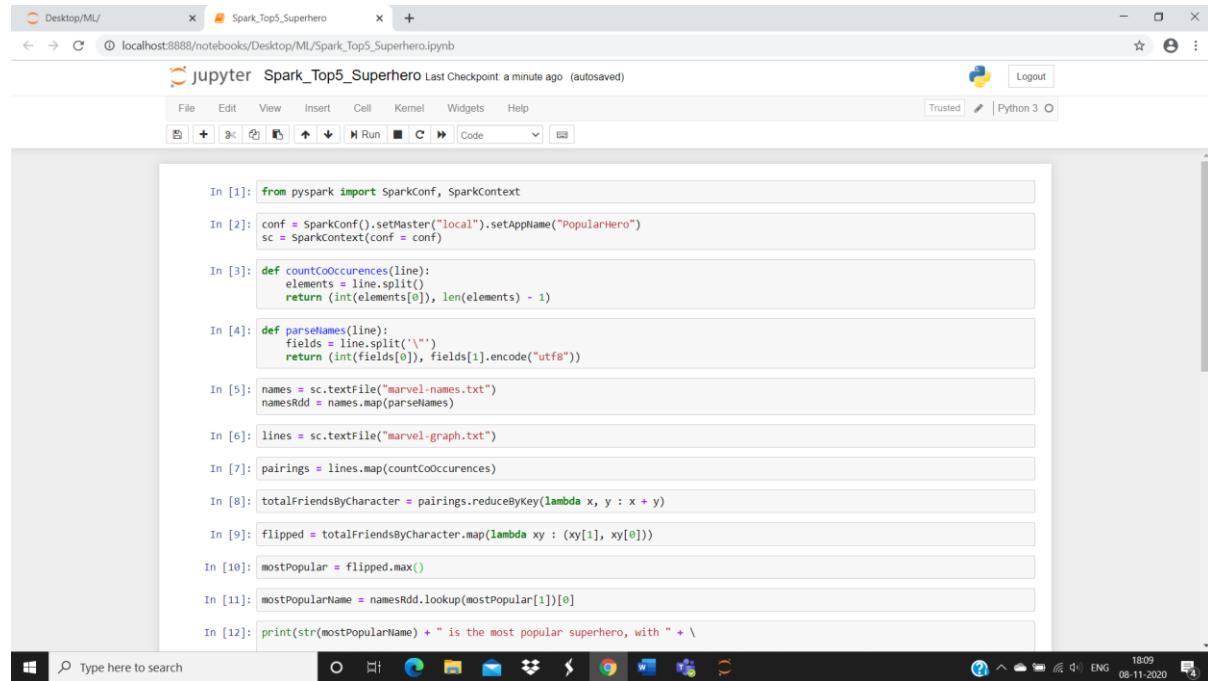
Link to Code: <https://github.com/chayangulati321/Big-Data>

Date: 27/10/20

Faculty Signature:

Remarks:

AIM : To find top 5 popular superhero name and their ID's



The screenshot shows a Jupyter Notebook interface with the title "Spark_Top5_Superhero". The notebook contains the following Python code:

```
In [1]: from pyspark import SparkConf, SparkContext
In [2]: conf = SparkConf().setMaster("local").setAppName("PopularHero")
sc = sparkContext(conf = conf)
In [3]: def countCoOccurrences(line):
    elements = line.split()
    return (int(elements[0]), len(elements) - 1)
In [4]: def parseNames(line):
    fields = line.split("\t")
    return (int(fields[0]), fields[1].encode("utf8"))
In [5]: names = sc.textFile("marvel-names.txt")
namesRdd = names.map(parseNames)
In [6]: lines = sc.textFile("marvel-graph.txt")
In [7]: pairings = lines.map(countCoOccurrences)
In [8]: totalFriendsByCharacter = pairings.reduceByKey(lambda x, y : x + y)
In [9]: flipped = totalFriendsByCharacter.map(lambda xy : (xy[1], xy[0]))
In [10]: mostPopular = flipped.max()
In [11]: mostPopularName = namesRdd.lookup(mostPopular[1])[0]
In [12]: print(str(mostPopularName) + " is the most popular superhero, with " + \
```

Fig1

The screenshot shows a Jupyter Notebook interface running on a Windows desktop. The notebook has one open cell titled "Spark_Top5_Superhero.ipynb". The code in the cell is as follows:

```
In [12]: print(str(mostPopularName) + " is the most popular superhero, with " + \ str(mostPopular[0]) + " co-appearances.")  
Out[12]: b'CAPTAIN AMERICA' is the most popular superhero, with 1933 co-appearances.  
  
In [13]: mostPopularChar = flipped.top(5)  
In [14]: mostPopularChar  
Out[14]: [(1933, 859), (1741, 5306), (1528, 2664), (1426, 5716), (1394, 6306)]  
• Top 5 superheroes with their frequencies  
  
In [15]: for i in range(len(mostPopularChar)):  
    x = namesRdd.lookup(mostPopularChar[i][1])  
    y = mostPopularChar[i][0]  
    print(x, '\t', y)  
  
Out[15]: [b'CAPTAIN AMERICA'] 1933  
[b'SPIDER-MAN/PETER PARKER'] 1741  
[b'IRON MAN/TONY STARK'] 1528  
[b'THING/BENJAMIN JONES'] 1426  
[b'WOLVERINE/LOGAN'] 1394  
• Top 5 superheroes with their ID's  
  
In [16]: for i in range(len(mostPopularChar)):  
    x = namesRdd.lookup(mostPopularChar[i][1])  
    y = mostPopularChar[i][1]  
    print(x, '\t', y)  
  
Out[16]: [b'CAPTAIN AMERICA'] 859  
[b'SPIDER-MAN/PETER PARKER'] 5306  
[b'IRON MAN/TONY STARK'] 2664  
[b'THING/BENJAMIN JONES'] 5716  
[b'WOLVERINE/LOGAN'] 6306
```

The notebook is running in Python 3. The status bar at the bottom right shows the date as 08-11-2020 and the time as 18:09.

Fig2

Lab – 17

Student Name and Roll Number: Chayan 18csu054

Semester /Section: 5/A

Link to Code: <https://github.com/chayangulati321/Big-Data>

Date: 09/11/20

Faculty Signature:

Remarks:

AIM : To find the lowest rated movie of all time. Consider only those movies who have more than 10 user ratings.

```
In [1]: 1 from pyspark import SparkConf, SparkContext
2 def loadMovieNames():
3     movieNames = {}
4     with open("u.item") as f:
5         for line in f:
6             fields = line.split('|')
7             movieNames[int(fields[0])] = fields[1]
8     return movieNames
9 def parseInput(line):
10    fields = line.split()
11    return (int(fields[1]), (float(fields[2]), 1.0))
12
13 if __name__ == "__main__":
14     conf = SparkConf().setAppName("WorstMovies")
15     sc = SparkContext(conf = conf)
16     movieNames = loadMovieNames()
17     lines = sc.textFile("u.data")
18     movieRatings = lines.map(parseInput)
19     ratingTotalsAndCount = movieRatings.reduceByKey(lambda movie1, movie2: ( movie1[0] + movie2[0], movie1[1] + movie2[1] ))
20     popularTotalsAndCount = ratingTotalsAndCount.filter(lambda key_val: key_val[1][1] > 10)
21     averageRatings = popularTotalsAndCount.mapValues(lambda totalAndCount : totalAndCount[0] / totalAndCount[1])
22     sortedMovies = averageRatings.sortBy(lambda x: x[1])
23     results = sortedMovies.take(1)
24     for result in results:
25         print(movieNames[result[0]], result[1])
```

Children of the Corn: The Gathering (1996) 1.3157894736842106

Fig1

Lab – 18

Student Name and Roll Number: Chayan 18csu054

Semester /Section: 5/A

Link to Code: <https://github.com/chayangulati321/Big-Data>

Date: 27/11/20

Faculty Signature:

Remarks:

ss

Spark Machine Learning implementation of Algorithms

AIM : To implement machine learning on a dataset using pyspark.

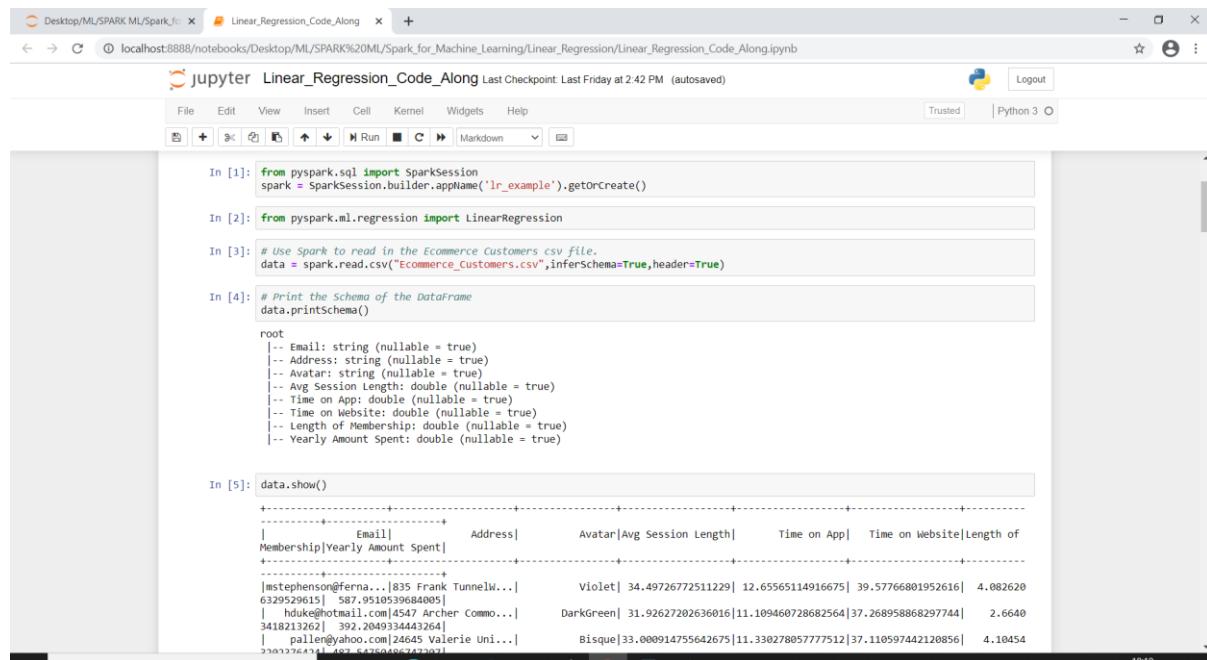
Step – 1 : First start a SparkSession with following commands:-

- \$ from pyspark.sql import SparkSession
- \$ SparkSession.builder.appName('any_name').getOrCreate()

After creating SparkSession import LinearRegression library

- \$ from pyspark.ml.regression import LinearRegression

Read the csv file and print the schema as shown in Fig1.



The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** Desktop/ML/SPARK ML/Spark_f... x Linear_Regression_Code_Along x +
- Header:** jupyter Linear_Regression_Code_Along Last Checkpoint: Last Friday at 2:42 PM (autosaved)
- Toolbar:** File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 O
- In [1]:**

```
from pyspark.sql import SparkSession  
spark = SparkSession.builder.appName('lr_example').getOrCreate()
```
- In [2]:**

```
from pyspark.ml.regression import LinearRegression
```
- In [3]:**

```
# Use Spark to read in the Ecommerce Customers csv file.  
data = spark.read.csv("Ecommerce_Customers.csv", inferSchema=True, header=True)
```
- In [4]:**

```
# Print the Schema of the DataFrame  
data.printSchema()
```

Output:

```
root  
 |-- Email: string (nullable = true)  
 |-- Address: string (nullable = true)  
 |-- Avatar: string (nullable = true)  
 |-- Avg Session Length: double (nullable = true)  
 |-- Time on App: double (nullable = true)  
 |-- Time on Website: double (nullable = true)  
 |-- Length of Membership: double (nullable = true)  
 |-- Yearly Amount Spent: double (nullable = true)
```
- In [5]:**

```
data.show()
```

Output:

+	Email	Address	Avatar	Avg Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent	+
	msstephenson@ferna...	1835 Frank TunnelW...	Violet	34.49726772511229	12.65565114916675	39.57766801952616	4.082620		+
	6329529615	587.9510539684005							+
	hduke@hotmail.com	4547 Archer Commo...	DarkGreen	31.92627202636016	11.109460728682564	37.26895868297744	2.6640		+
	3418213262	392.2049334443264							+
	pallen@yahoo.com	24645 Valerie Uni...	Bisque	33.000914755642675	11.330278057777512	37.110597442120856	4.10454		+
	3301376471	1497.647804862473071							+

Fig1

Step – 2 : Check the head of dataset. Import few more libraries :-

- `$ from pyspark.ml.linalg import Vectors`
- `$ from pyspark.ml.feature import VectorAssembler`

VA is a transformer that combines a given list of columns into a single vector column.

The screenshot shows a Jupyter Notebook interface with the title "Linear_Regression_Code_Along". The notebook has several cells:

- In [6]:** `data.head()`
Out[6]:
Row(Email='mstephenson@fernandez.com', Address='835 Frank TunnelWrightmouth, MI 82180-9605', Avatar='Violet', Avg Session Length=34.49726772511229, Time on App=12.65565114916675, Time on Website=39.57766801952616, Length of Membership=4.0826206329529615, Yearly Amount Spent=587.9510539684005)
- In [7]:** `for item in data.head():
 print(item)`
mstephenson@fernandez.com
835 Frank TunnelWrightmouth, MI 82180-9605
Violet
34.49726772511229
12.65565114916675
39.57766801952616
4.0826206329529615
587.9510539684005
- Setting Up DataFrame for Machine Learning**
- In [8]:** `# A few things we need to do before Spark can accept the data
It needs to be in the form of two columns
("label", "features")

Import VectorAssembler and Vectors
from pyspark.ml.linalg import Vectors
from pyspark.ml.feature import VectorAssembler`
- In [9]:** `data.columns`
Out[9]: ['Email',
 'Address',
 'Avatar',
 'Avg Session Length',
 'Time on App',
 'Length of Membership',
 'Yearly Amount Spent']

Fig 2

Step – 3 : Take Input columns as all numerical except ‘yearly amount spent’ and transform them to new column named ‘features’.

The screenshot shows a Jupyter Notebook interface with the title "Linear_Regression_Code_Along". The notebook contains the following code:

```
In [9]: data.columns
Out[9]: ['Email',
 'Address',
 'Avatar',
 'Avg Session Length',
 'Time on App',
 'Time on Website',
 'Length of Membership',
 'Yearly Amount Spent']

In [10]: assembler = VectorAssembler(
    inputCols=["Avg Session Length", "Time on App",
               "Time on Website", "Length of Membership"],
    outputCol="features")

In [11]: output = assembler.transform(data)

In [12]: output.select("features").show()
+-----+
| features|
+-----+
|[34.4972677251122...|
|[31.9262720263601...|
|[33.0009147556426...|
|[34.3055566297555...|
|[33.3306725236463...|
|[33.8710378793419...|
|[32.0215955013870...|
|[32.7391429383883...|
|[33.9877728956656...|
|[31.9365486184489...|
|[33.9925727749537...|
|[33.8793608248049...|
|[29.5324289670579...|
+-----+
```

Fig 3

Step – 4 : All features as ‘featuers’ and target feature as ‘yearly amount spent’. Divide the data randomly into training (70%) and testing (30%).

The screenshot shows a Jupyter Notebook interface with the title "Linear_Regression_Code_Along". The notebook contains the following code:

```
In [14]: final_data = output.select("features", "Yearly Amount Spent")
In [17]: final_data.head()
Out[17]: Row(features=DenseVector([34.4973, 12.6557, 39.5777, 4.0826]), Yearly Amount Spent=587.9510539684005)
In [18]: train_data,test_data = final_data.randomSplit([0.7,0.3])
In [19]: train_data.describe().show()
+-----+
|summary|Yearly Amount Spent|
+-----+
| count | 352 |
| mean | 501.49902646844134 |
| stdDev | 78.37085357152726 |
| min | 256.67898229905585 |
| max | 765.5184619388373 |
+-----+

In [20]: test_data.describe().show()
+-----+
|summary|Yearly Amount Spent|
+-----+
| count | 148 |
| mean | 494.1173095432702 |
| stdDev | 81.5499564517557 |
| min | 275.9184206503857 |
| max | 708.9351848669818 |
+-----+
```

Fig 4

Step – 5 : Create a object and fit the model to it. Check the test data residual means how much test data is different from train data.

```

In [21]: # Create a Linear Regression Model object
lr = LinearRegression(labelCol='Yearly_Amount_Spent')

In [22]: # Fit the model to the data and call this model lrModel
lrModel = lr.fit(train_data)

In [23]: # print the coefficients and intercept for linear regression
print("Coefficients: {} Intercept: {}".format(lrModel.coefficients,lrModel.intercept))

Coefficients: [25.50410368282677, 38.79227768099469, 0.8531679940527187, 61.29518404951587] Intercept: -1059.2994203885783

In [24]: test_results = lrModel.evaluate(test_data)

In [25]: # Interesting results...
test_results.residuals.show()

+-----+
| residuals|
+-----+
| 9.5759269394905|
| -13.200826257672304|
| -21.307536353678074|
| -1.8107935602769203|
| -6.103901714125243|
| -10.32217239377809|
| -2.23426849597282|
| -5.747763849821524|
| -26.8024591675488|
| -7.68834716735239|
| 0.402163519602096|
| -2.30315973593784|
| -11.18028754307909|
| 14.310843622095206|
| 7.831809617624026|
| -5.650948163218503|
| -8.669279406550812|
+-----+

```

Fig 5

Step – 6 : After training the model on train data now test it at test data and get the predictions. Root mean square error (rmse) is 10.09 which means how much predicted values different from actual values.

```

In [26]: unlabeled_data = test_data.select('features')

In [27]: predictions = lrModel.transform(unlabeled_data)

In [28]: predictions.show()

+-----+-----+
| features| prediction|
+-----+-----+
| [29.5324289670579,...] | 399.064424133137|
| [31.06613181616375,...] | 1462.13411946534666|
| [31.129743499119,...] | 508.25459019344385|
| [31.2606468698795,...] | 423.1374248172283|
| [31.5171218025062,...] | 282.02232236451096|
| [31.5261978982398,...] | 419.4166985861159|
| [31.576131971322,...] | 543.46001083888356|
| [31.6253601348308,...] | 382.0846646067457|
| [31.6739155932749,...] | 502.52752707743|
| [31.7287699002873,...] | 546.4632806453753|
| [31.8293464559211,...] | 384.723401611642|
| [31.8627411090001,...] | 558.6910769099645|
| [31.8648313011111,...] | 451.1156809089277|
| [31.880801844692,...] | 534.531051515134|
| [31.9548038565348,...] | 432.16607932230205|
| [31.9673209479824,...] | 451.40078940207076|
| [32.0005045178551,...] | 452.1665004352635|
| [32.0305497162129,...] | 589.3421092485082|
| [32.0478146331398,...] | 486.63065955869547|
| [32.0637746203136,...] | 390.62999734038704|
+-----+
only showing top 20 rows

In [29]: print("RMSE: {}".format(test_results.rootMeanSquareError))
print("MSE: {}".format(test_results.meanSquareError))

RMSE: 10.091469776113366
MSE: 101.83776224220955

```

Fig 6