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

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

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 ☐	Programme Elective ✓	Open Elective ☐
7. Pre-requisite(s), if any: Data Engineering			
8. Frequency of offering (check one): Odd ☐	Even ✓	Either semester ☐	Every semester ☐

9. Brief Syllabus:

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.

Total lecture, Tutorial and Practical Hours for this course (Take 15 teaching weeks per semester): 90 hours

The class size is maximum 30 learners.

	Practice	
Lectures: 45 hours	**Tutorials :** 0 hours	**Lab Work:** 75 hours

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 **No. of 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

- https://www.qwiklabs.com/quests/50

Supplementary MOOC Courses

- https://www.coursera.org/specializations/big-data
- https://www.coursera.org/learn/gcp-fundamentals

Certification courses/programs for Skill Development

- 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*, Wilely 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 moview 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

<table>
<tr><th colspan="2">Marks Distribution</th></tr>
<tr><th>Continuous Evaluation(50 Marks)</th><th>End Semester Exam (20 Marks)</th></tr>
<tr><td>Each experiment shall be evaluated for 10 marks and at the end of the semester proportional marks shall be awarded out of 50.</td><td rowspan="2">End semester practical evaluation including Mini project (if any) carries 20 marks.</td></tr>
<tr><td>Following is the breakup of 10 marks for each
4 Marks: Observation & conduct of experiment. Teacher may ask questions about experiment.
3 Marks: For report writing
3 Marks: For the 15 minutes quiz to be conducted in every lab.</td></tr>
</table>

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 – 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

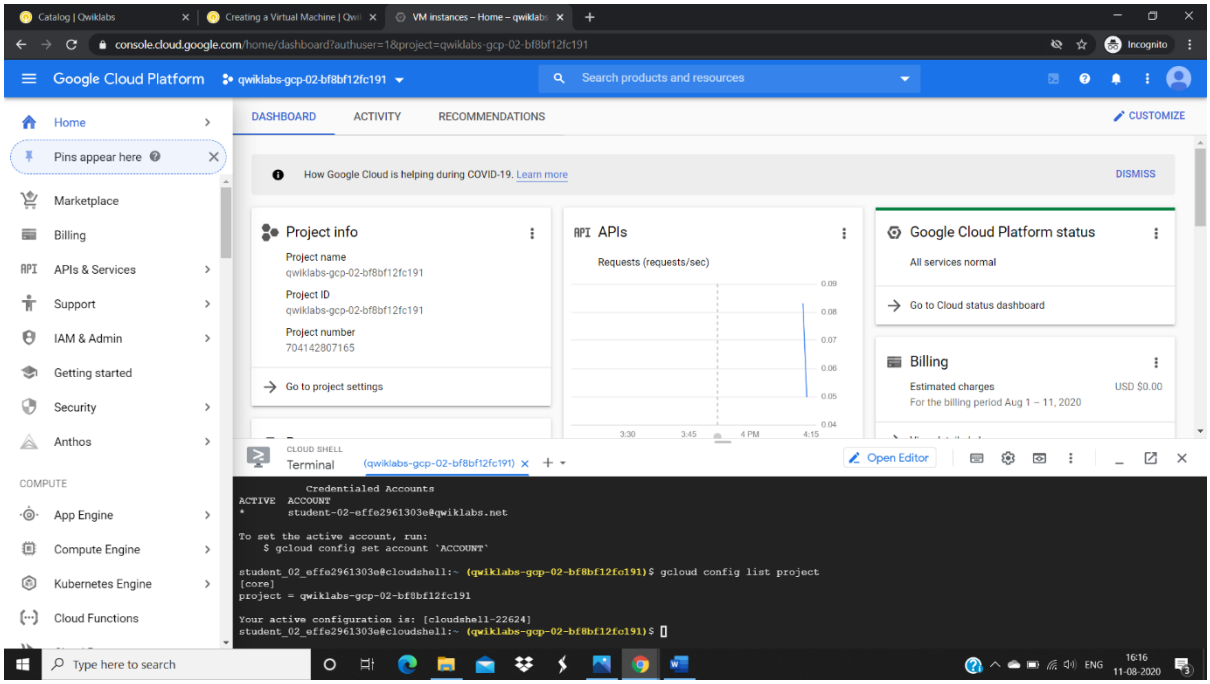


Fig1

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**.

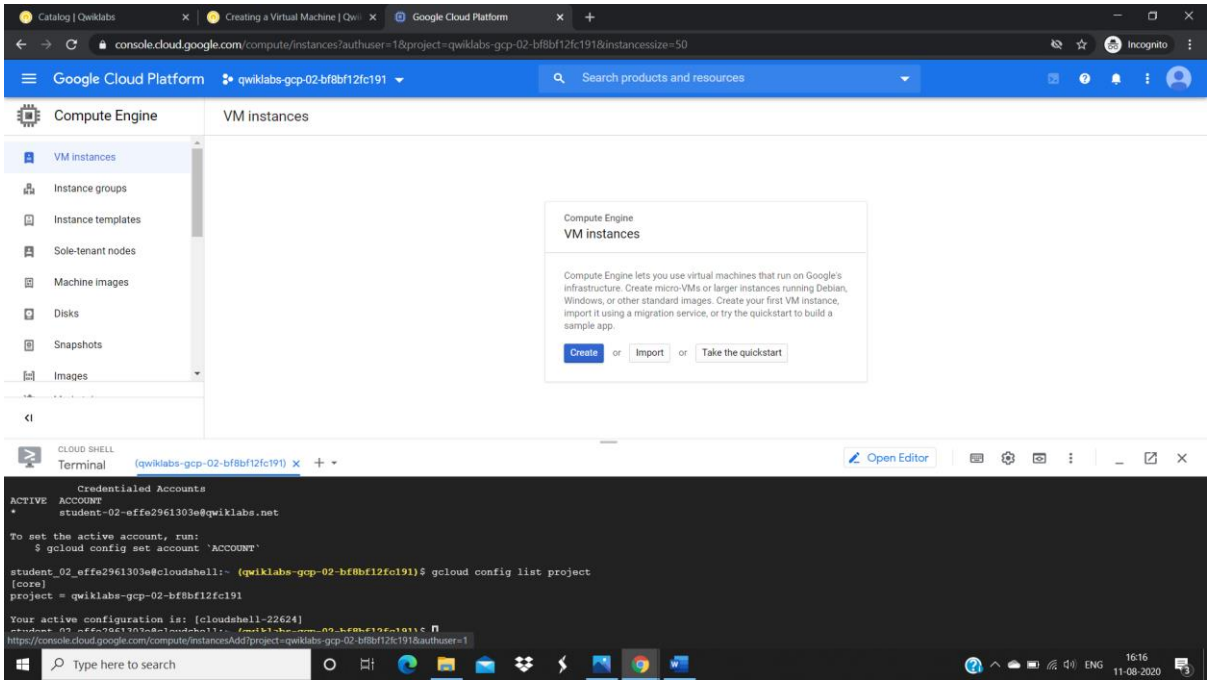


Fig2

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

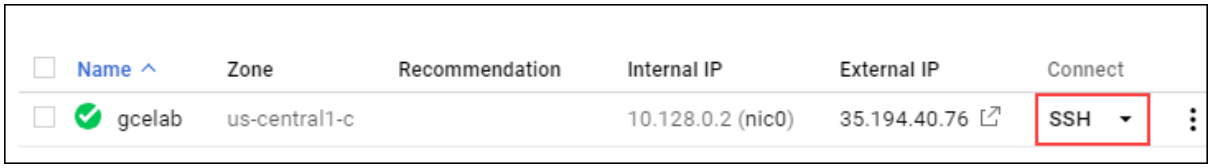


Fig4

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

- $sudo su –
- $apt-get update

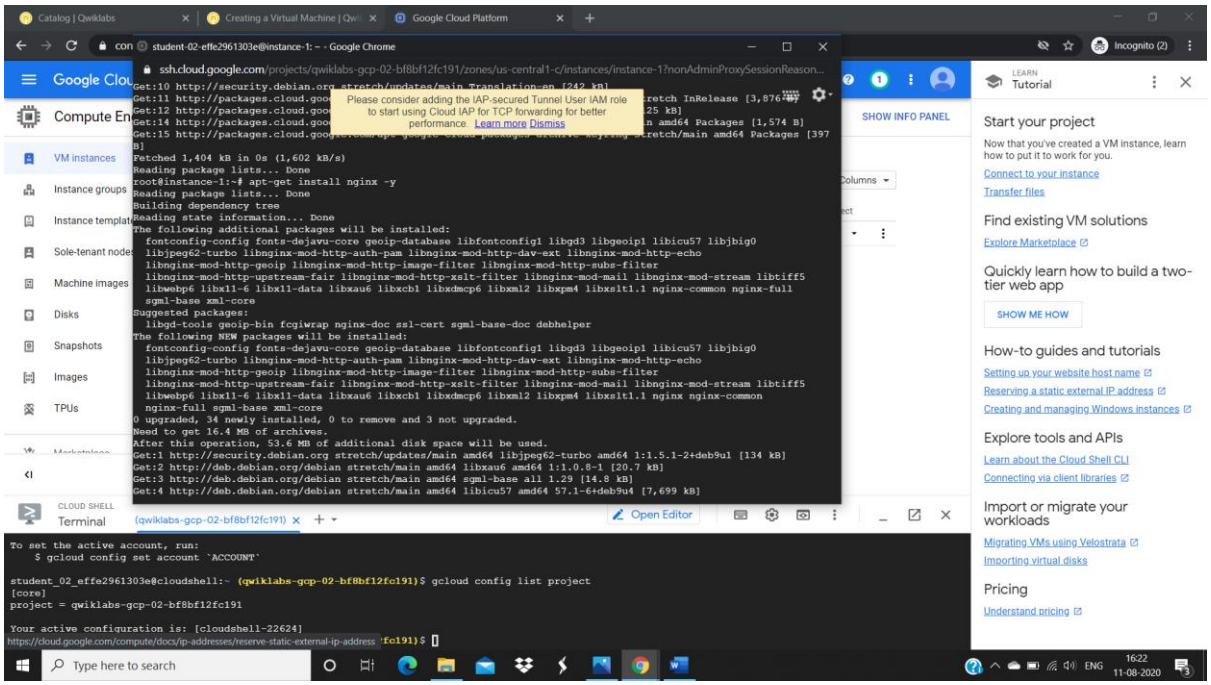


Fig5

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

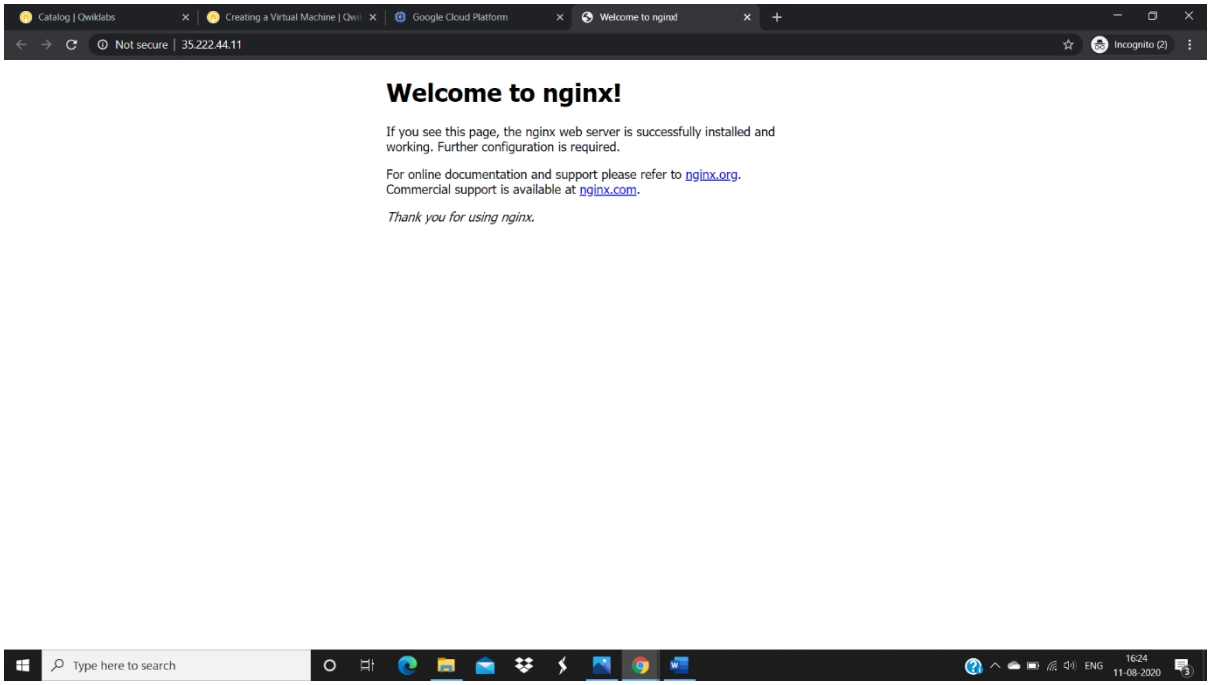


Fig6

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-centrall-c

Select **Navigation menu** > **Compute Engine** > **VM instances**

Create a new instance with gcloud

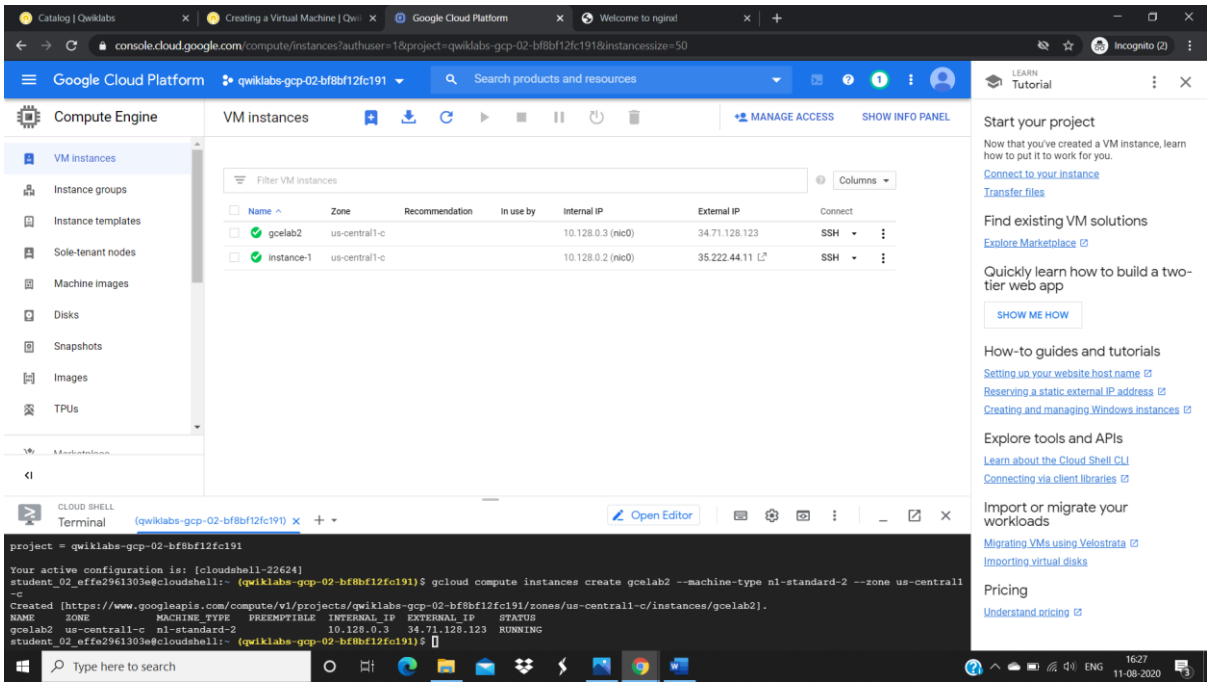


Fig7

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-centrall-c
- Now you'll type **Y** to continue
- exit

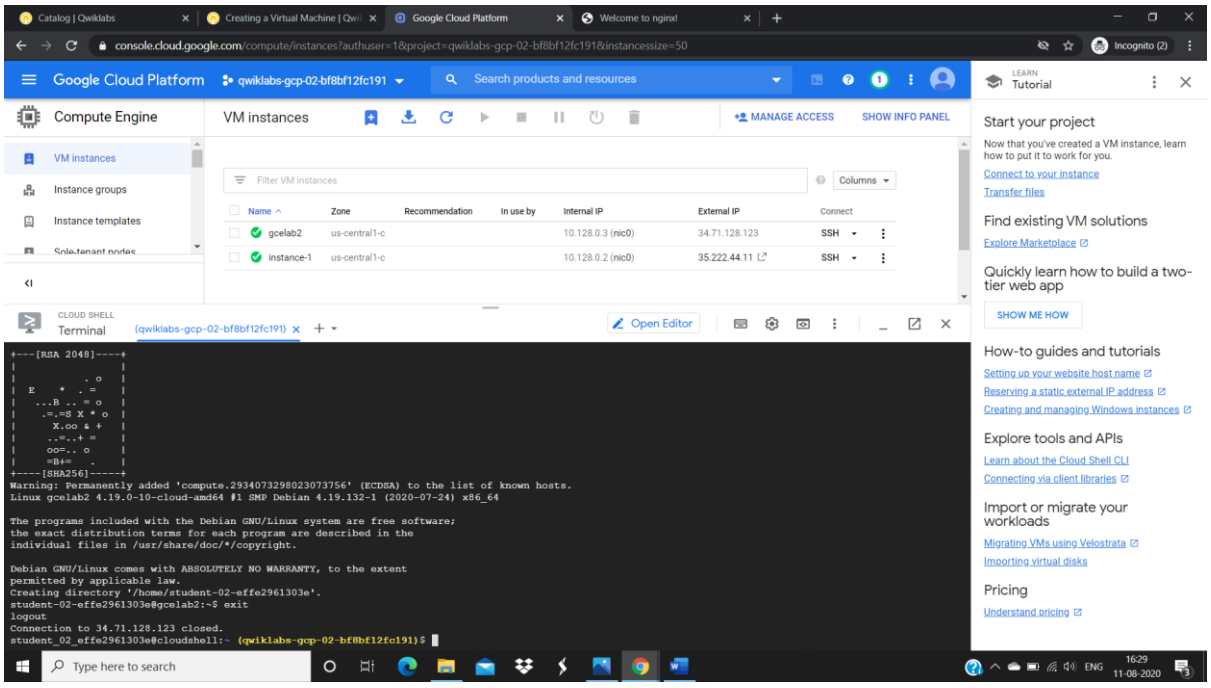


Fig8

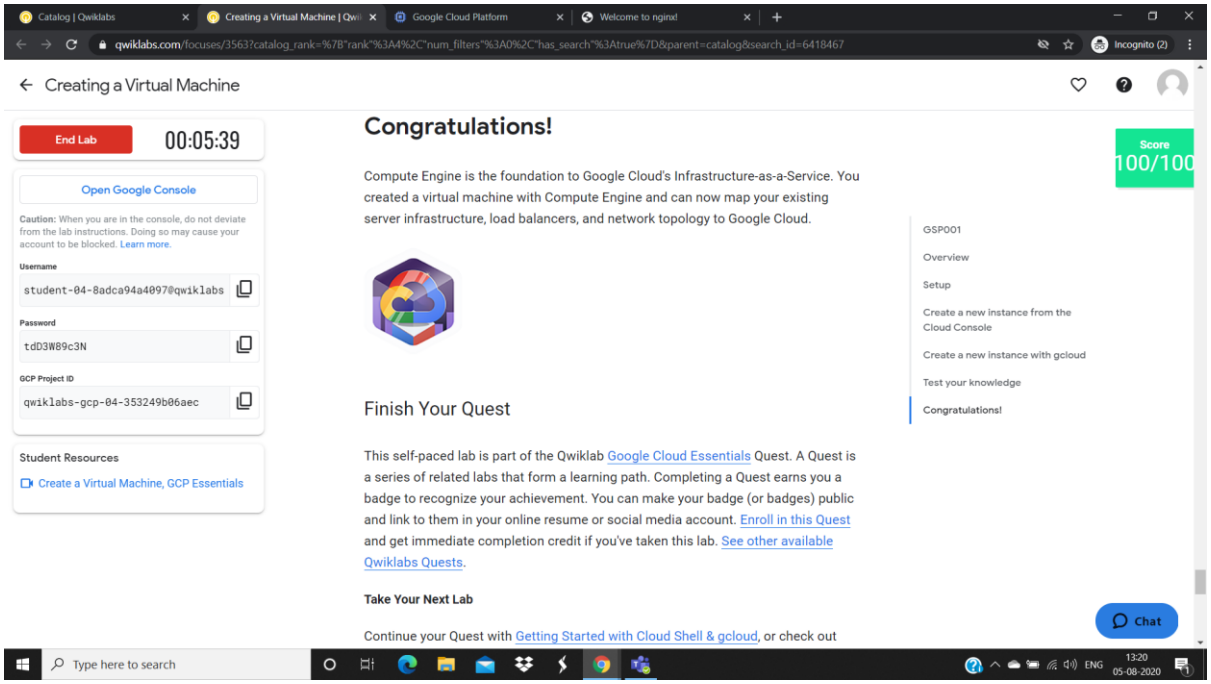

Creating a Virtual Machine
End Lab
00:05:39
Open Google Console
Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.
Username
student-04-8adca94a4097@qwiklabs
Password
tdD3W89c3N
GCP Project ID
qwiklabs-gcp-04-353249b06aec
Student Resources
Create a Virtual Machine, GCP Essentials
Congratulations!
Compute Engine is the foundation to Google Cloud's Infrastructure-as-a-Service. You created a virtual machine with Compute Engine and can now map your existing server infrastructure, load balancers, and network topology to Google Cloud.
Score
100/100
GSP001
Overview
Setup
Create a new instance from the Cloud Console
Create a new instance with gcloud
Test your knowledge
Congratulations!
Finish Your Quest
This self-paced lab is part of the Qwiklab Google Cloud Essentials Quest. A Quest is a series of related labs that form a learning path. Completing a Quest earns you a badge to recognize your achievement. You can make your badge (or badges) public and link to them in your online resume or social media account. Enroll in this Quest and get immediate completion credit if you've taken this lab. See other available Qwiklabs Quests.
Take Your Next Lab
Continue your Quest with Getting Started with Cloud Shell & gcloud, or check out
Chat
Type here to search
ENG
13:20
05-08-2020

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:

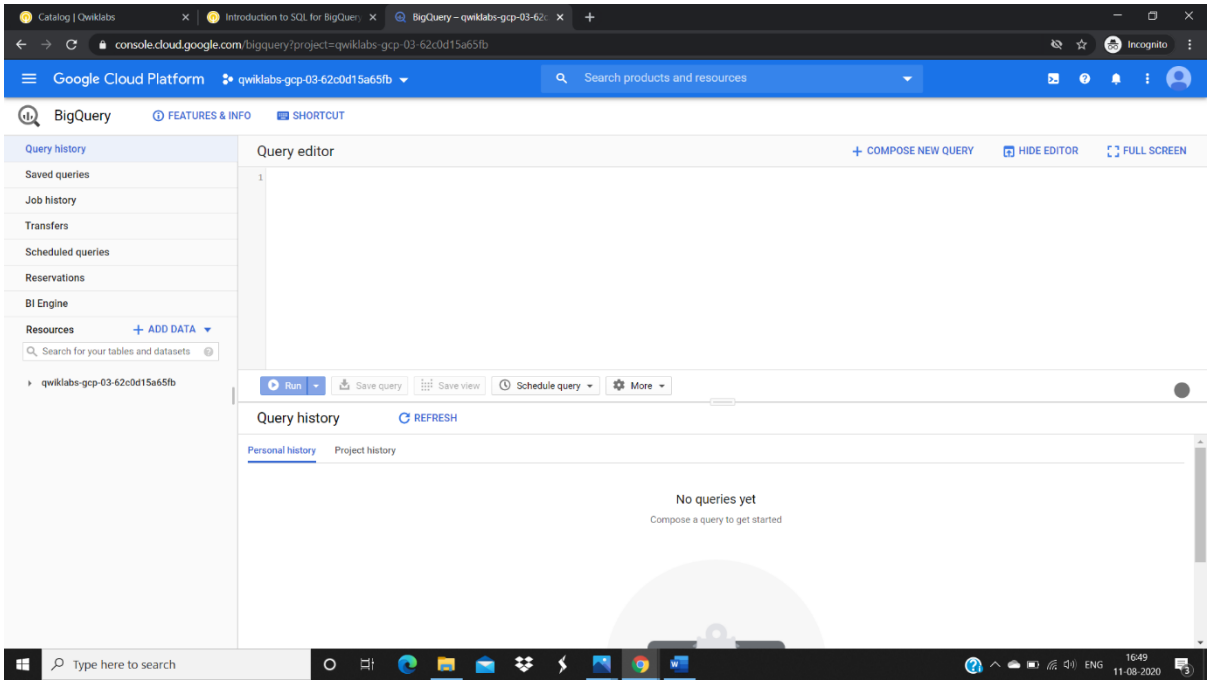


Fig1

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 tab will open, and you will now have a new project called bigquery-public-data added to the Resources panel
- Click on **bigquery-public-data** > **london_bicycles** > **cycle_hire**

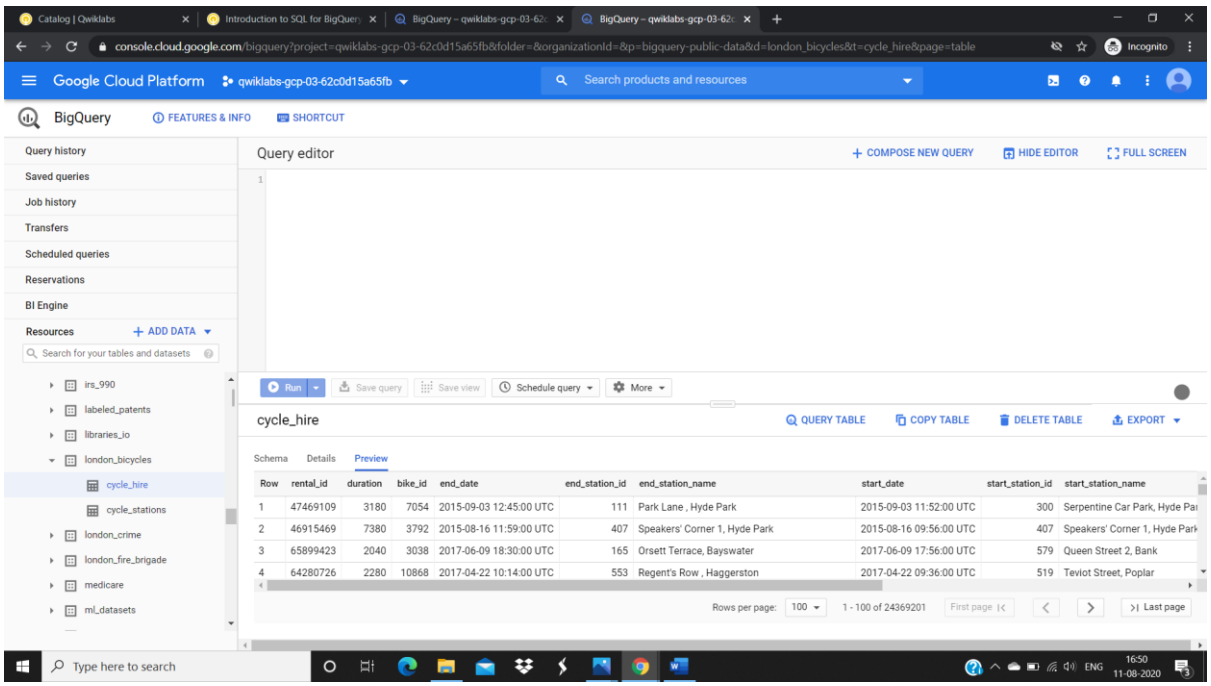


Fig2

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;

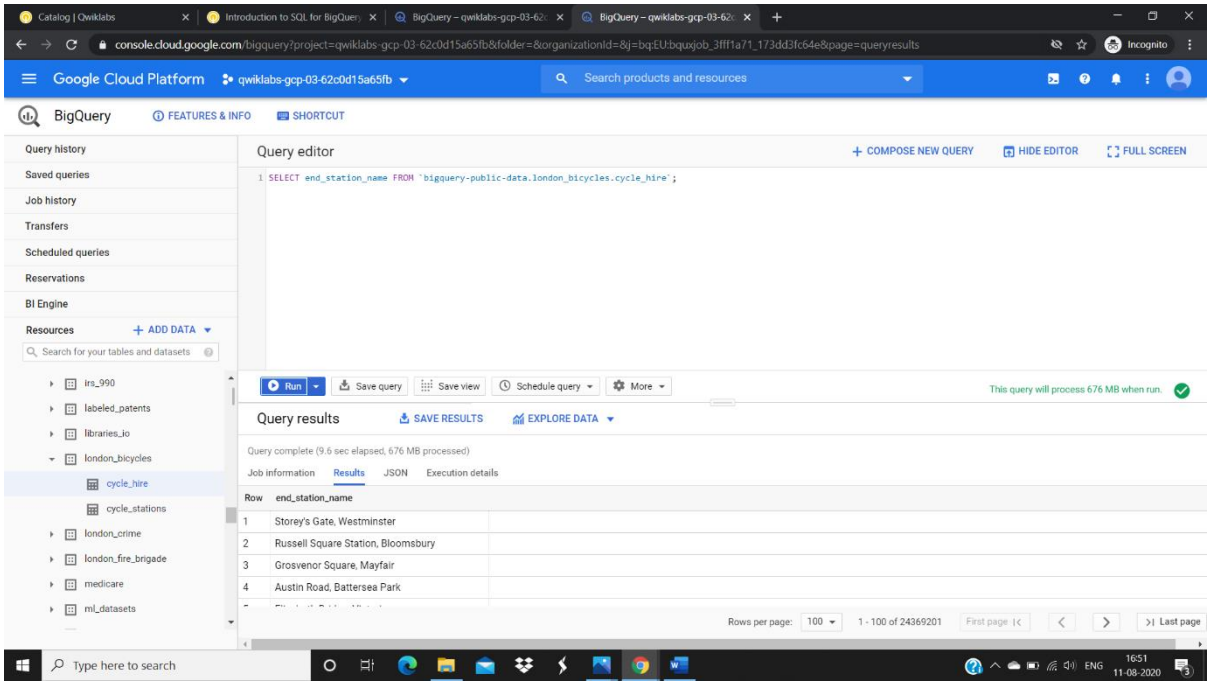


Fig3

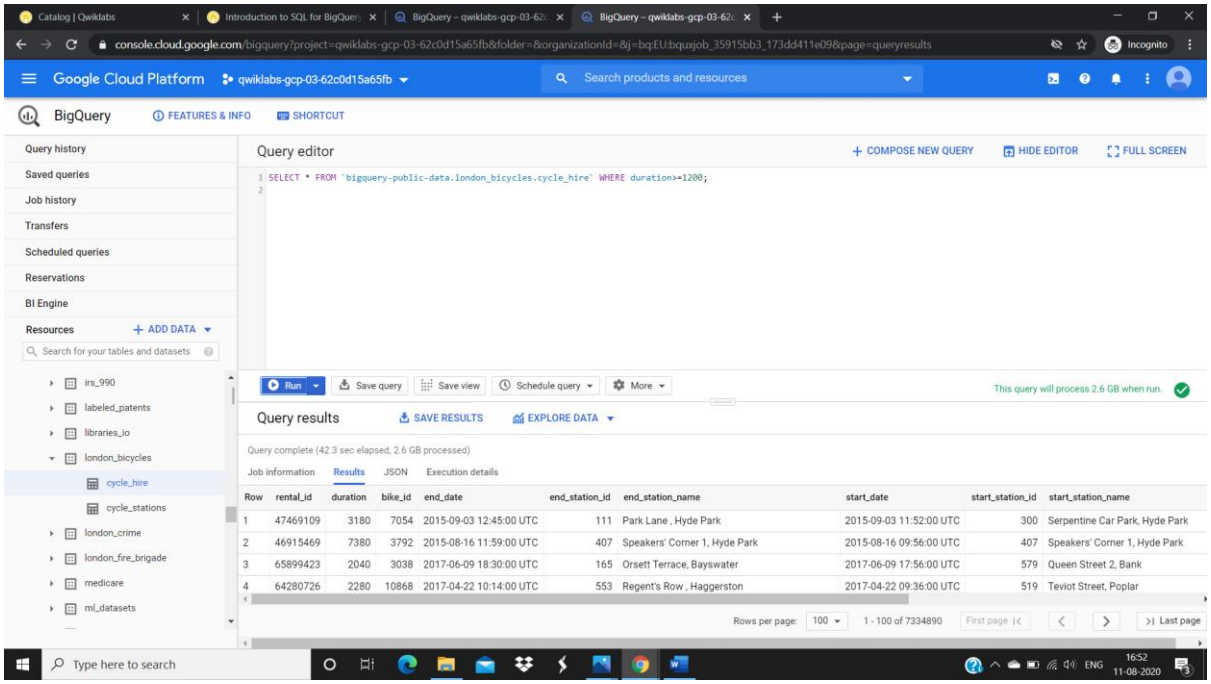


Fig4

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;

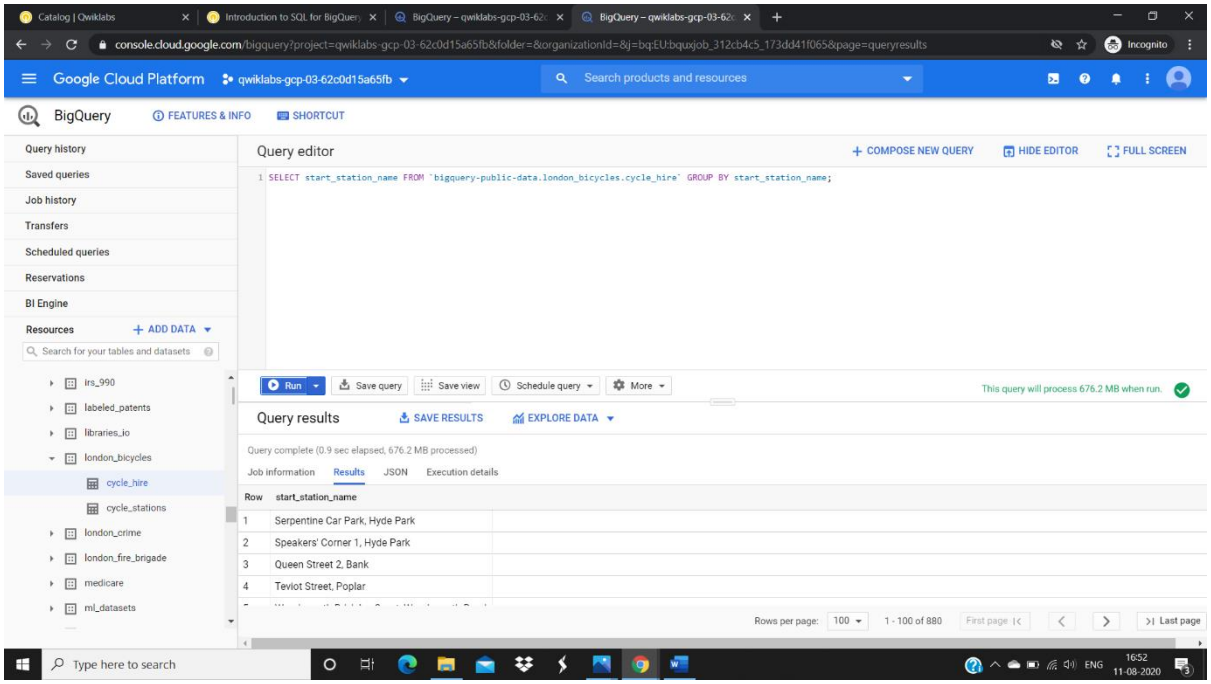

Catalog | Qwiklabs
Introduction to SQL for BigQuery
BigQuery – qwiklabs-gcp-03-62c
BigQuery – qwiklabs-gcp-03-62c
console.cloud.google.com/bigquery?project=qwiklabs-gcp-03-62c0d15a65fb&folder=&organizationId=&j=bq:EU:bquxjob_312cb4c5_173dd41f065&page=queryresults
Incognito
Google Cloud Platform
qwiklabs-gcp-03-62c0d15a65fb
Search products and resources
BigQuery
FEATURES & INFO
SHORTCUT
Query history
Saved queries
Job history
Transfers
Scheduled queries
Reservations
BI Engine
Resources
ADD DATA
Search for your tables and datasets
irs_990
labeled_patents
libraries_io
london_bicycles
cycle_hire
cycle_stations
london_crime
london_fire_brigade
medicare
ml_datasets
Query editor
COMPOSE NEW QUERY
HIDE EDITOR
FULL SCREEN
1 SELECT start_station_name FROM `bigquery-public-data.london_bicycles.cycle_hire` GROUP BY start_station_name;
Run
Save query
Save view
Schedule query
More
This query will process 676.2 MB when run.
Query results
SAVE RESULTS
EXPLORE DATA
Query complete (0.9 sec elapsed, 676.2 MB processed)
Job information
Results
JSON
Execution details
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
Rows per page: 100
1 - 100 of 880
First page
Last page
Type here to search
ENG
16:52
11-08-2020

Fig5

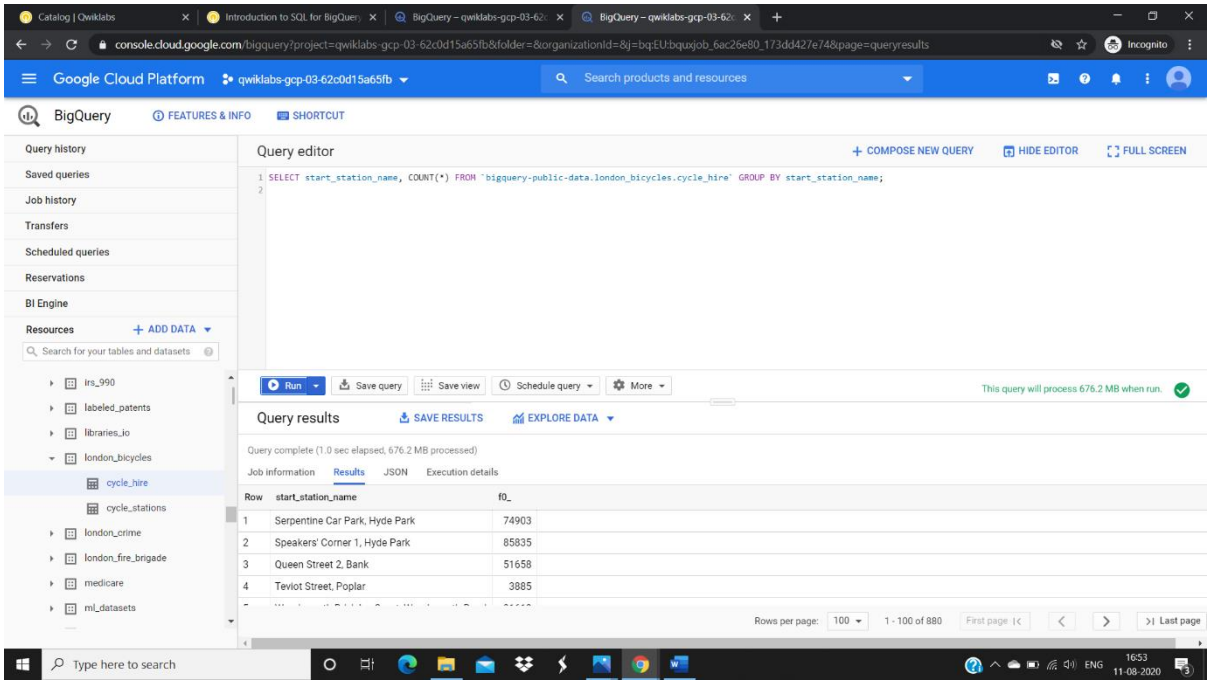

Catalog | Qwiklabs
Introduction to SQL for BigQuery
BigQuery – qwiklabs-gcp-03-62c
BigQuery – qwiklabs-gcp-03-62c
console.cloud.google.com/bigquery?project=qwiklabs-gcp-03-62c0d15a65fb&folder=&organizationId=&j=bq:EU:bquxjob_6ac26e80_173dd427e74&page=queryresults
Incognito
Google Cloud Platform
qwiklabs-gcp-03-62c0d15a65fb
Search products and resources
BigQuery
FEATURES & INFO
SHORTCUT
Query history
Saved queries
Job history
Transfers
Scheduled queries
Reservations
BI Engine
Resources
ADD DATA
Search for your tables and datasets
irs_990
labeled_patents
libraries_io
london_bicycles
cycle_hire
cycle_stations
london_crime
london_fire_brigade
medicare
ml_datasets
Query editor
COMPOSE NEW QUERY
HIDE EDITOR
FULL SCREEN
1 SELECT start_station_name, COUNT(*) FROM `bigquery-public-data.london_bicycles.cycle_hire` GROUP BY start_station_name;
2
Run
Save query
Save view
Schedule query
More
This query will process 676.2 MB when run.
Query results
SAVE RESULTS
EXPLORE DATA
Query complete (1.0 sec elapsed, 676.2 MB processed)
Job information
Results
JSON
Execution details
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
Rows per page: 100
1 - 100 of 880
First page
Last page
Type here to search
ENG
16:53
11-08-2020

Fig6

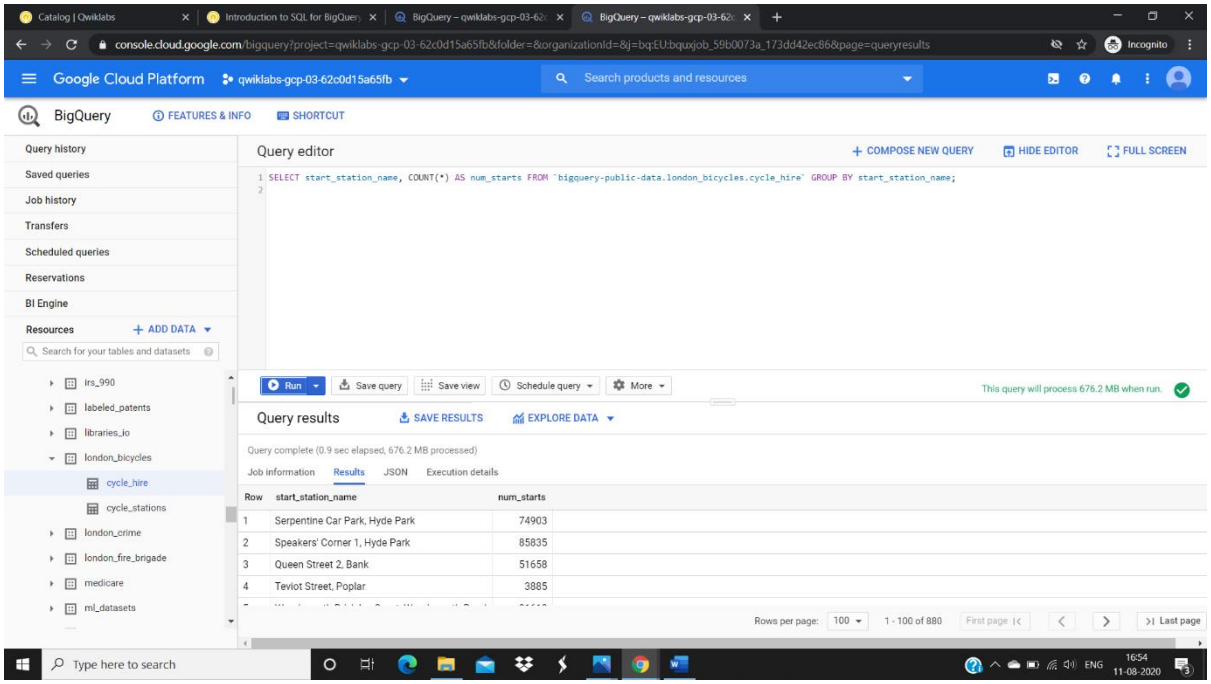

Google Cloud Platform
qwiklabs-gcp-03-62c0d15a65fb
Search products and resources
BigQuery
FEATURES & INFO
SHORTCUT
Query history
Saved queries
Job history
Transfers
Scheduled queries
Reservations
BI Engine
Resources
ADD DATA
Search for your tables and datasets
irs_990
labeled_patents
libraries_io
london_bicycles
cycle_hire
cycle_stations
london_crime
london_fire_brigade
medicare
ml_datasets
Query editor
COMPOSE NEW QUERY
HIDE EDITOR
FULL SCREEN
SELECT start_station_name, COUNT(*) AS num_starts FROM `bigquery-public-data.london_bicycles.cycle_hire` GROUP BY start_station_name;
Run
Save query
Save view
Schedule query
More
This query will process 676.2 MB when run.
Query results
SAVE RESULTS
EXPLORE DATA
Query complete (0.9 sec elapsed, 676.2 MB processed)
Job information
Results
JSON
Execution details
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
Rows per page:
100
1 - 100 of 880
First page
Last page
Type here to search
ENG
16:54
11-08-2020

Fig7

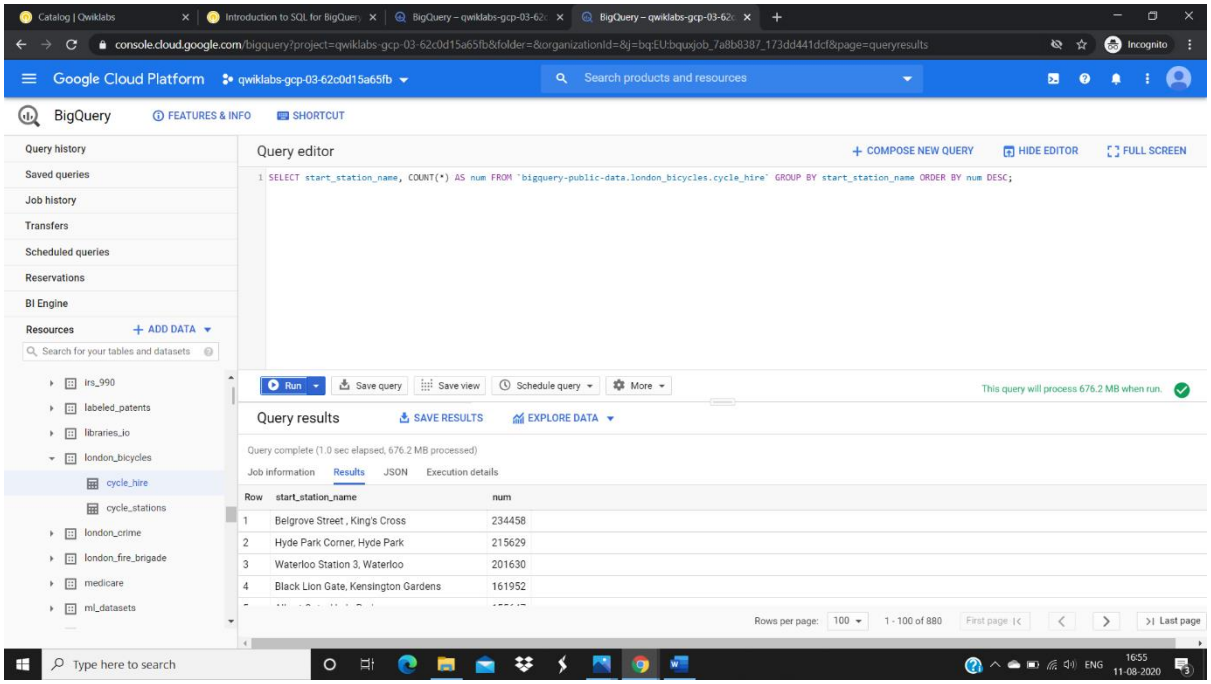

Google Cloud Platform
qwiklabs-gcp-03-62c0d15a65fb
Search products and resources
BigQuery
FEATURES & INFO
SHORTCUT
Query history
Saved queries
Job history
Transfers
Scheduled queries
Reservations
BI Engine
Resources
ADD DATA
Search for your tables and datasets
irs_990
labeled_patents
libraries_io
london_bicycles
cycle_hire
cycle_stations
london_crime
london_fire_brigade
medicare
ml_datasets
Query editor
COMPOSE NEW QUERY
HIDE EDITOR
FULL SCREEN
SELECT start_station_name, COUNT(*) AS num FROM `bigquery-public-data.london_bicycles.cycle_hire` GROUP BY start_station_name ORDER BY num DESC;
Run
Save query
Save view
Schedule query
More
This query will process 676.2 MB when run.
Query results
SAVE RESULTS
EXPLORE DATA
Query complete (1.0 sec elapsed, 676.2 MB processed)
Job information
Results
JSON
Execution details
Row
start_station_name
num
1
Belgrove 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
Rows per page:
100
1 - 100 of 880
First page
Last page
Type here to search
ENG
16:55
11-08-2020

Fig8

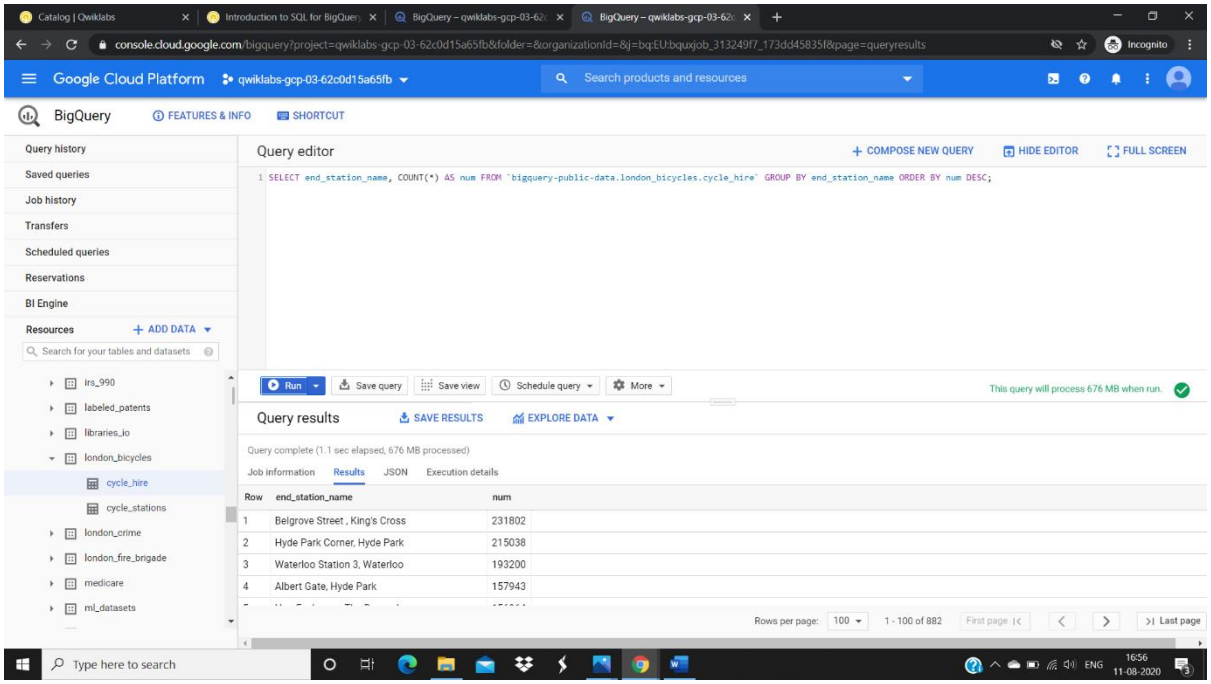


Fig9

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**

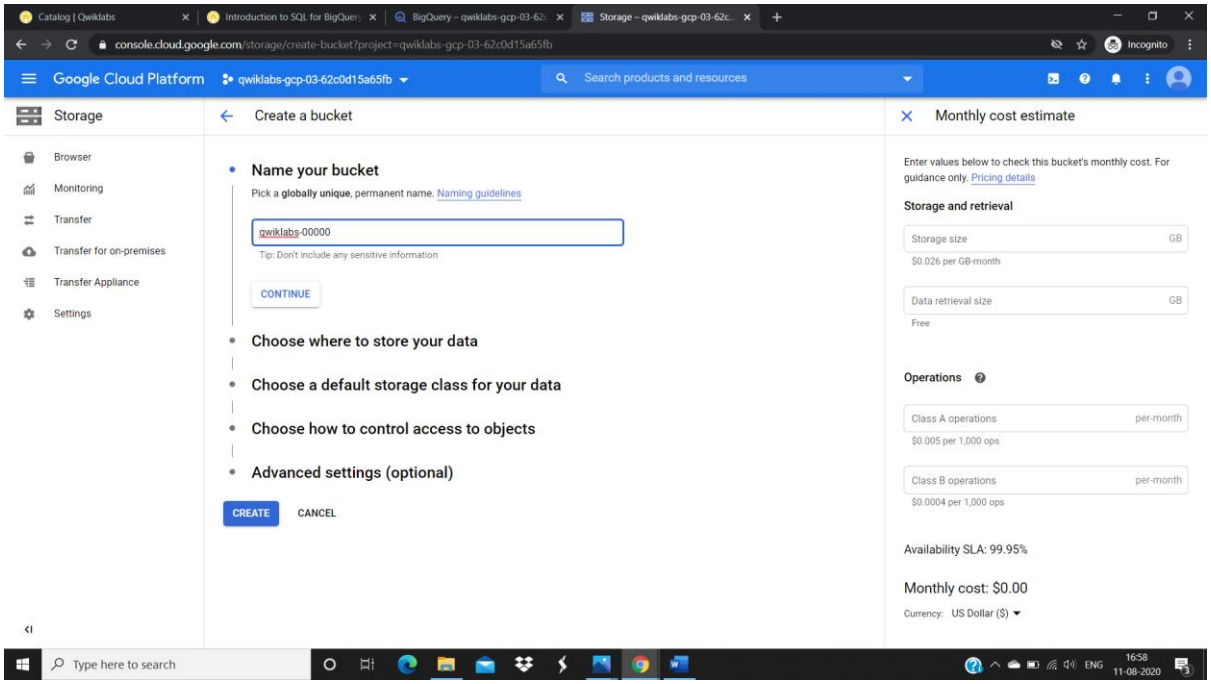


Fig10

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.

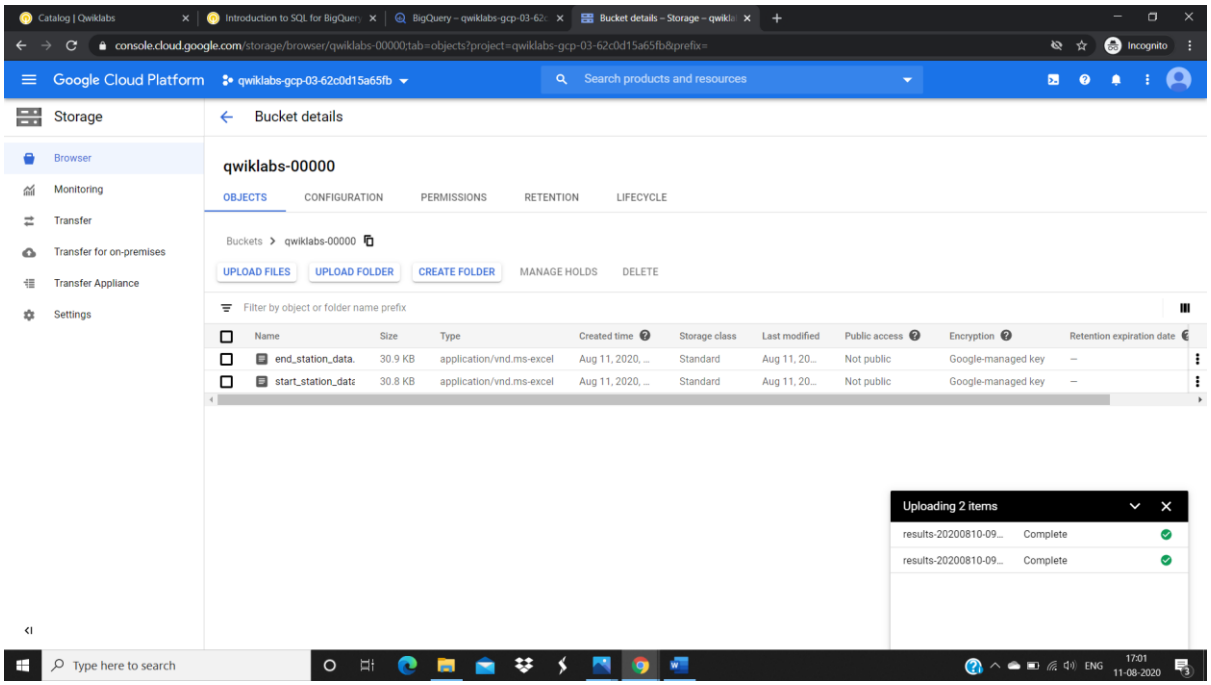


Fig11

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**

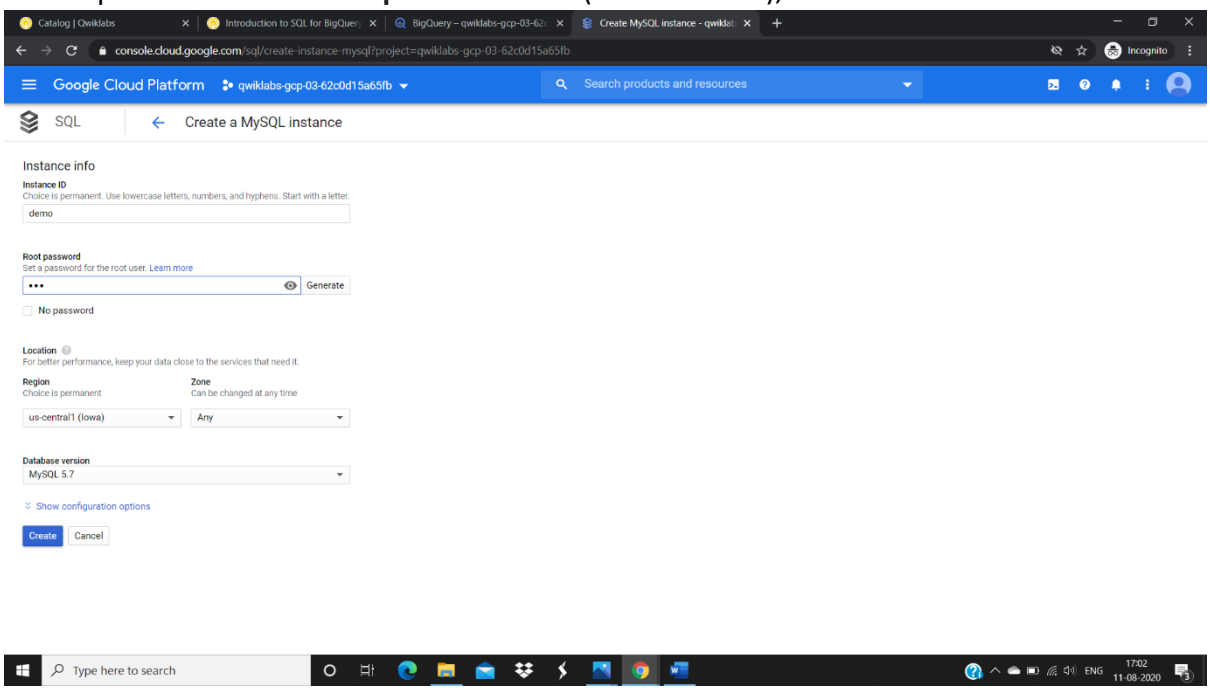


Fig12

Step – 8 : Click on the Cloud SQL instance

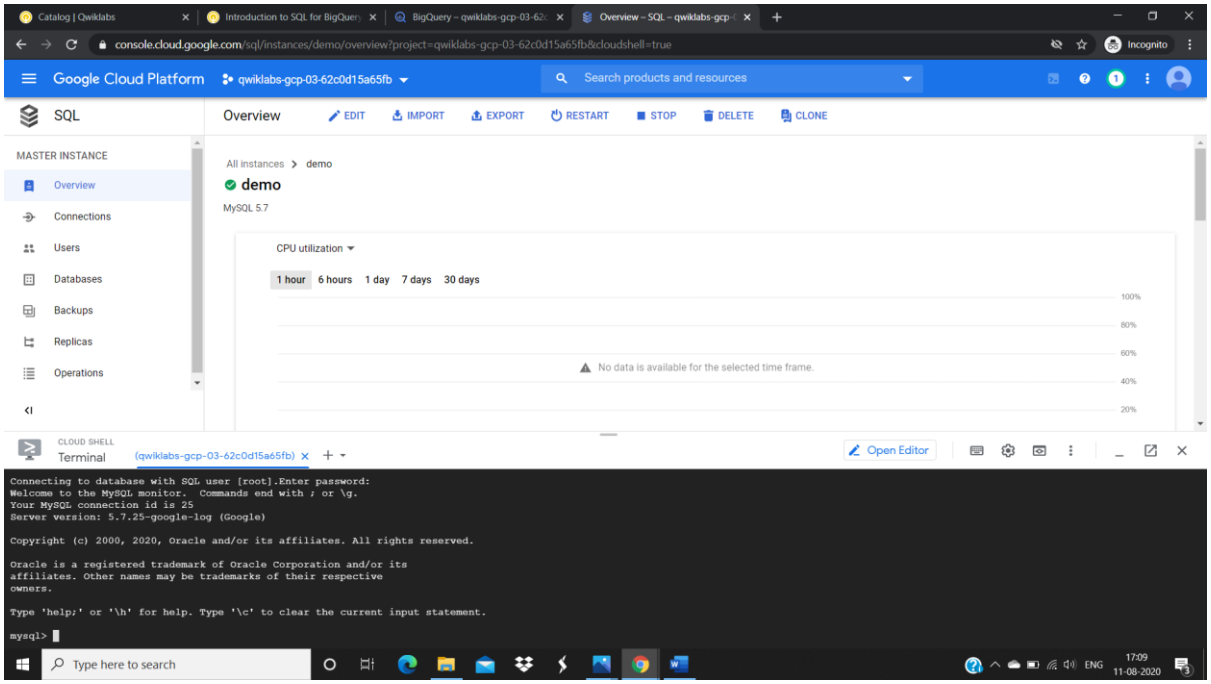


Fig13

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

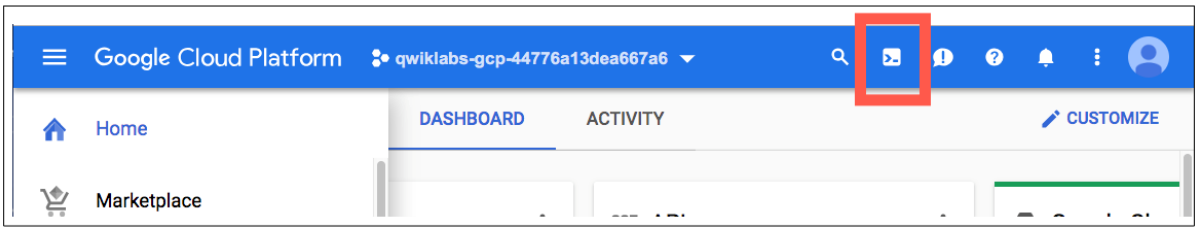


Fig14

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;

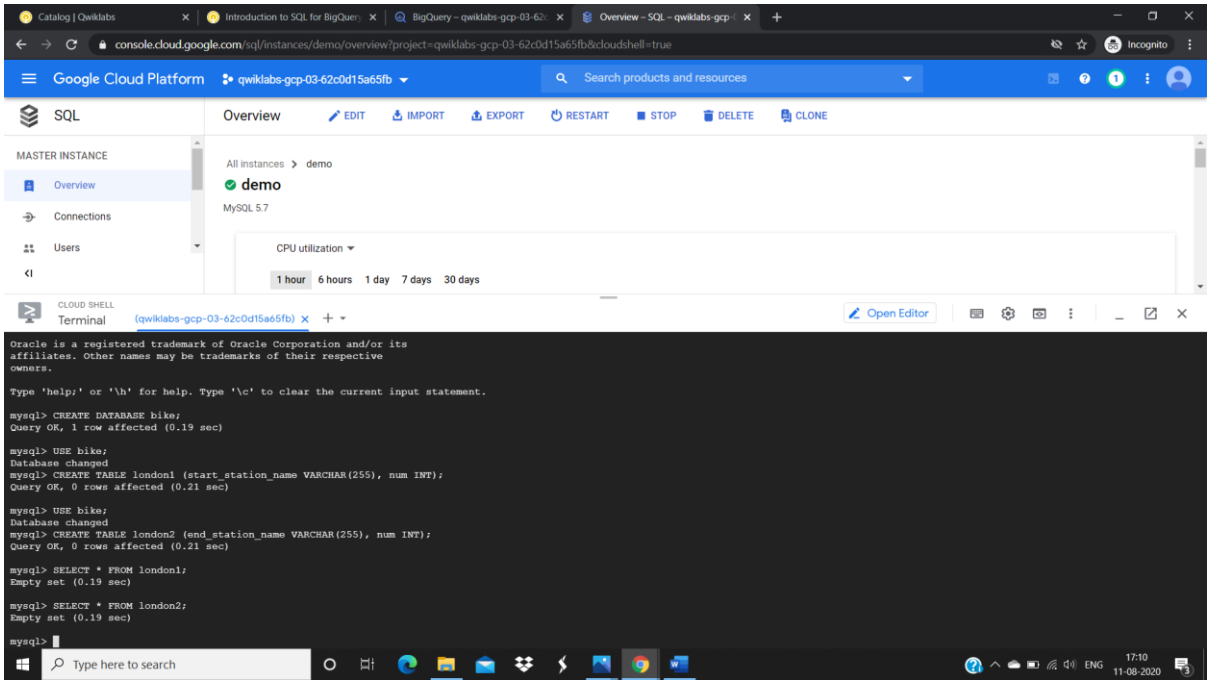


Fig15

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**:

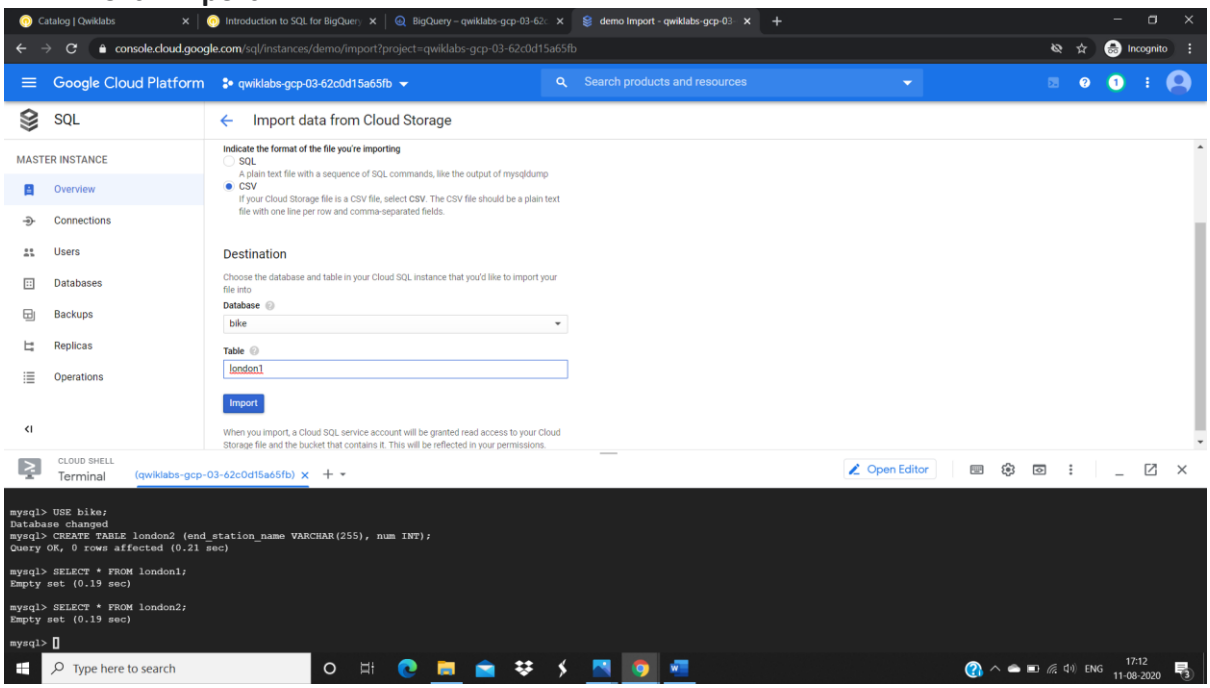


Fig16

Do the same for the other CSV file

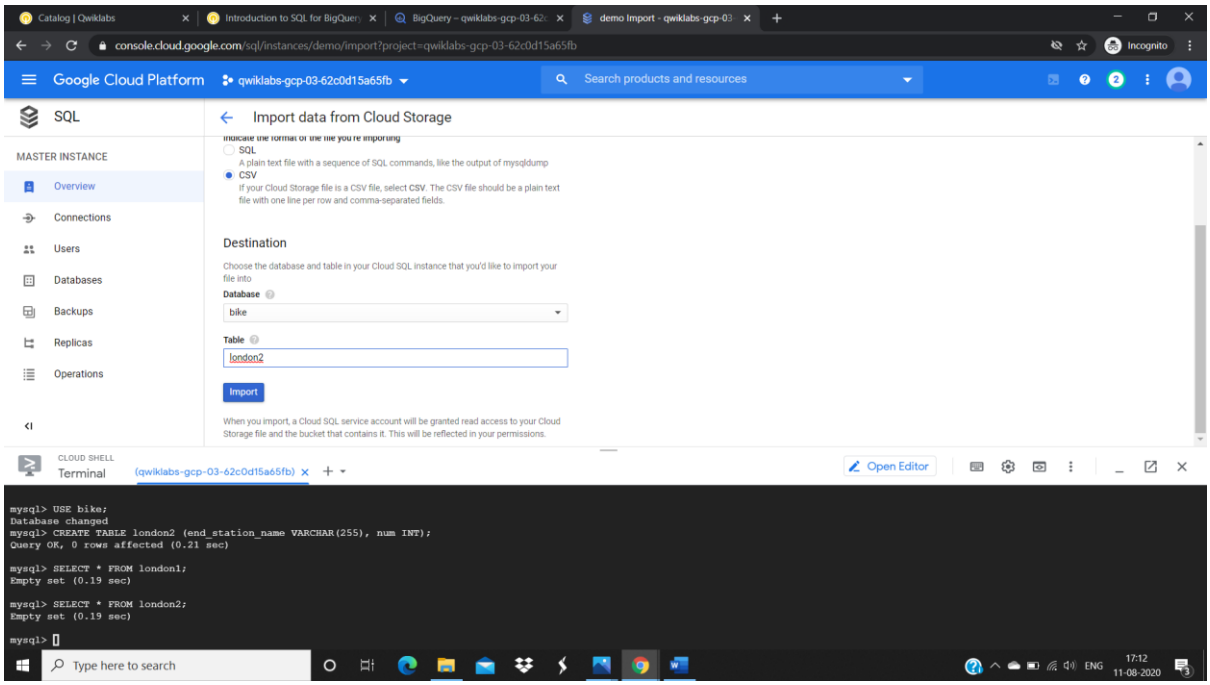


Fig17

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;

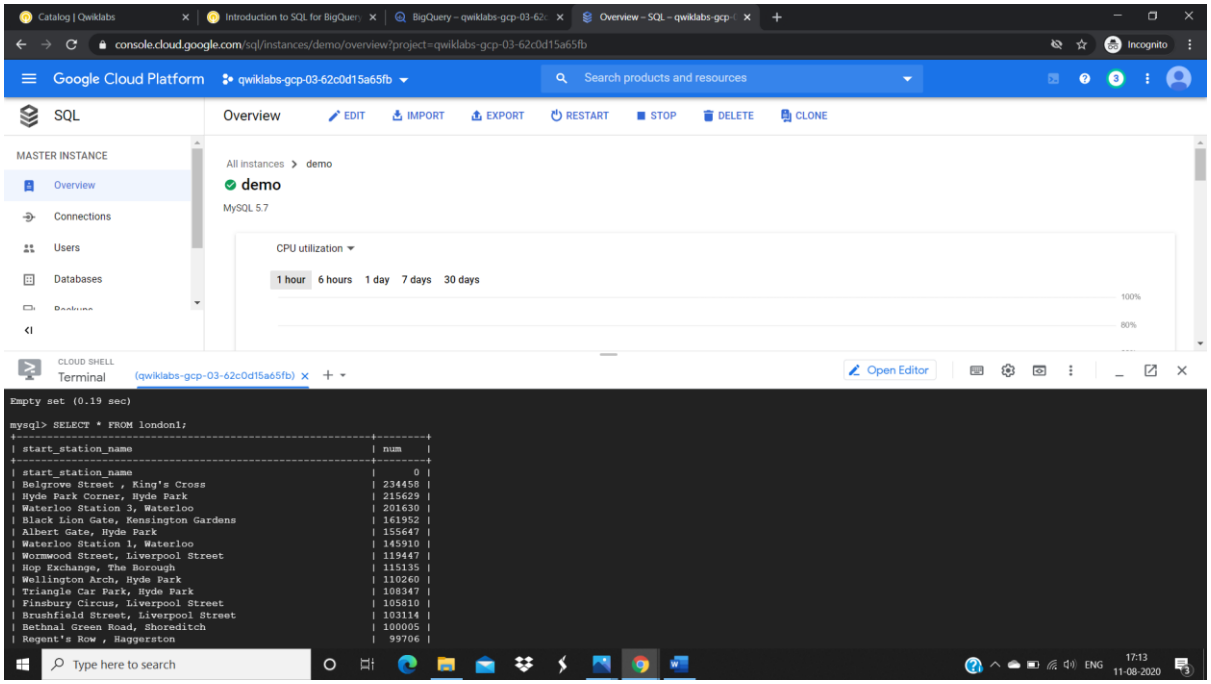


Fig18

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;

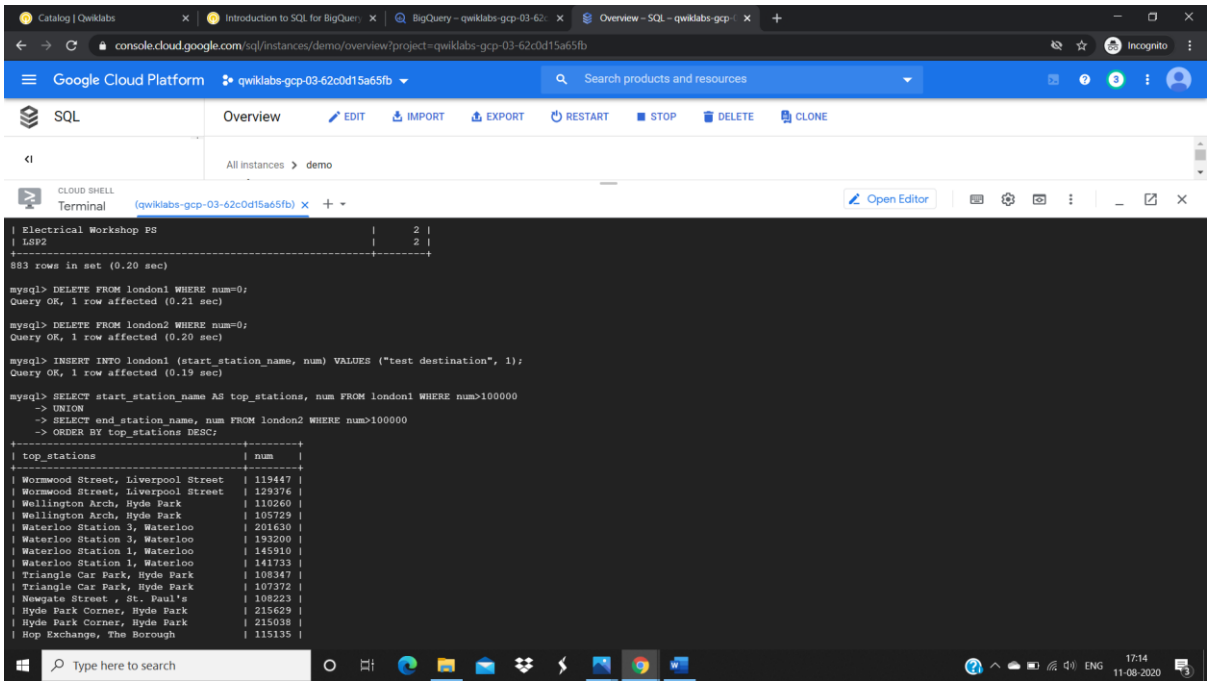


Fig19

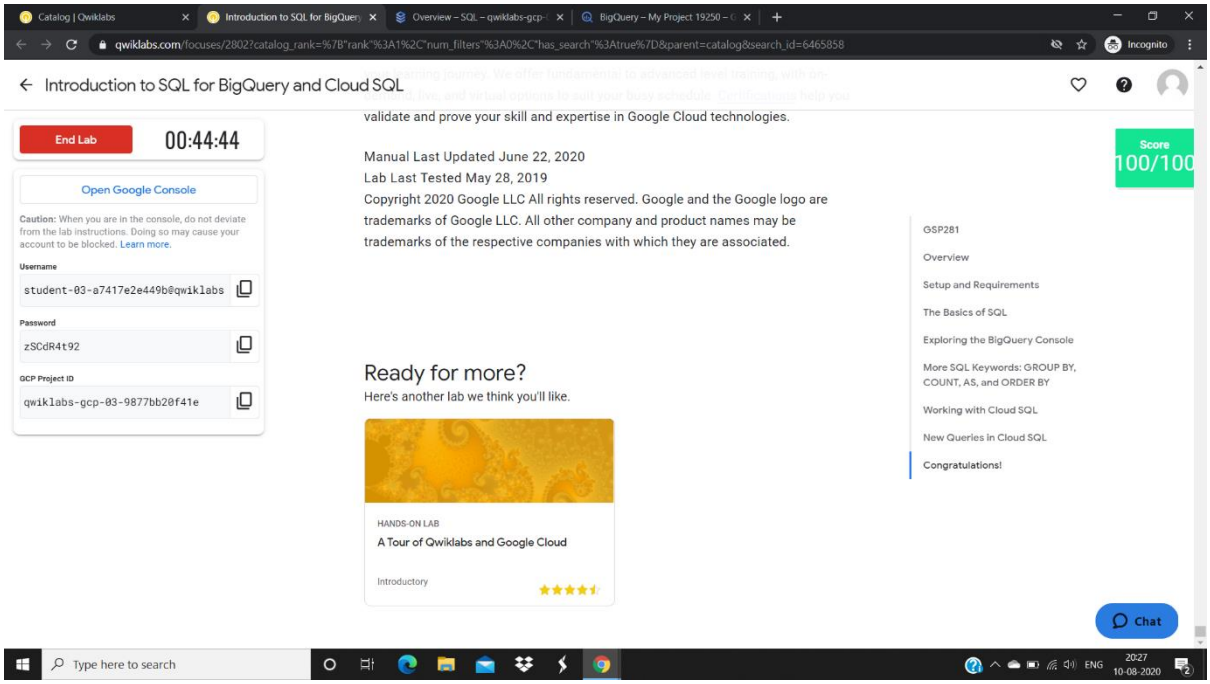

Introduction to SQL for BigQuery and Cloud SQL
End Lab
00:44:44
Open Google Console
Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.
Username
student-03-a7417e2e449b@qwiklabs
Password
zSCdR4t92
GCP Project ID
qwiklabs-gcp-03-9877bb20f41e
validate and prove your skill and expertise in Google Cloud technologies.
Manual Last Updated June 22, 2020
Lab Last Tested May 28, 2019
Copyright 2020 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.
Score
100/100
GSP281
Overview
Setup and Requirements
The Basics of SQL
Exploring the BigQuery Console
More SQL Keywords: GROUP BY, COUNT, AS, and ORDER BY
Working with Cloud SQL
New Queries in Cloud SQL
Congratulations!
Ready for more?
Here's another lab we think you'll like.
HANDS-ON LAB
A Tour of Qwiklabs and Google Cloud
Introductory
Chat
Type here to search
ENG
20:27
10-08-2020

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.

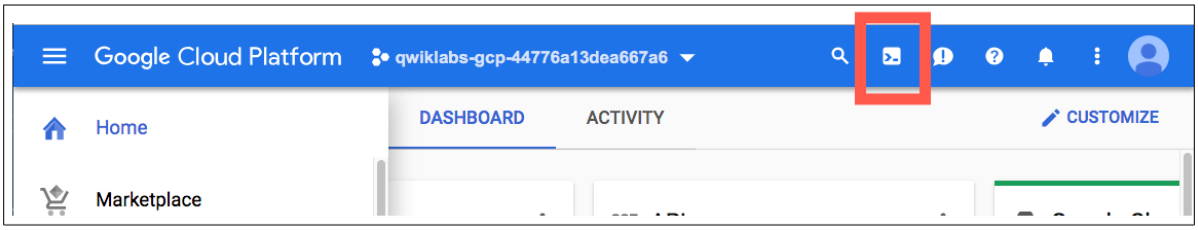


Fig1

Then Click Continue

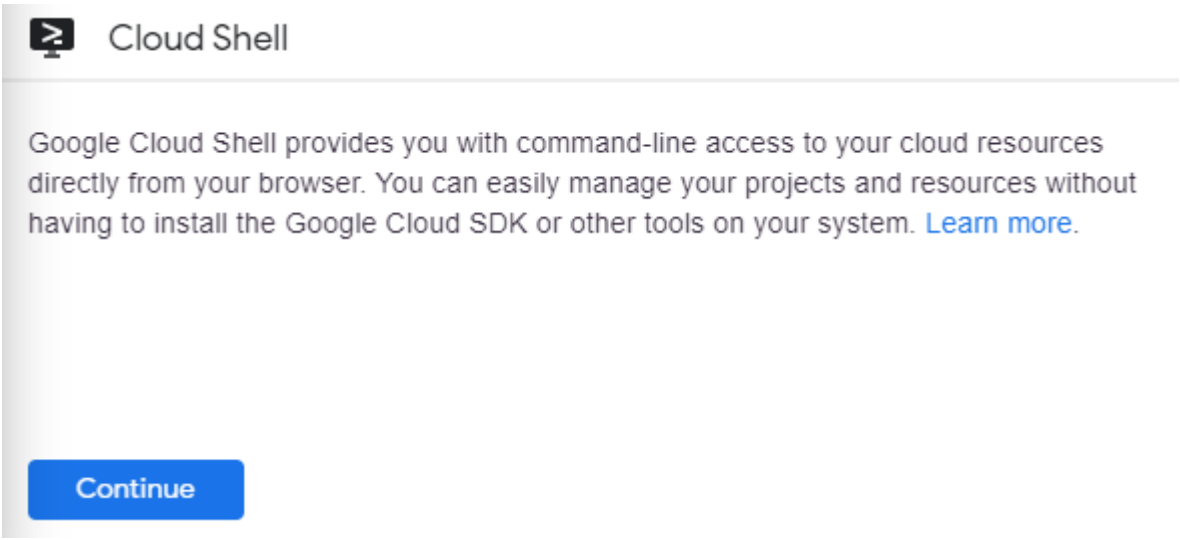


Fig2

Use the following commands

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

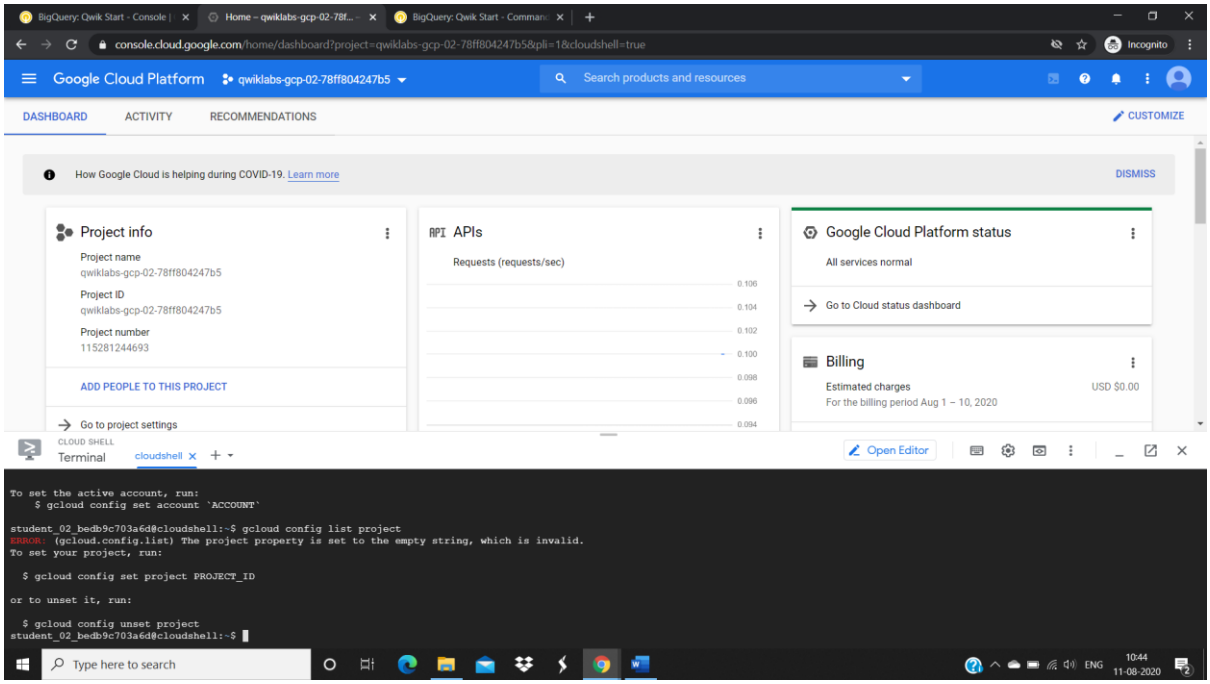


Fig3

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

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

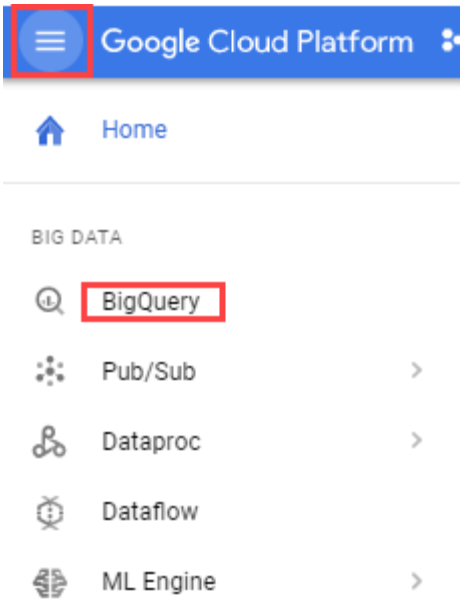


Fig4

Query into the BigQuery Query editor. Using following commands:

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

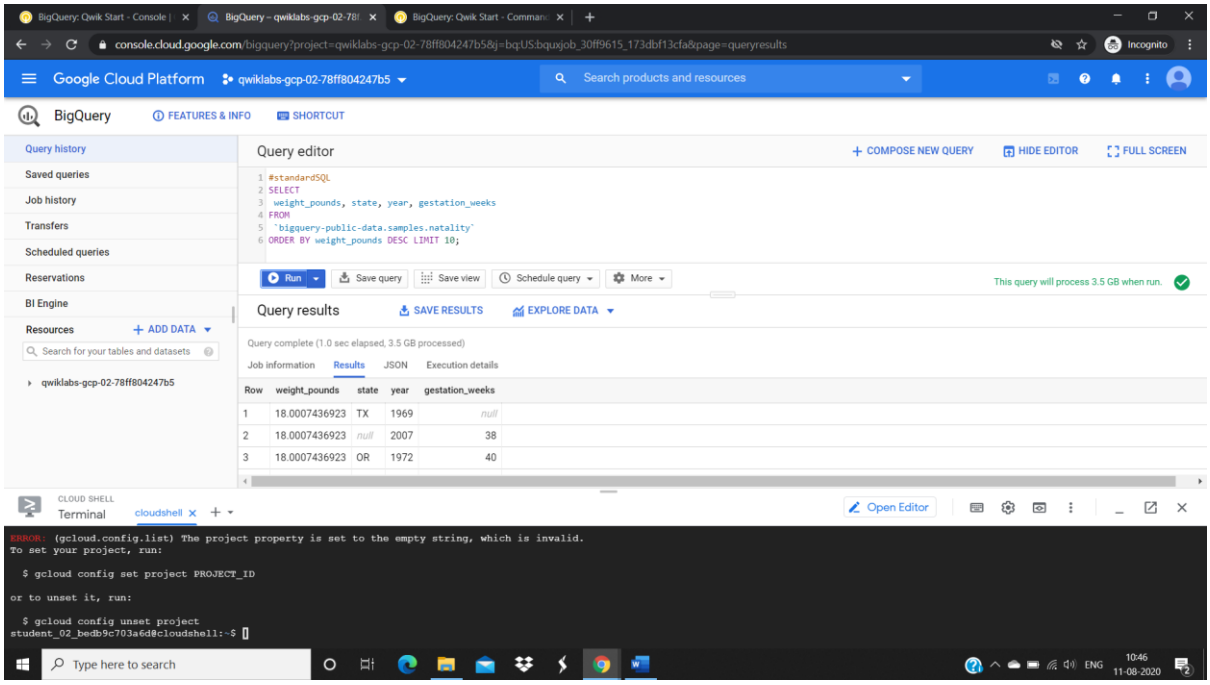


Fig5

Click Run

Query results 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

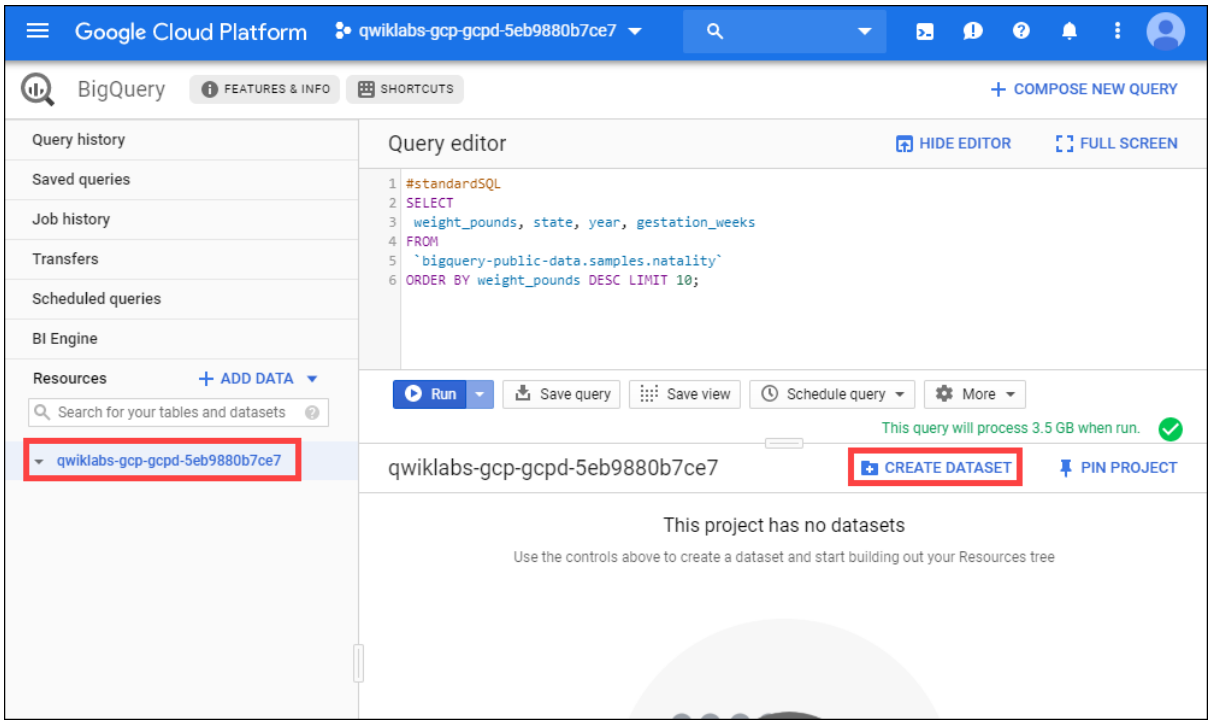


Fig7

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

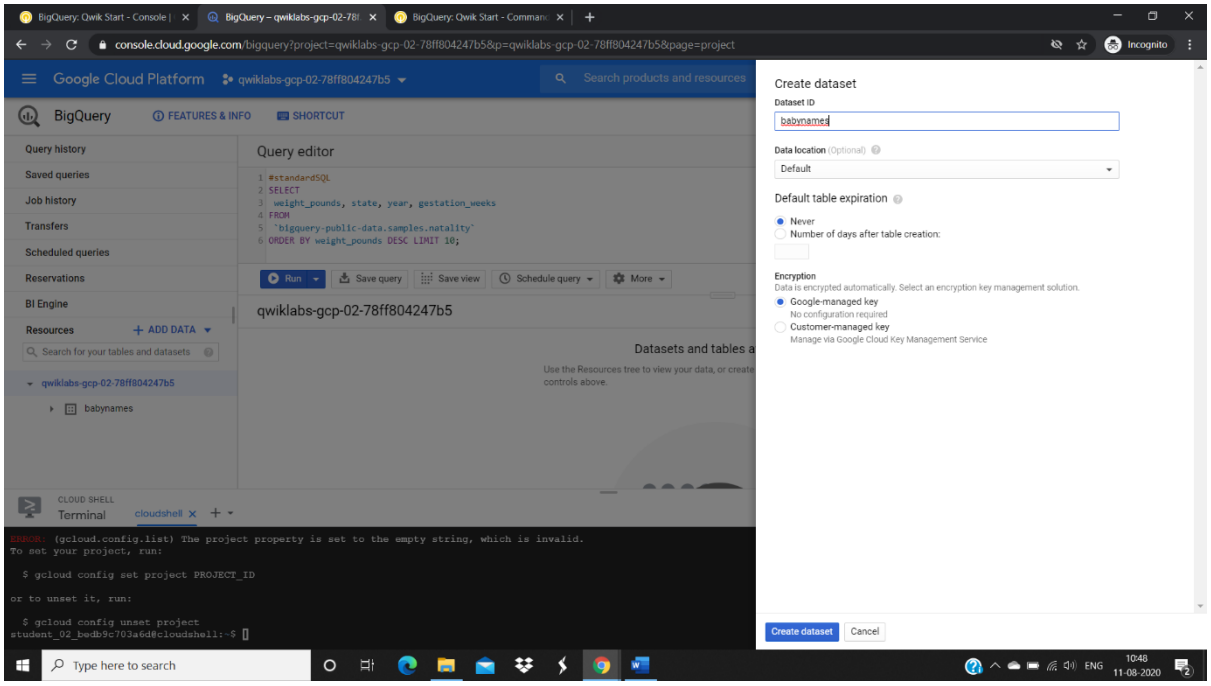


Fig8

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

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

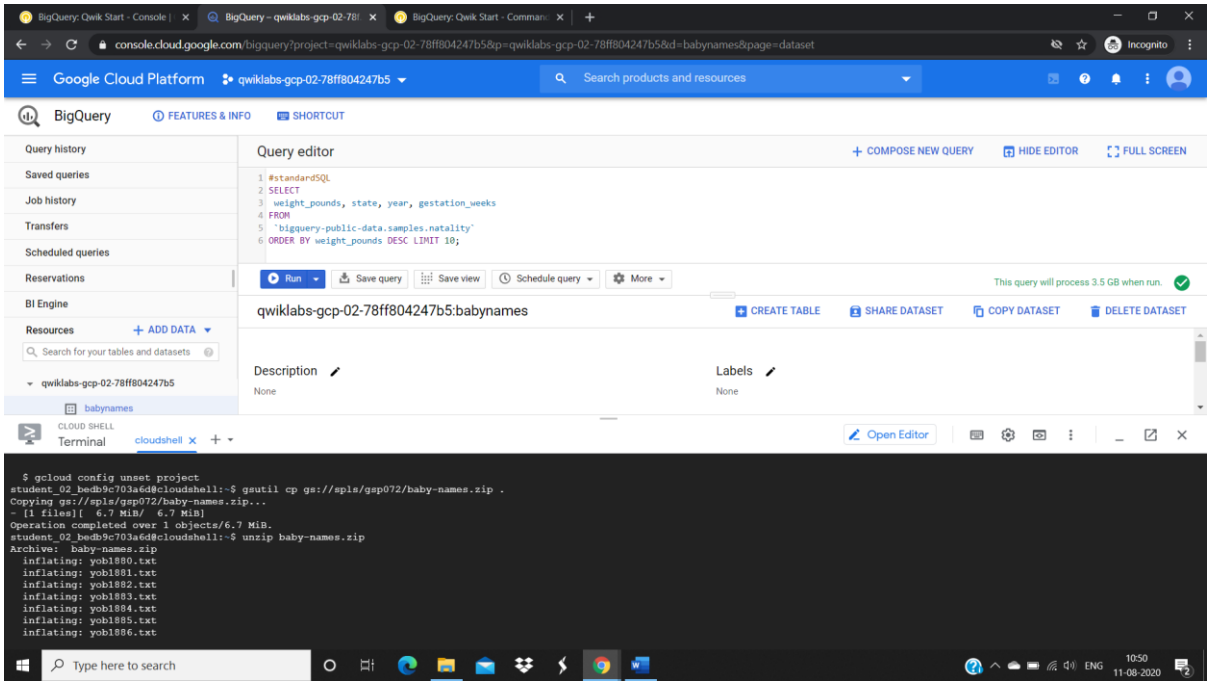


Fig9

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**.

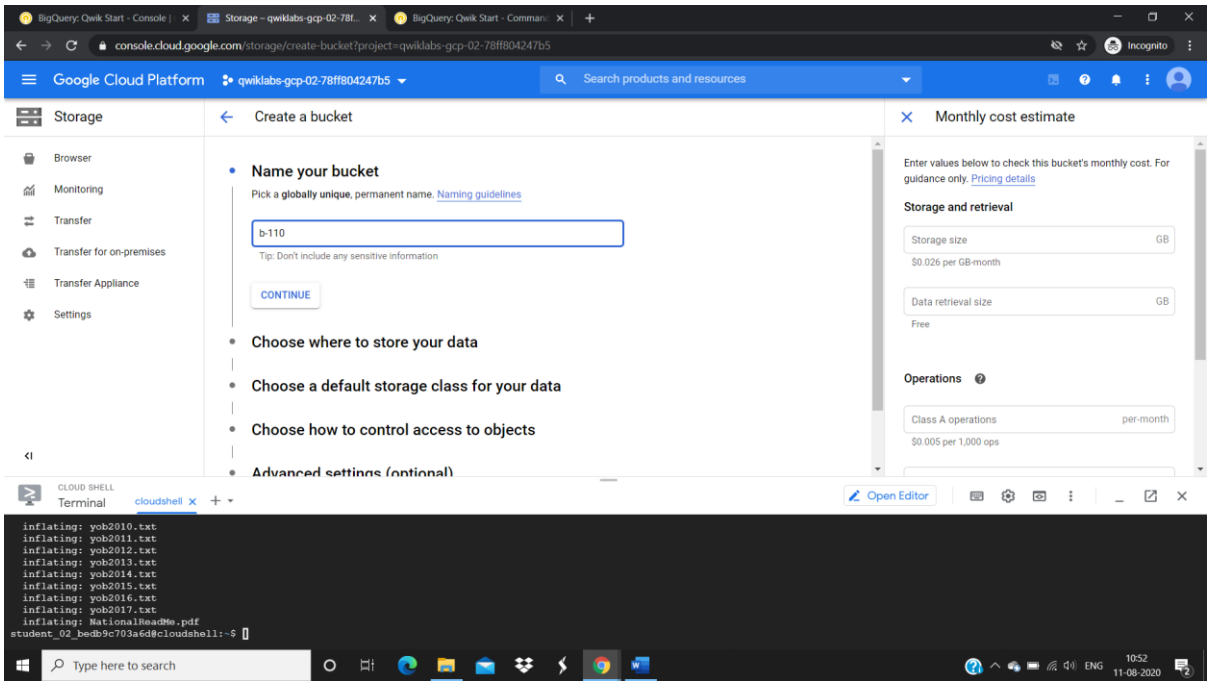


Fig10

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>

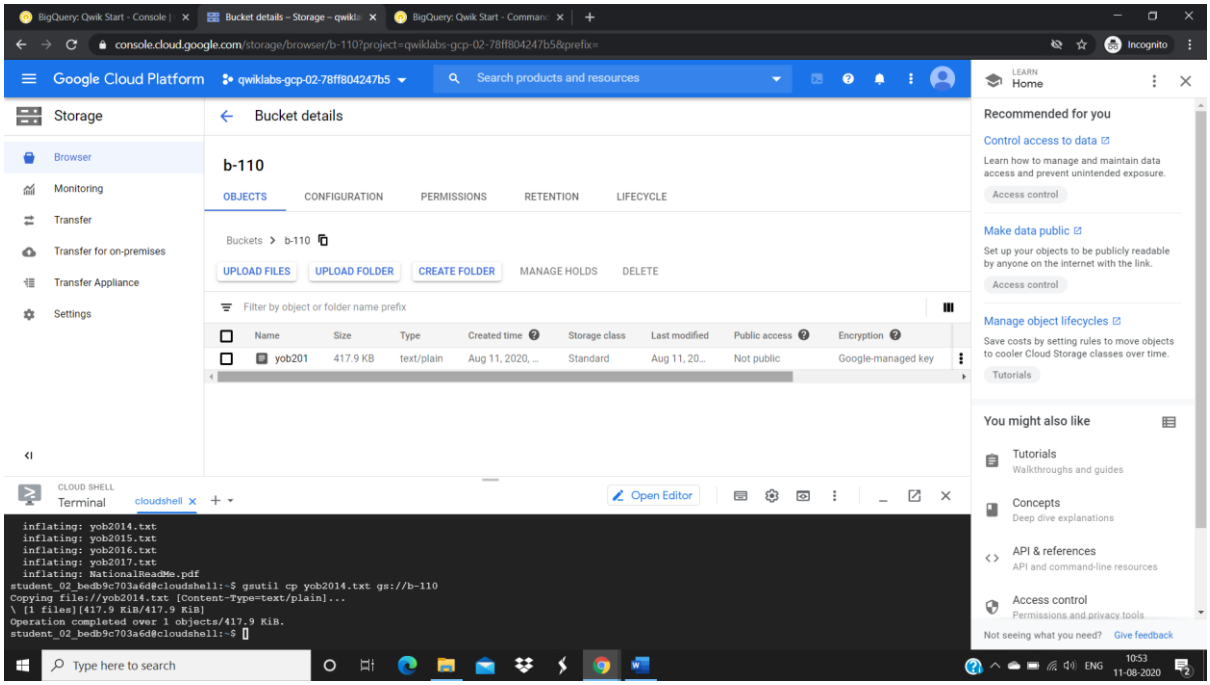


Fig11

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**.

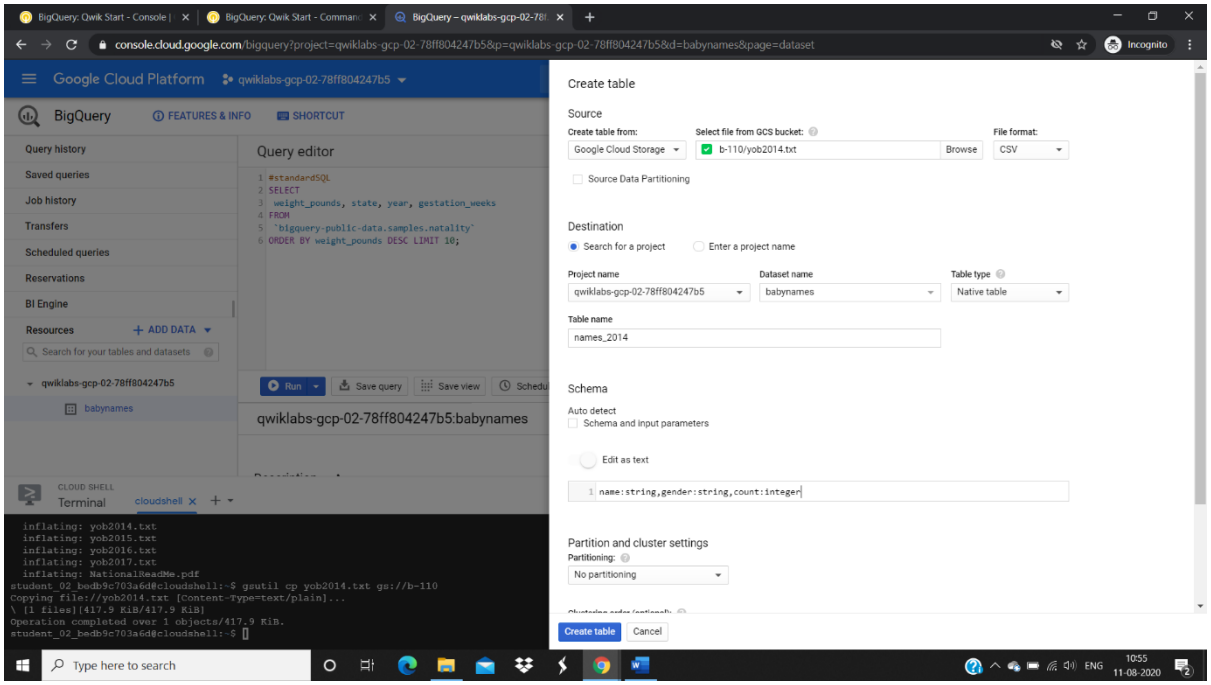


Fig12

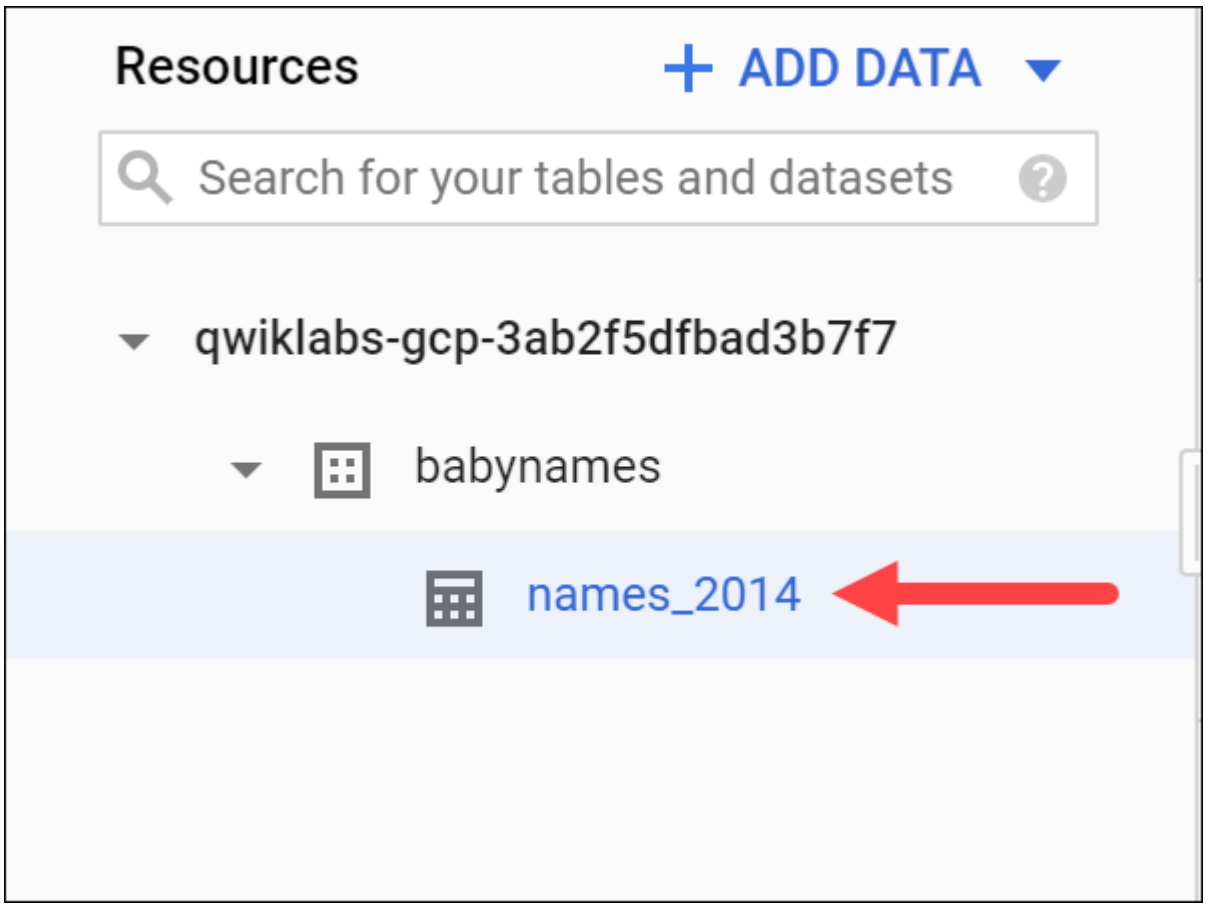

Resources
ADD DATA
Search for your tables and datasets
qwiklabs-gcp-3ab2f5dfbad3b7f7
babynames
names_2014

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.

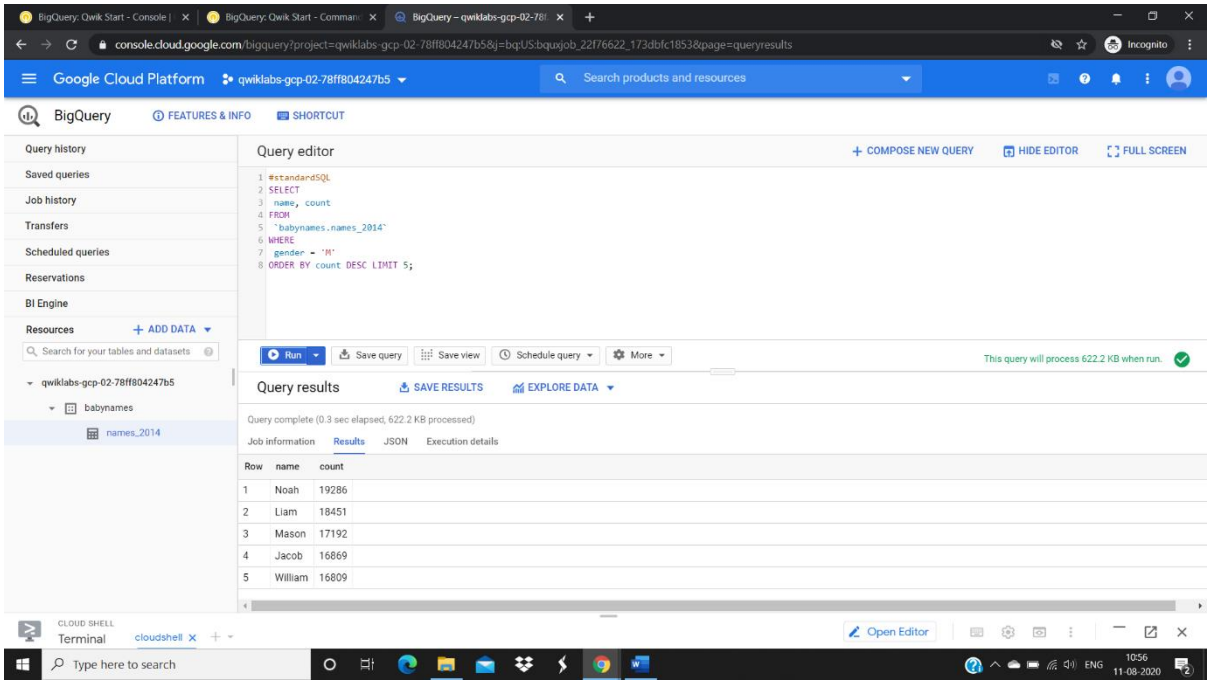


Fig14

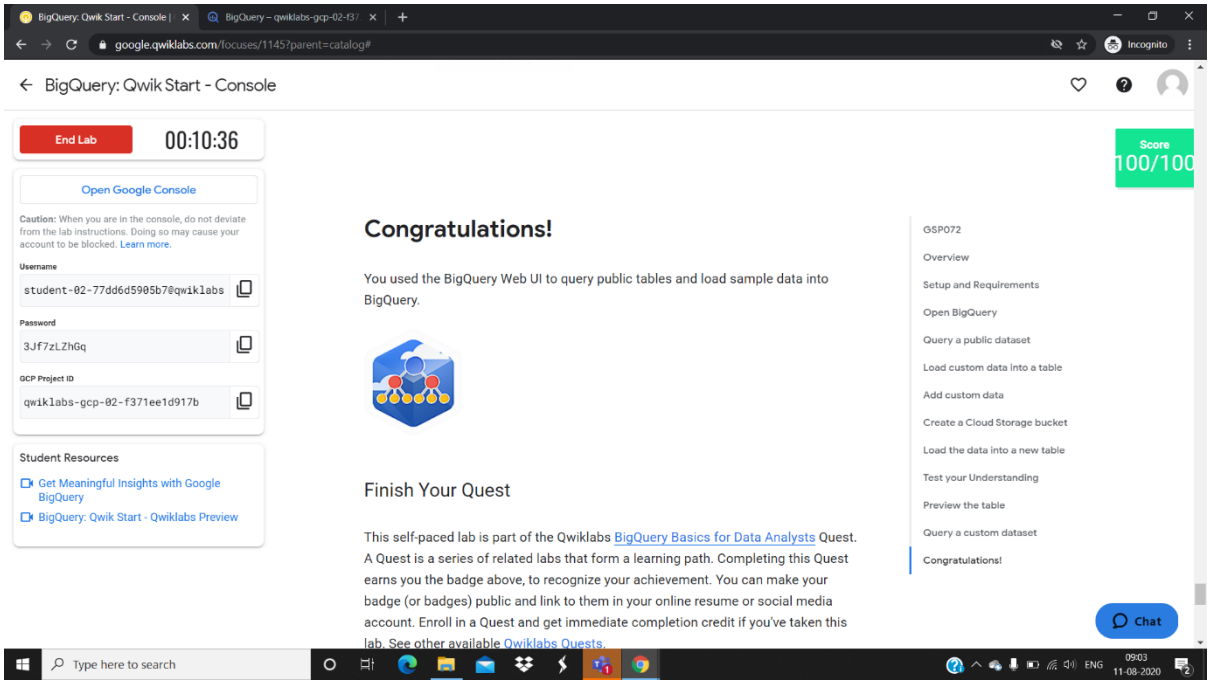

BigQuery: Qwik Start - Console
google.qwiklabs.com/focuses/1145?parent=catalog#
Incognito
BigQuery: Qwik Start - Console
End Lab
00:10:36
Open Google Console
Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.
Username
student-02-77dd6d5905b7@qwiklabs
Password
3Jf7zLZhGq
GCP Project ID
qwiklabs-gcp-02-f371ee1d917b
Student Resources
Get Meaningful Insights with Google BigQuery
BigQuery: Qwik Start - Qwiklabs Preview
Score
100/100
Congratulations!
You used the BigQuery Web UI to query public tables and load sample data into BigQuery.
Finish Your Quest
This self-paced lab is part of the Qwiklabs BigQuery Basics for Data Analysts Quest. A Quest is a series of related labs that form a learning path. Completing this Quest earns you the badge above, to recognize your achievement. You can make your badge (or badges) public and link to them in your online resume or social media account. Enroll in a Quest and get immediate completion credit if you've taken this lab. See other available Qwiklabs Quests.
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
Congratulations!
Chat
Type here to search
ENG
09:03
11-08-2020

Fig15

BigQuery: Qwik Start – Command Line

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

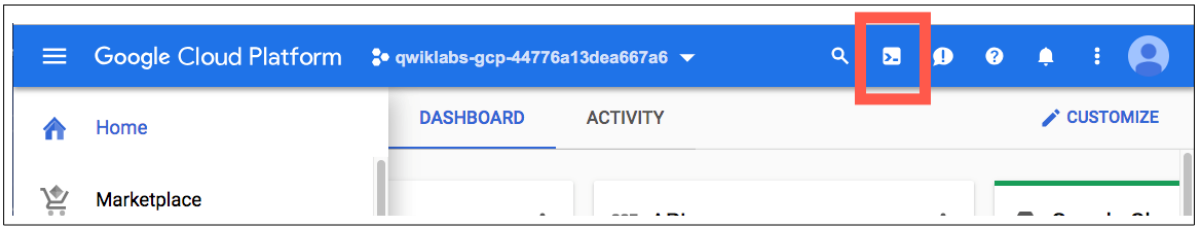


Fig1

Then Click Continue

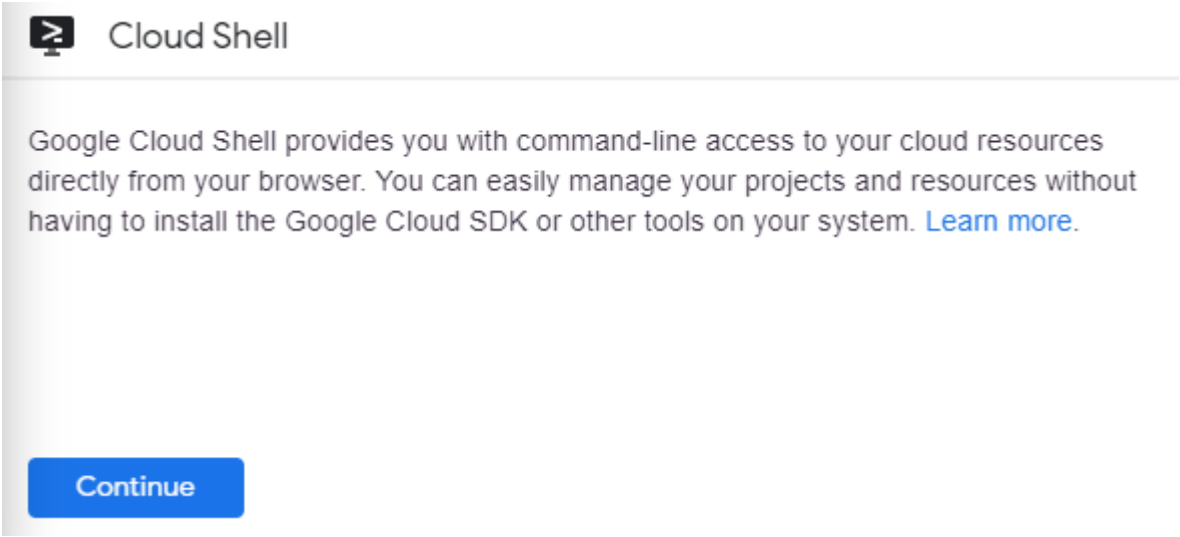


Fig2

Use the following commands

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

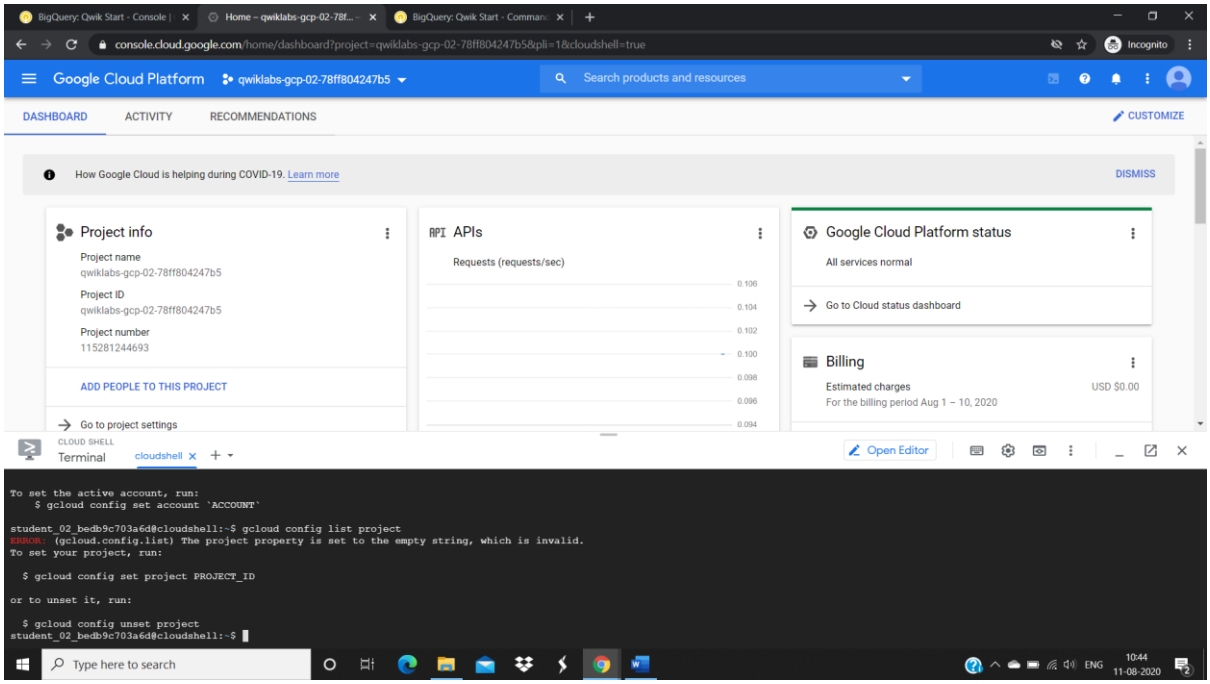


Fig3

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 bigquery-public-data:samples.shakespeare

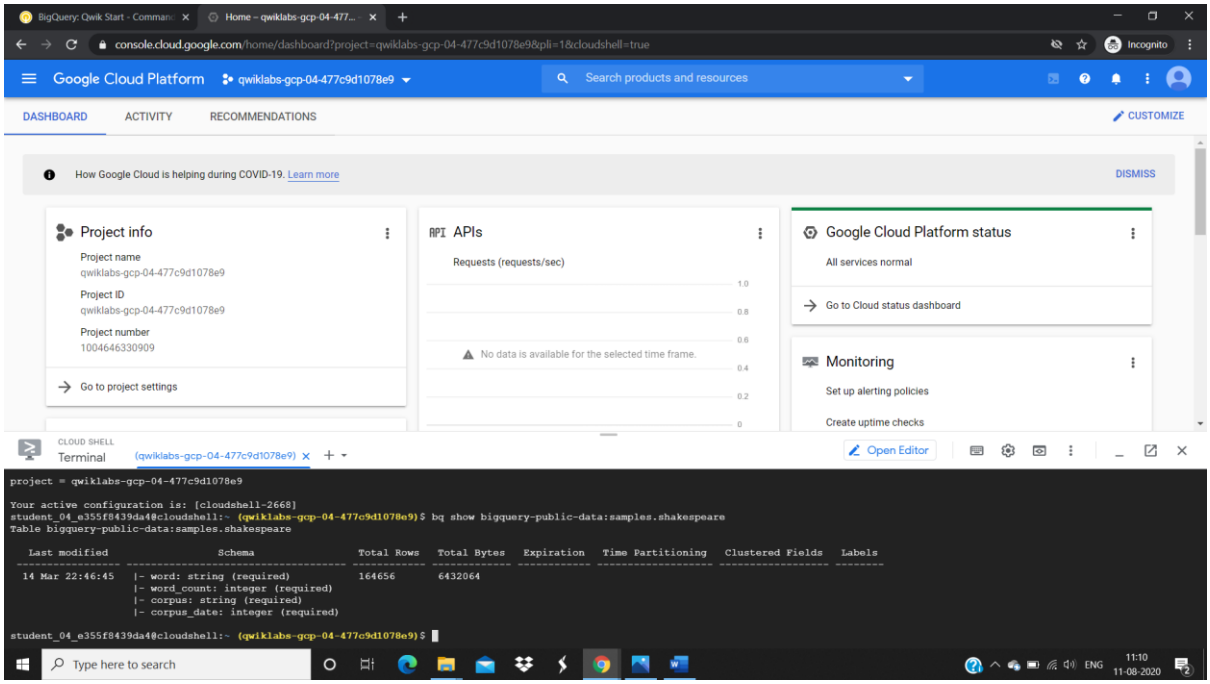


Fig4

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

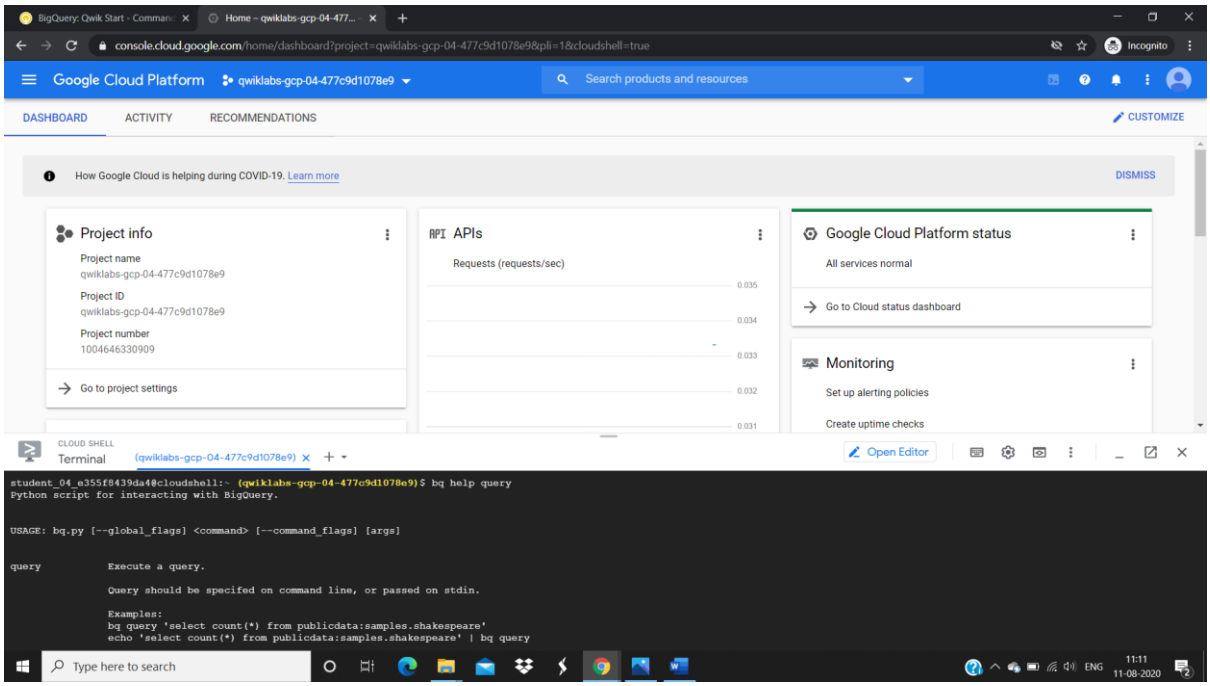


Fig5

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’

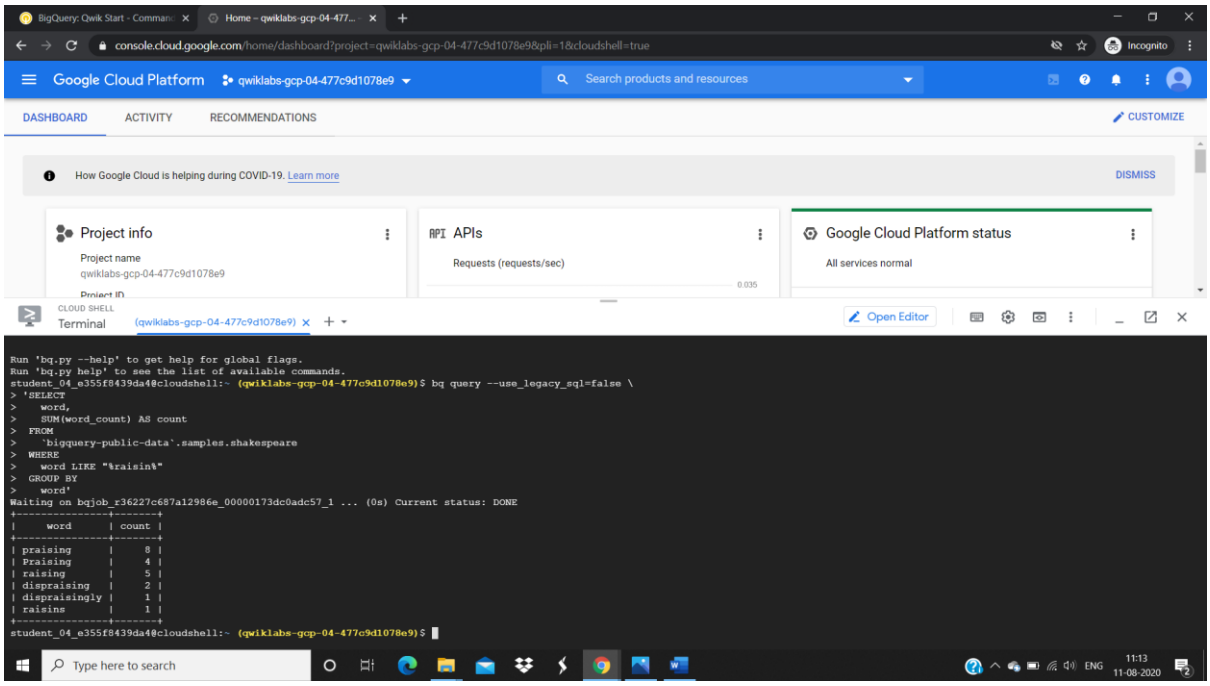


Fig6

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 bigquery-public-data Project ID to list the datasets in that specific project, followed by a colon (:).

- $bq ls bigquery-public-data:

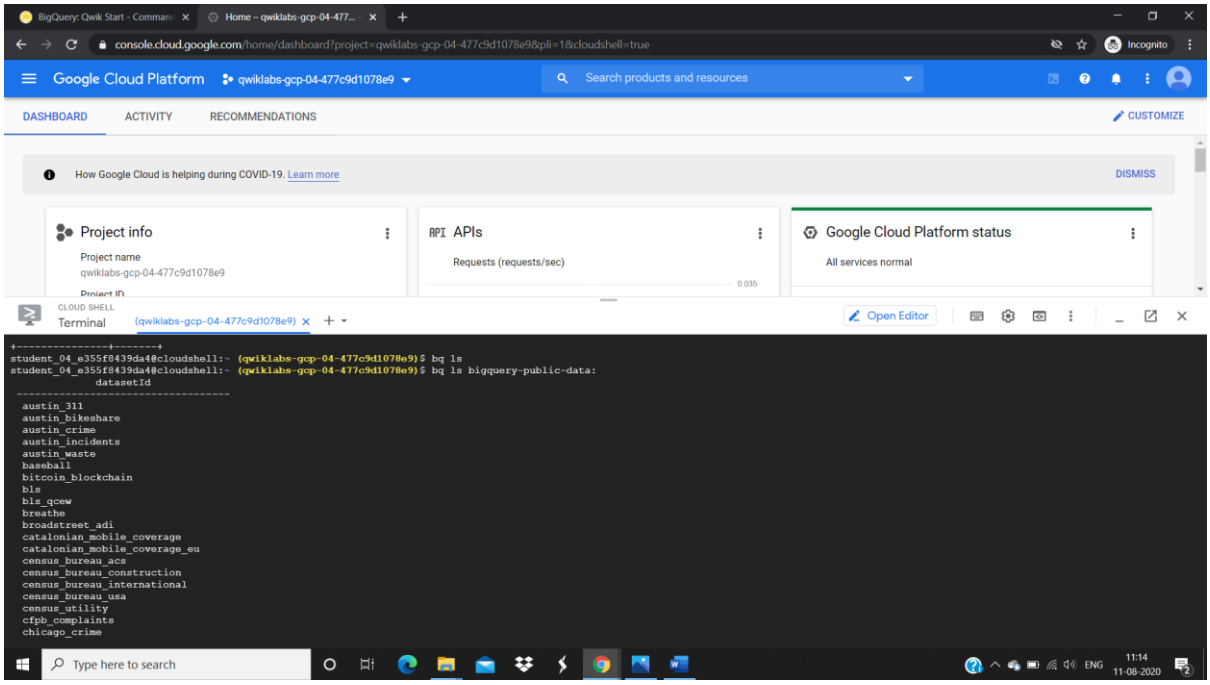


Fig7

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

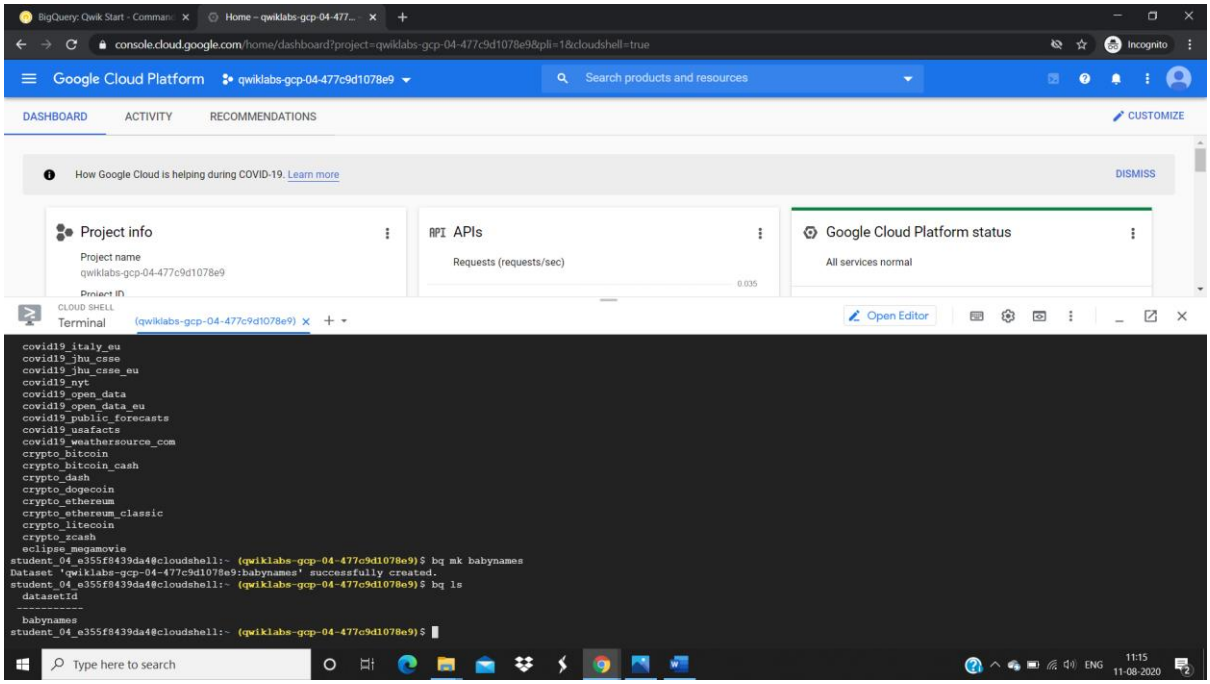


Fig8

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

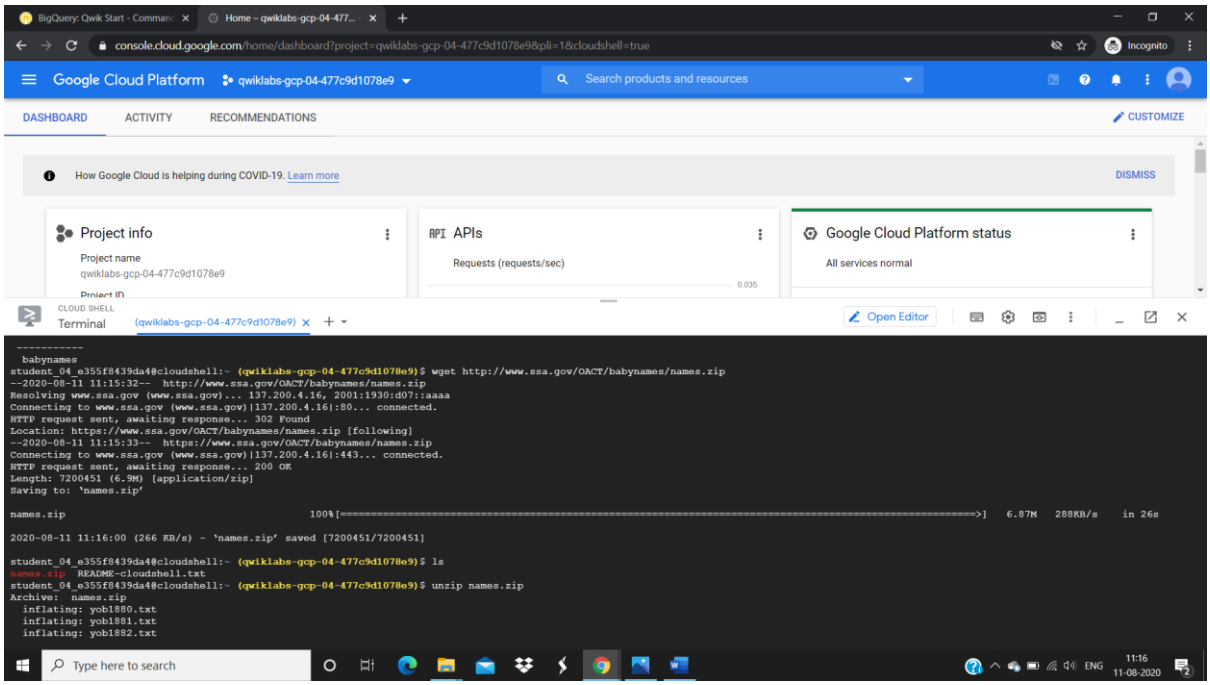


Fig9

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

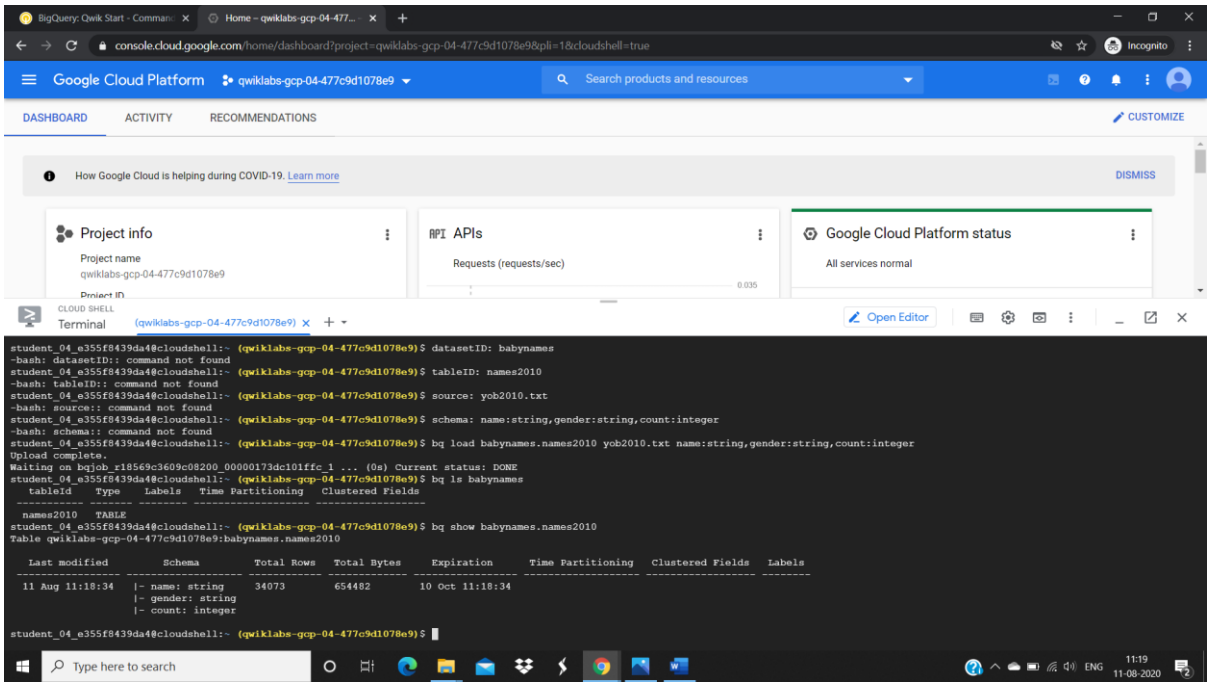


Fig10

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”

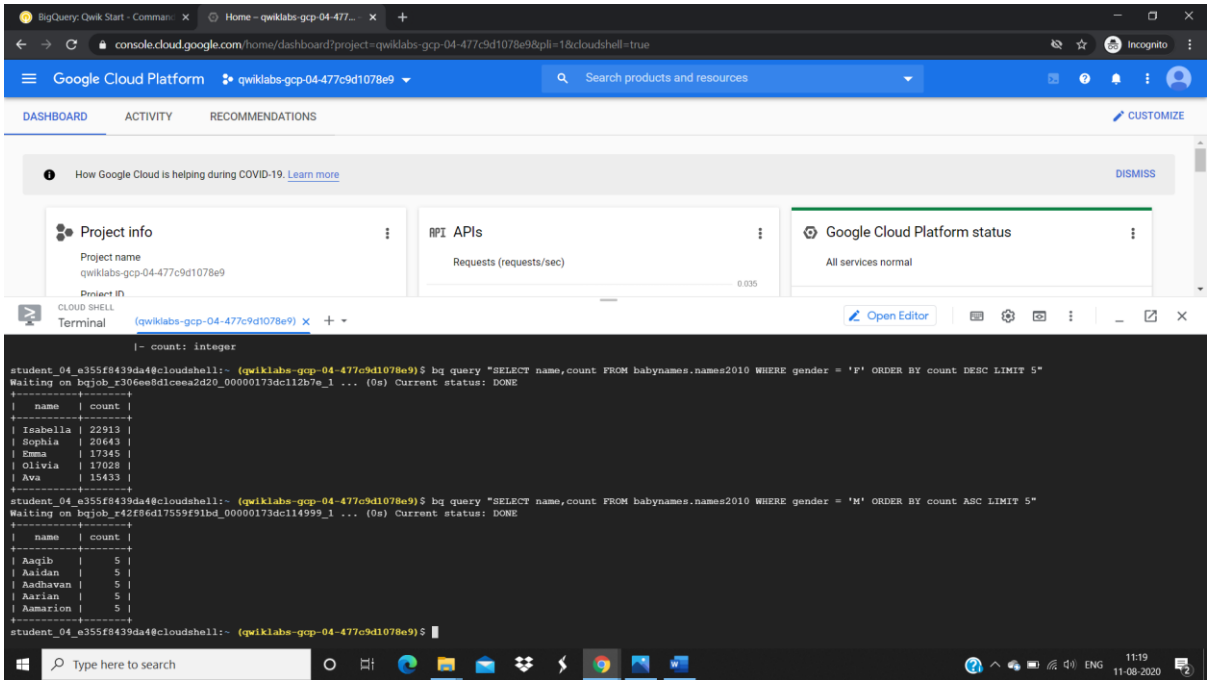


Fig11

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".

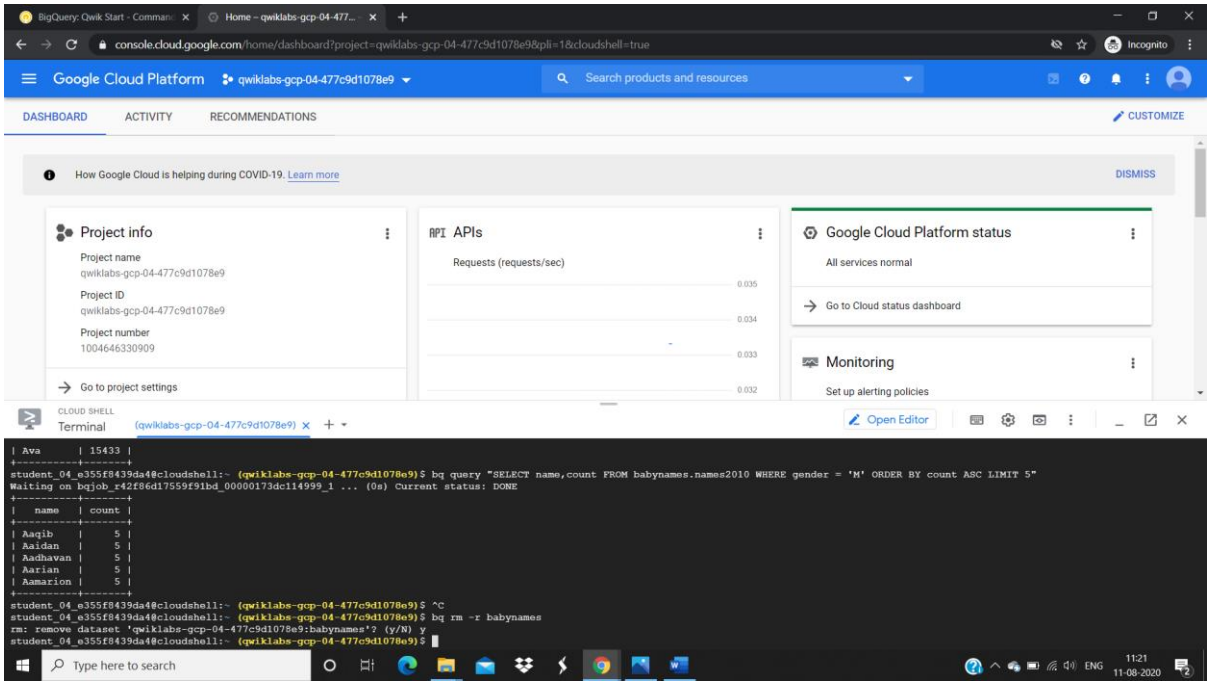


Fig12

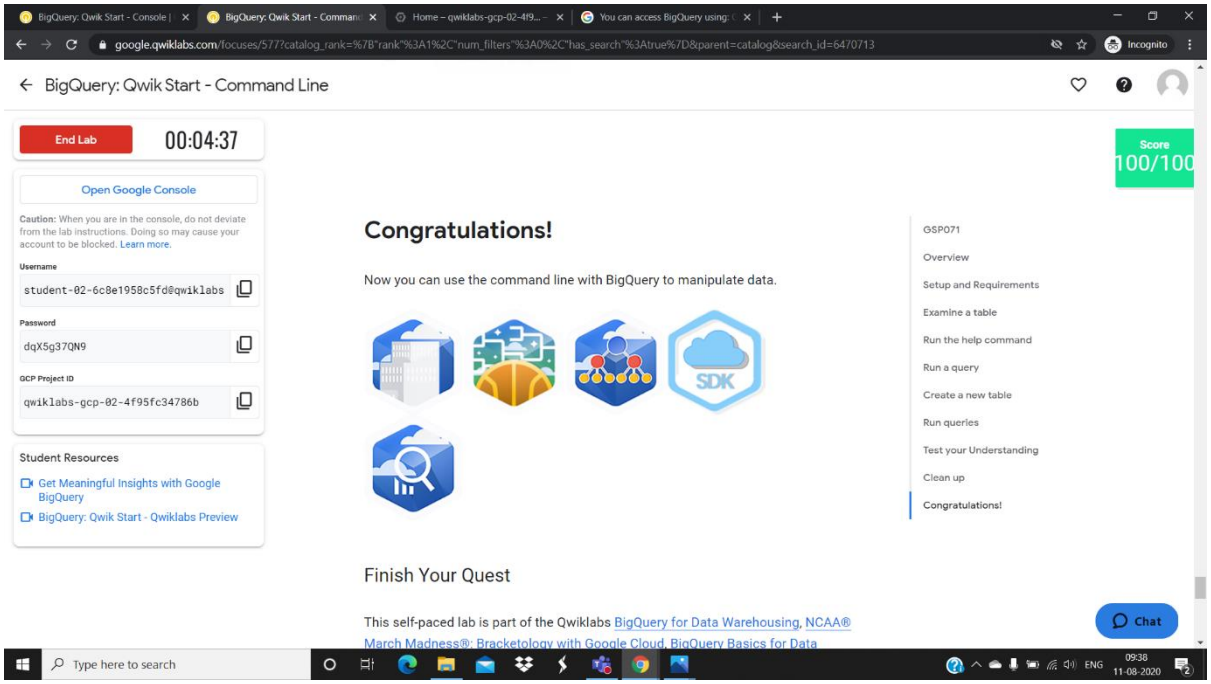

BigQuery: Qwik Start - Command Line
End Lab
00:04:37
Open Google Console
Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.
Username
student-02-6c8e1958c5fd@qwiklabs
Password
dqX5g37QN9
GCP Project ID
qwiklabs-gcp-02-4f95fc34786b
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.
SDK
GSP071
Overview
Setup and Requirements
Examine a table
Run the help command
Run a query
Create a new table
Run queries
Test your Understanding
Clean up
Congratulations!
Finish Your Quest
This self-paced lab is part of the Qwiklabs BigQuery for Data Warehousing, NCAA® March Madness®: Bracketology with Google Cloud, BigQuery Basics for Data
Chat
Type here to search
ENG
09:38
11-08-2020

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

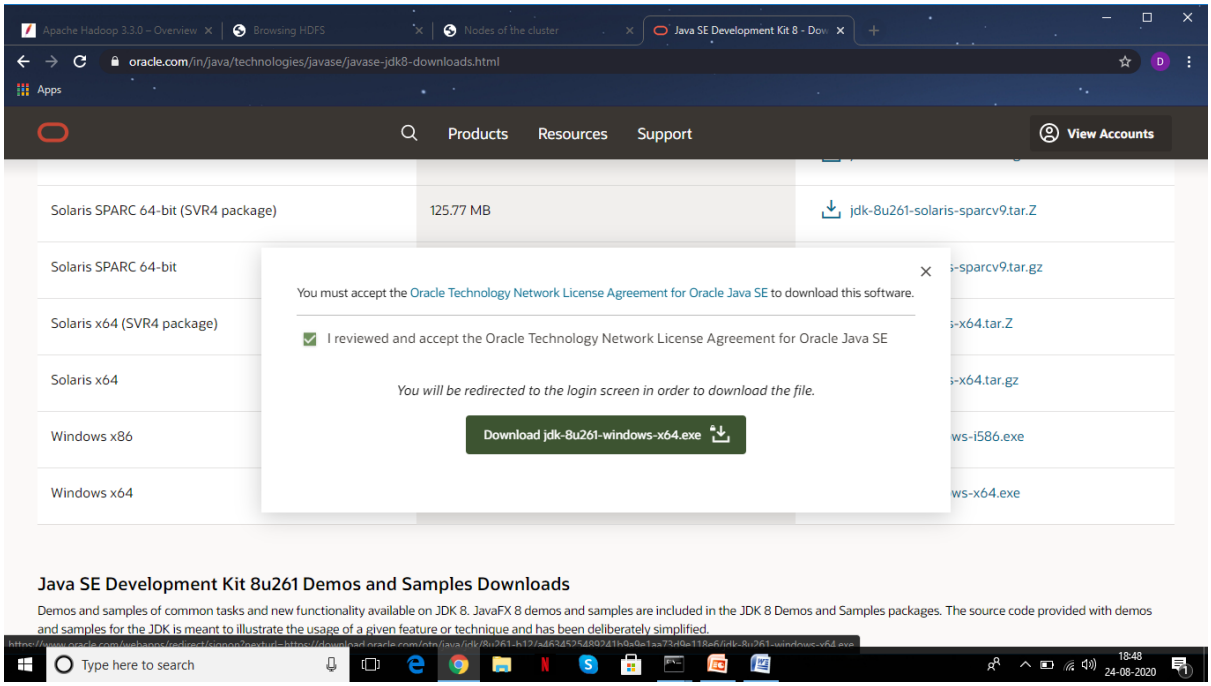


Fig1

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_261and 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.

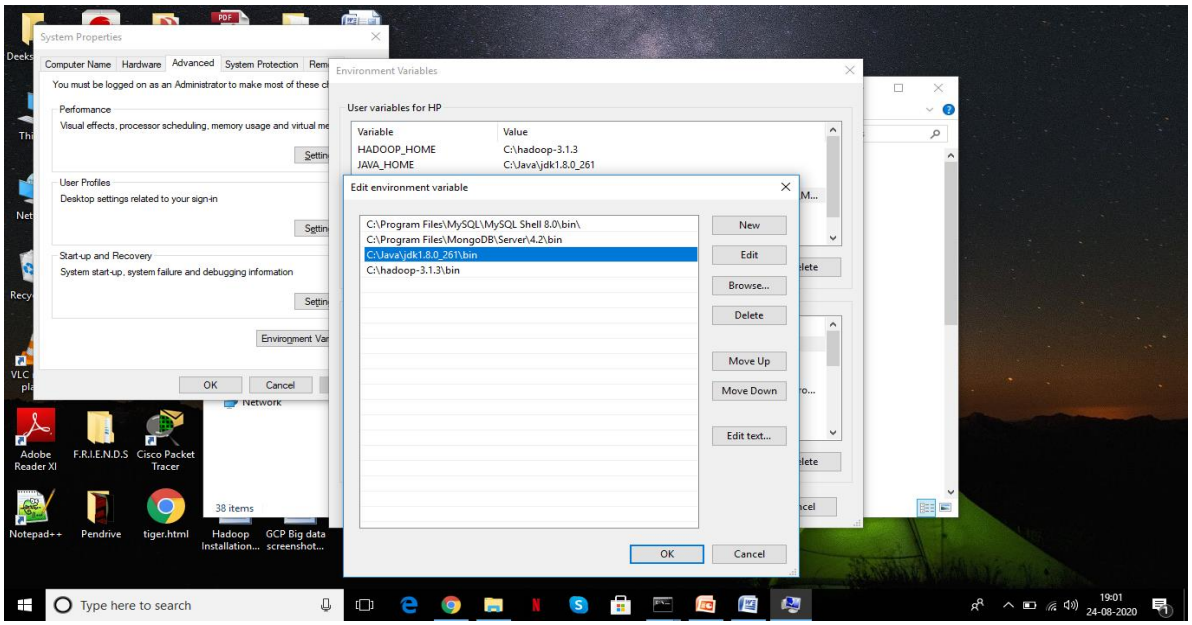


Fig2

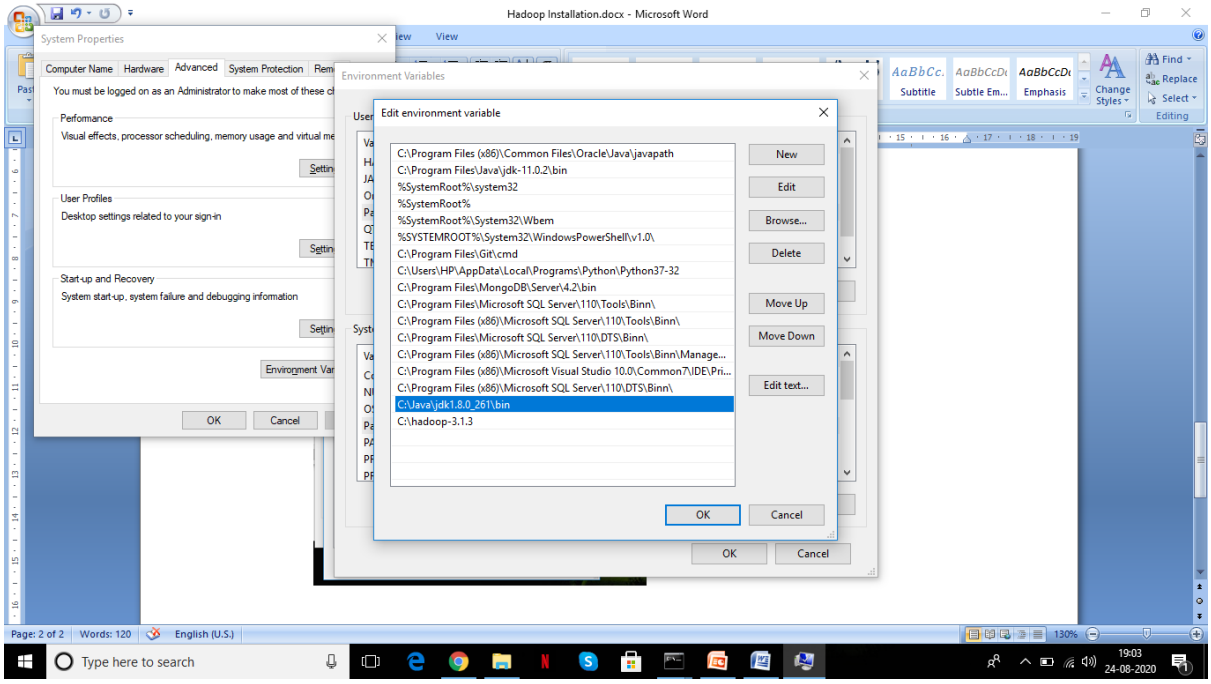


Fig3

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

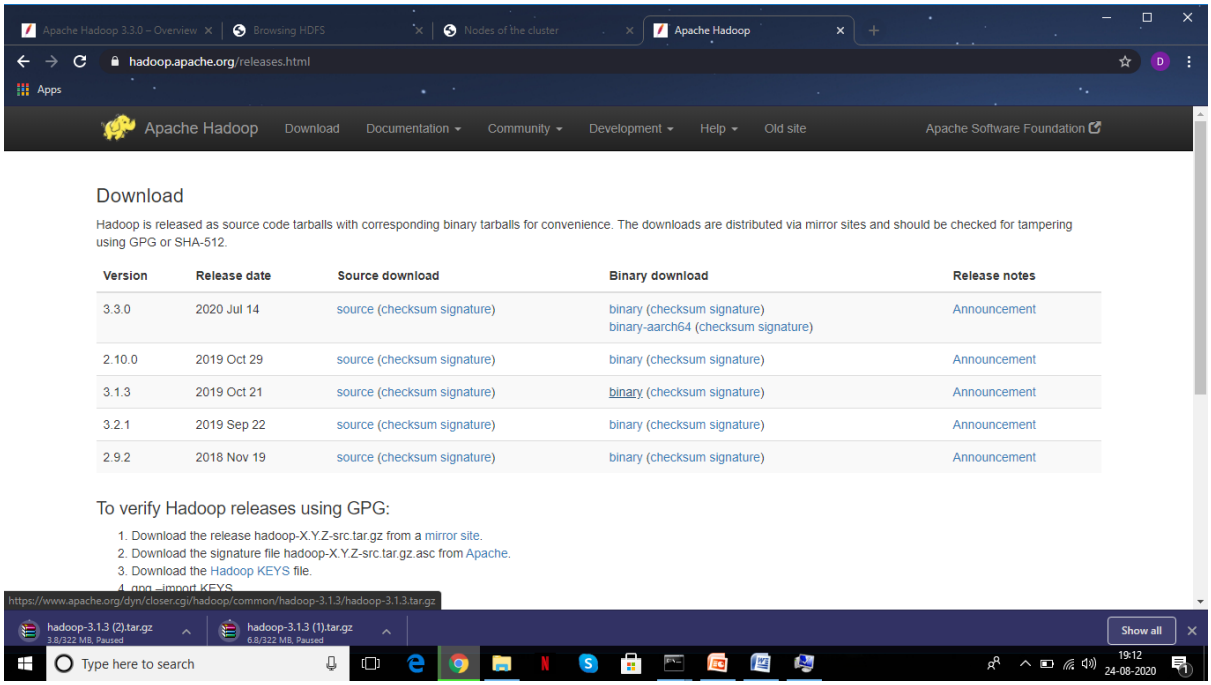

Apache Hadoop 3.3.0 – Overview
Browsing HDFS
Nodes of the cluster
Apache Hadoop
hadoop.apache.org/releases.html
Apps
Apache Hadoop
Download
Documentation
Community
Development
Help
Old site
Apache Software Foundation
Download
Hadoop is released as source code tarballs with corresponding binary tarballs 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.
https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.1.3/hadoop-3.1.3.tar.gz
hadoop-3.1.3 (2).tar.gz
hadoop-3.1.3 (1).tar.gz
Show all
Type here to search
19:12
24-08-2020

Fig4

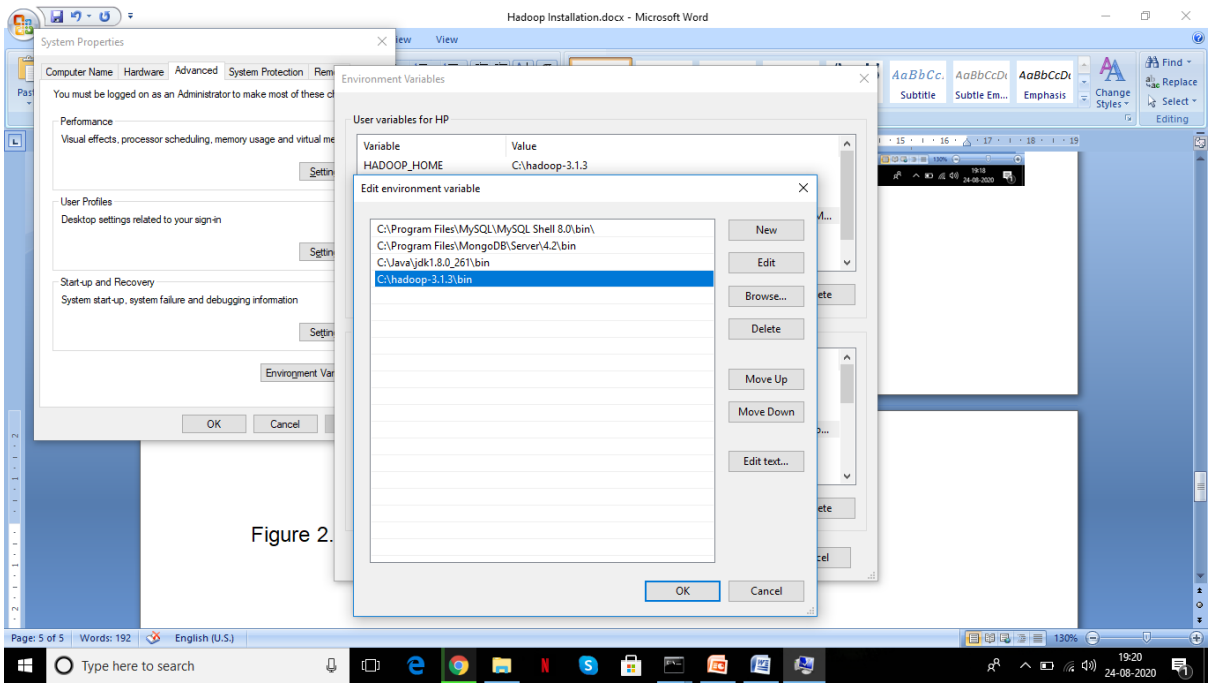

Hadoop Installation.docx - Microsoft Word
System Properties
Computer Name
Hardware
Advanced
System Protection
You must be logged on as an Administrator to make most of these ch
Performance
Visual effects, processor scheduling, memory usage and virtual me
User Profiles
Desktop settings related to your sign-in
Start-up and Recovery
System start-up, system failure and debugging information
Environment Variables
User variables for HP
Variable
Value
HADOOP_HOME
C:\hadoop-3.1.3
Edit environment variable
C:\Program Files\MySQL\MySQL Shell 8.0\bin\
C:\Program Files\MongoDB\Server\4.2\bin
C:\Java\jdk1.8.0_261\bin
C:\hadoop-3.1.3\bin
New
Edit
Browse...
Delete
Move Up
Move Down
Edit text...
OK
Cancel
Figure 2.
Page: 5 of 5
Words: 192
English (U.S.)
Type here to search
19:20
24-08-2020

Fig5

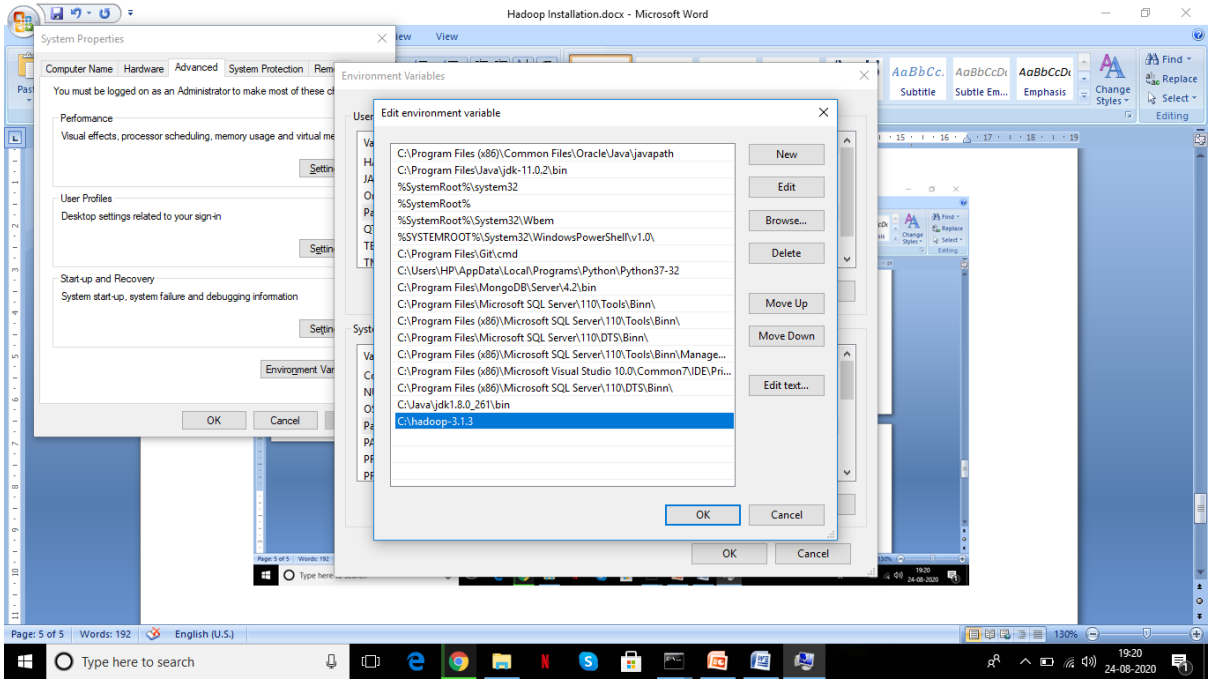


Fig6

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

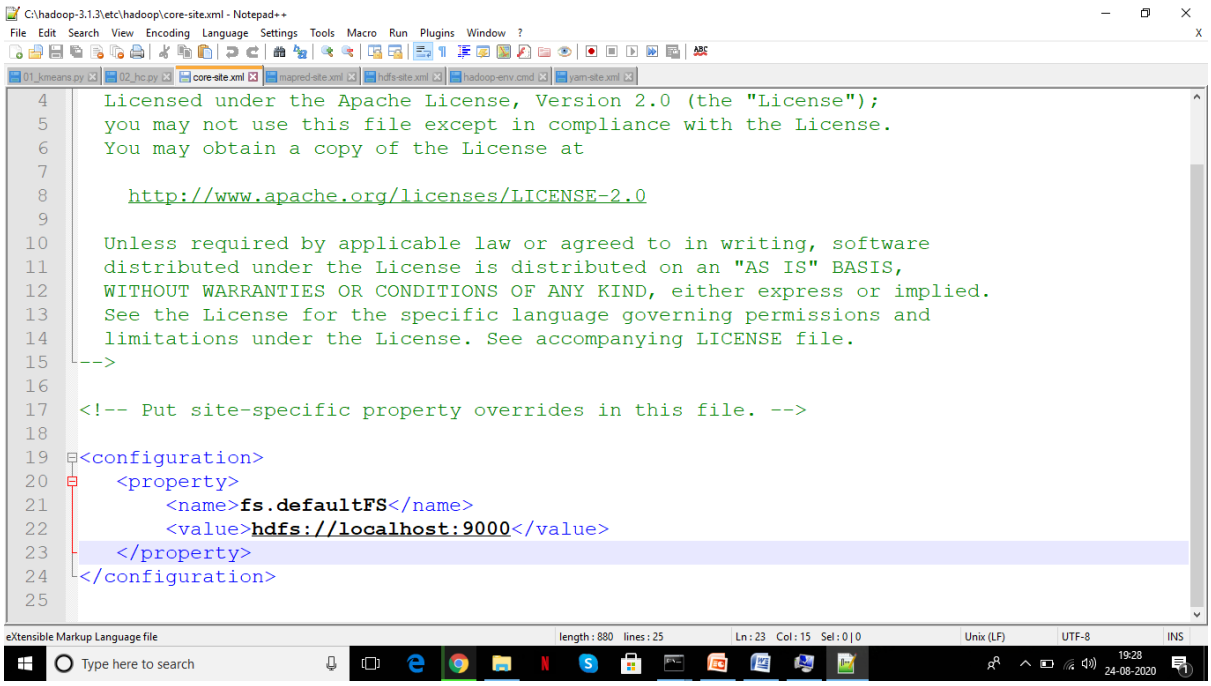


Fig7

- hadoop-env.cmd

```
@rem The only required environment variable is JAVA_HOME.  All others are
@rem optional.  When running a distributed configuration it is best to
@rem set JAVA_HOME in this file, so that it is correctly defined on
@rem remote nodes.

@rem The java implementation to use.  Required.
set JAVA_HOME=C:\Java\jdk1.8.0_261

@rem The jsvc implementation to use. Jsvc is required to run secure datanode
```

Fig8

- hdfs-site.xml

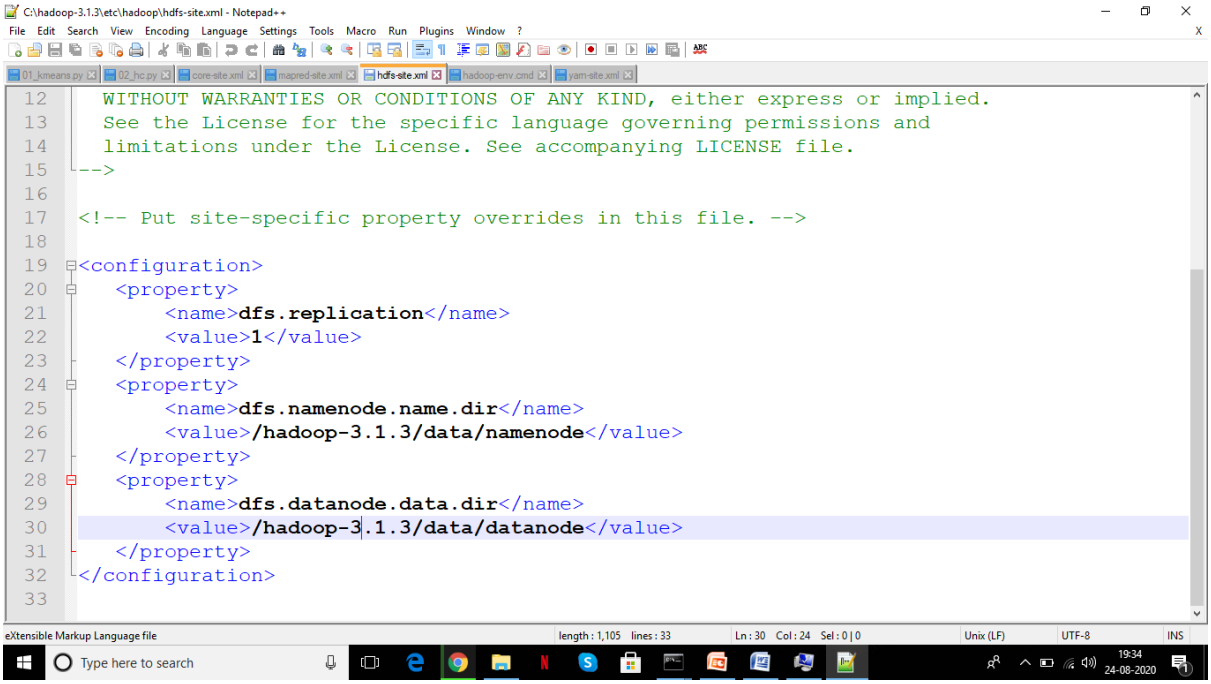


Fig9

- mapred-site.xml

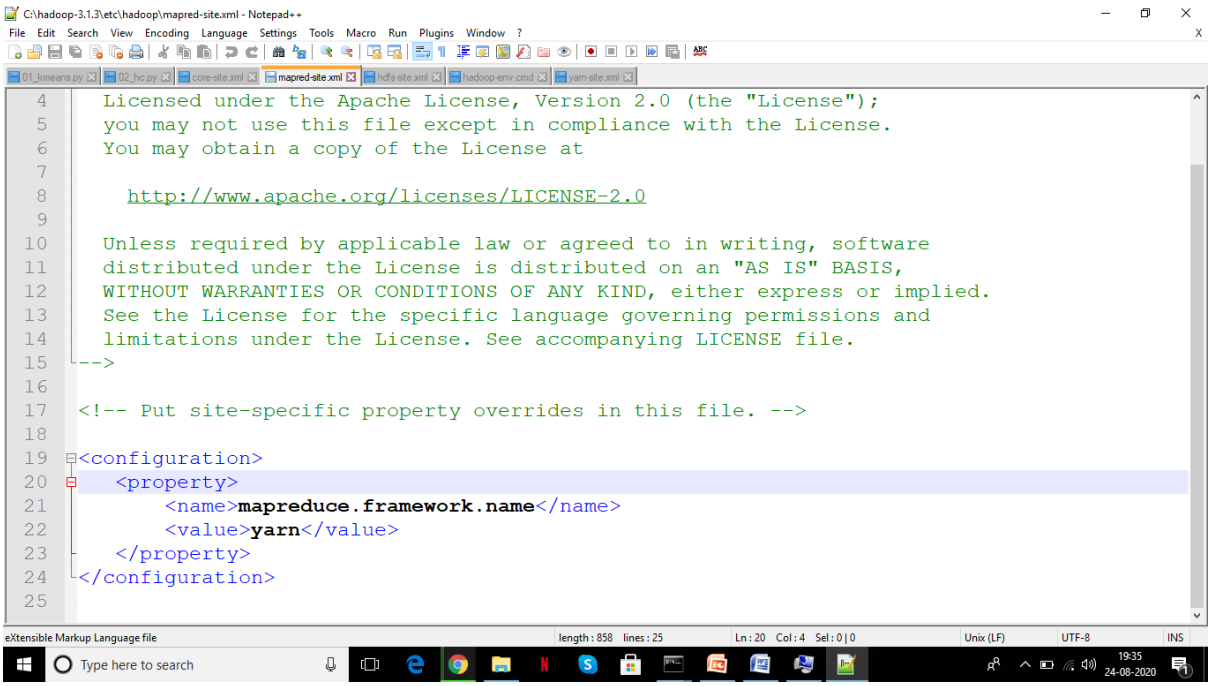


Fig10

- yarn-site.xml

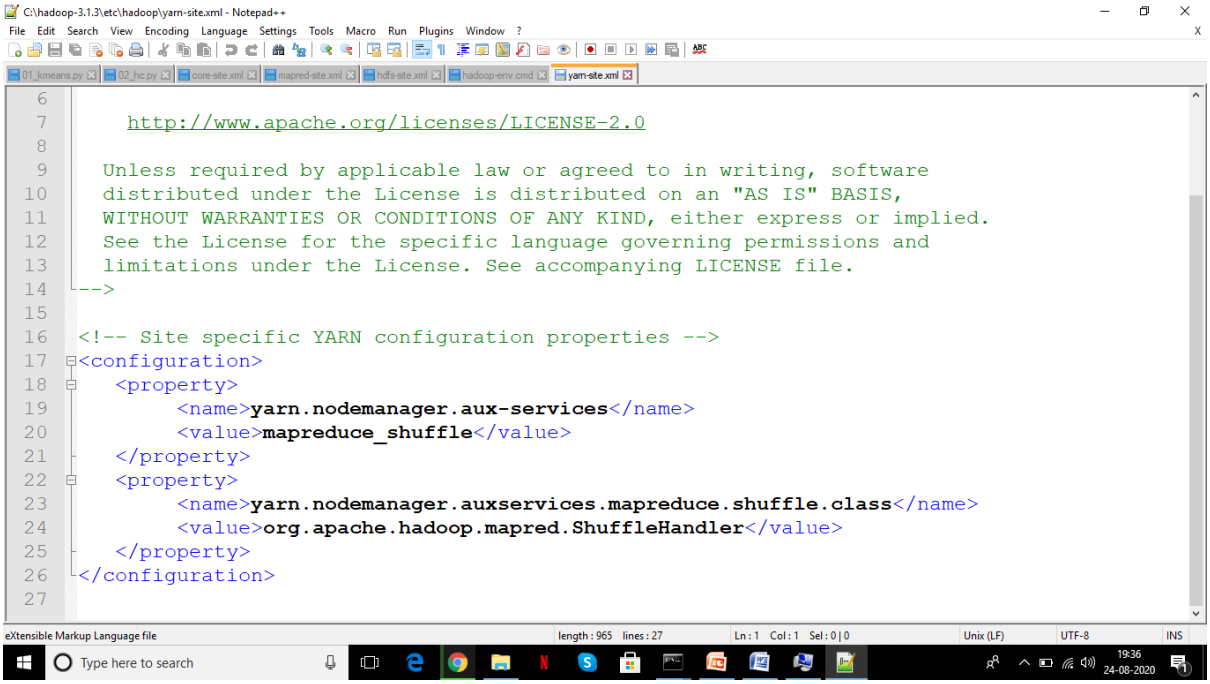


Fig11

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

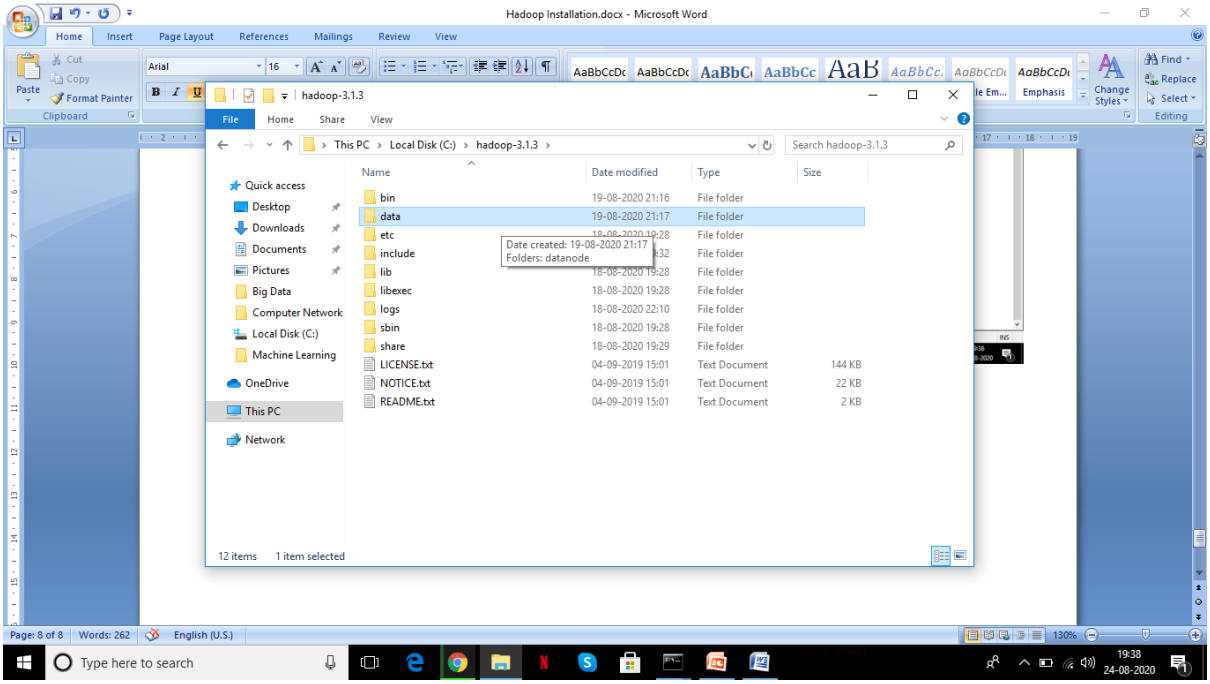


Fig12

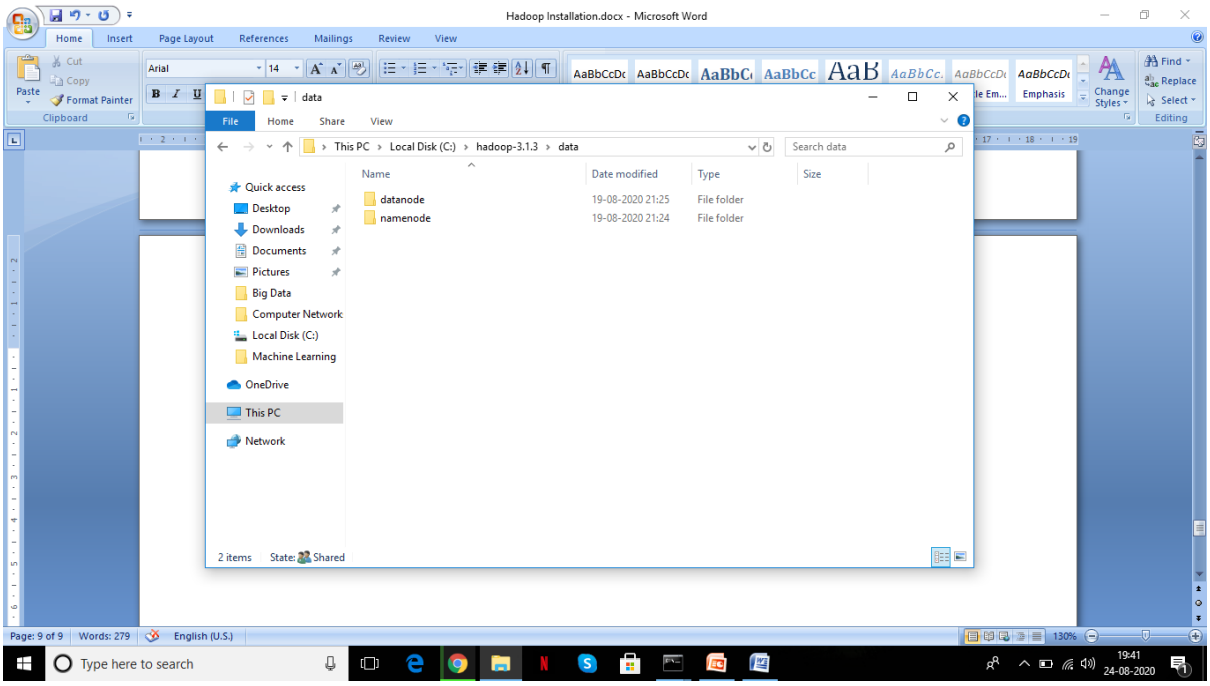


Fig13

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

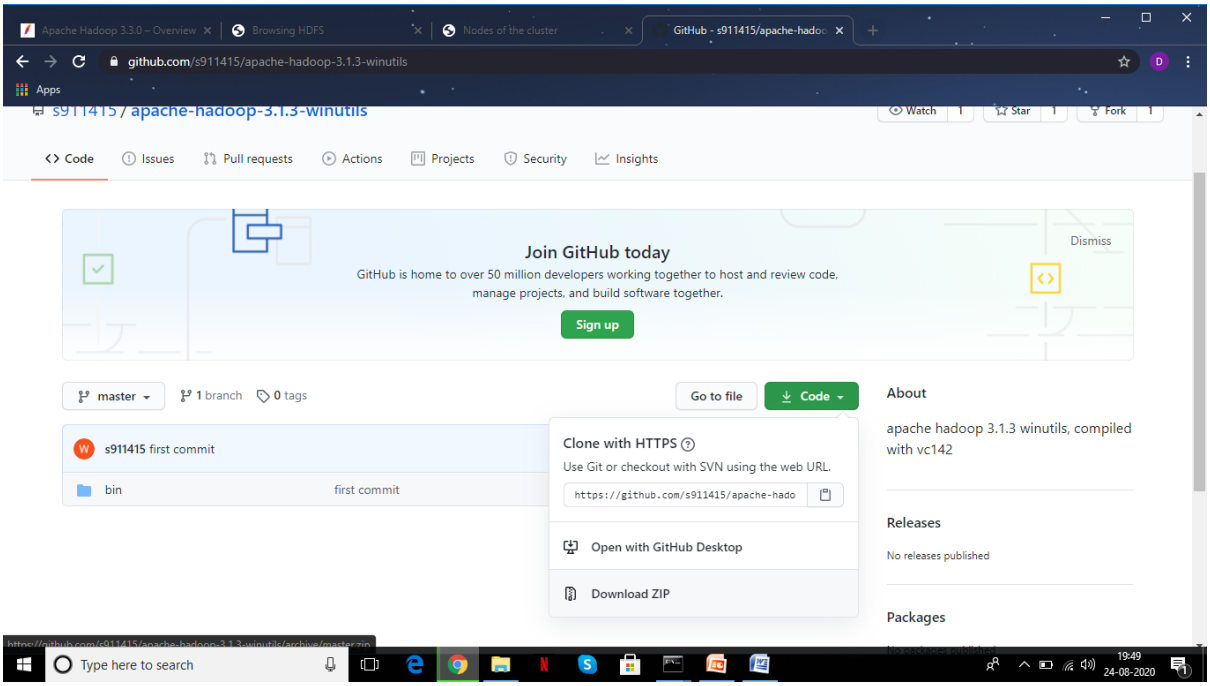


Fig14

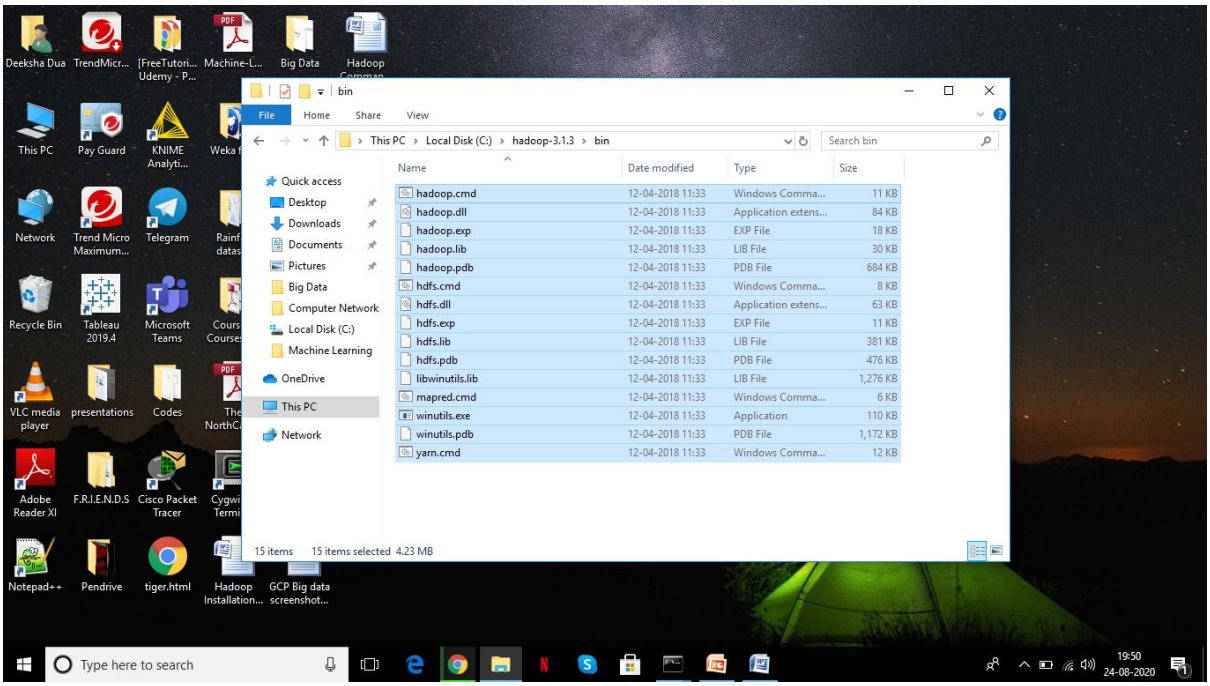


Fig15

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

- $hadoop version

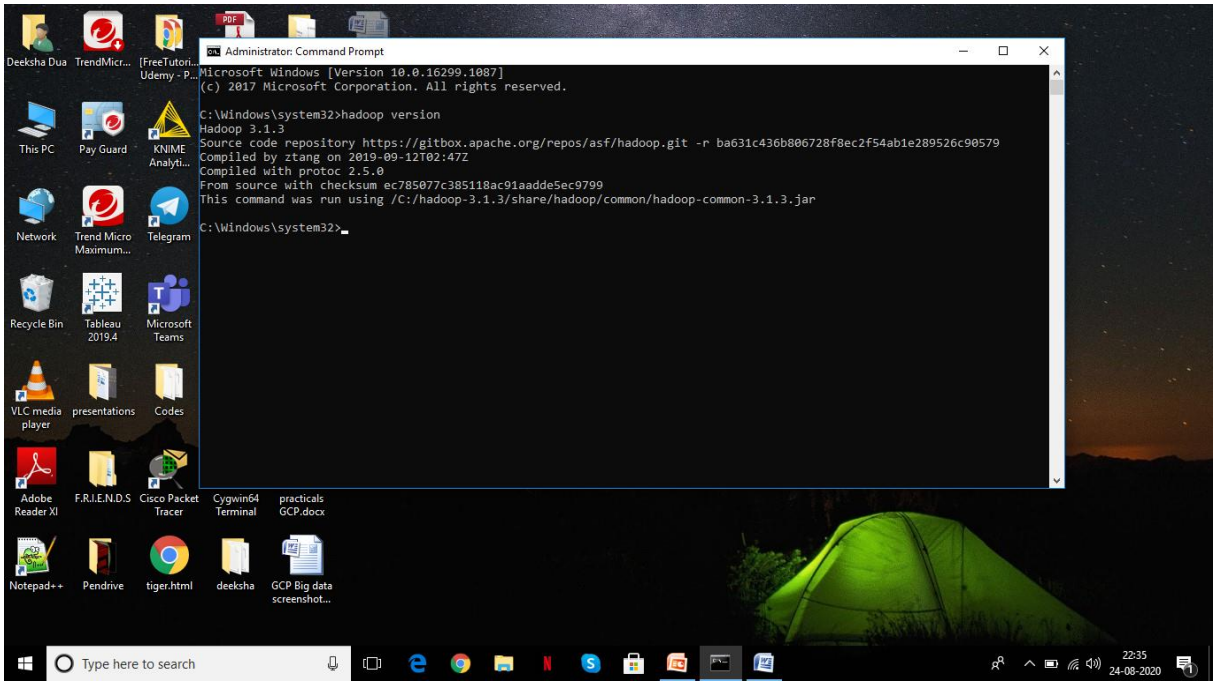


Fig16

Format the namenode

- $hdfs namenode -format

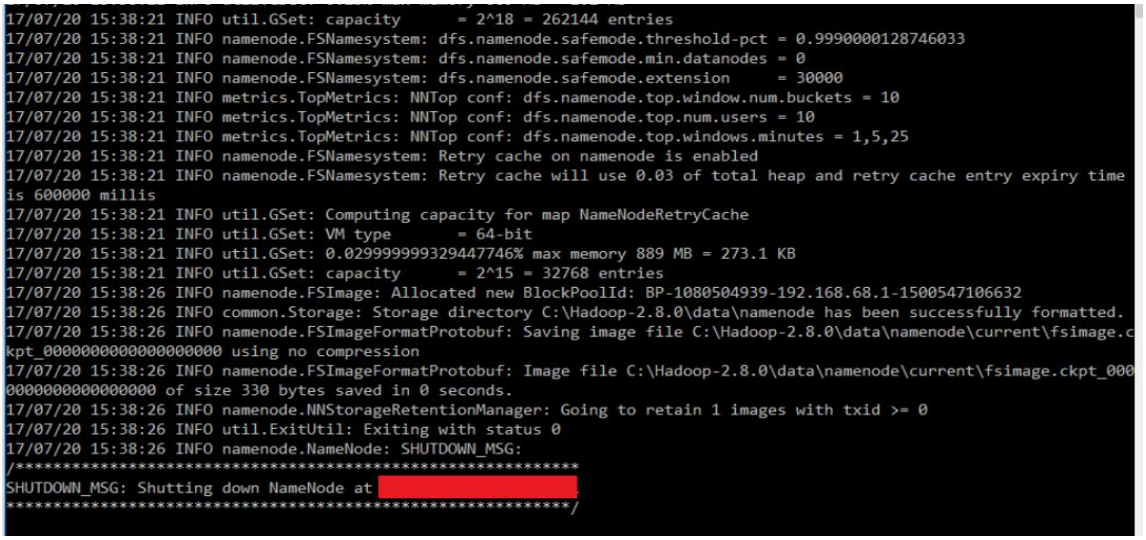


Fig17

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

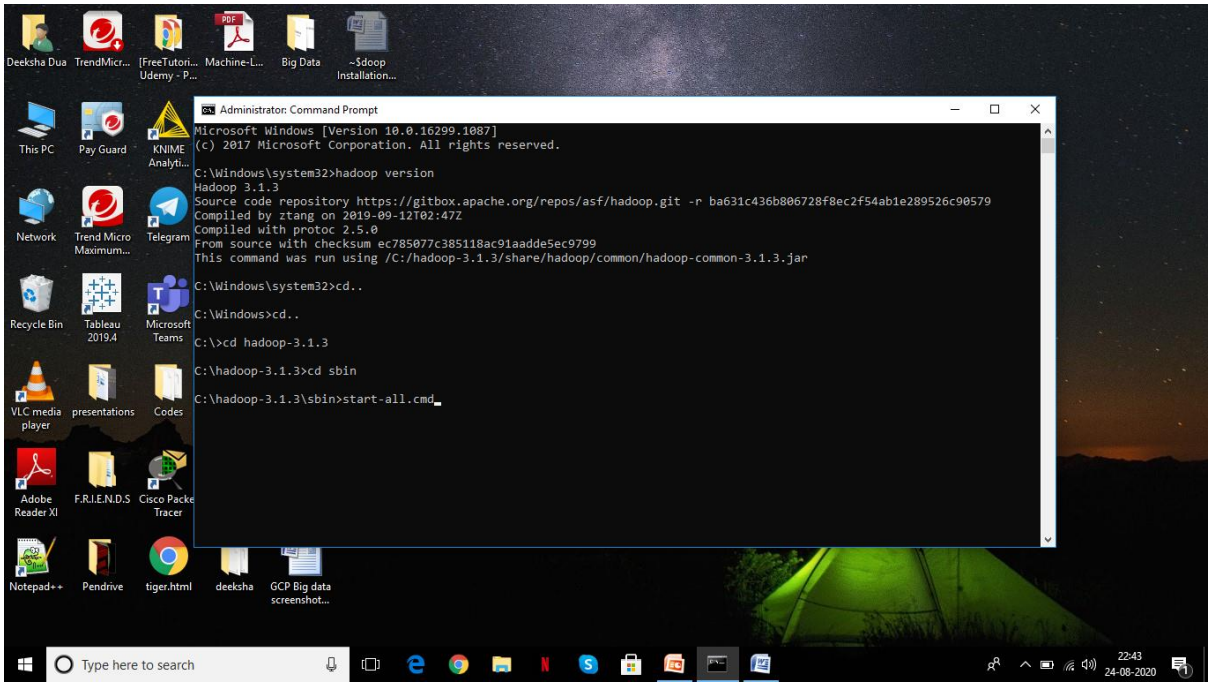


Fig18

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

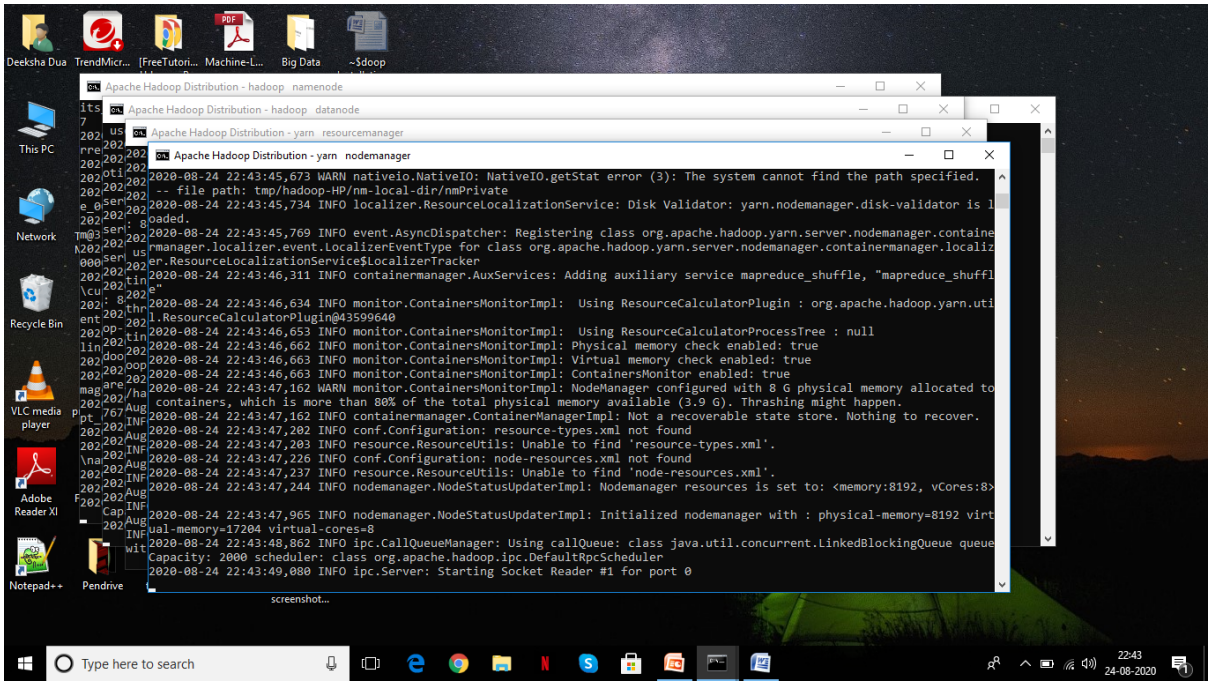


Fig19

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

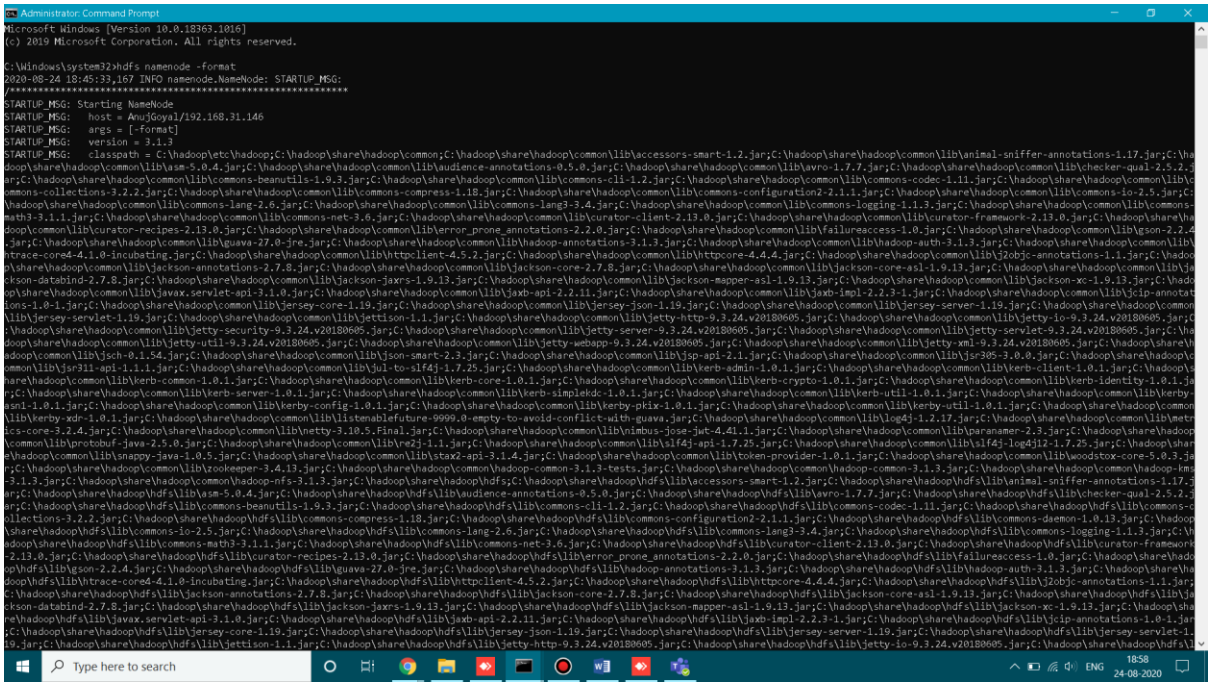


Fig1

- 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

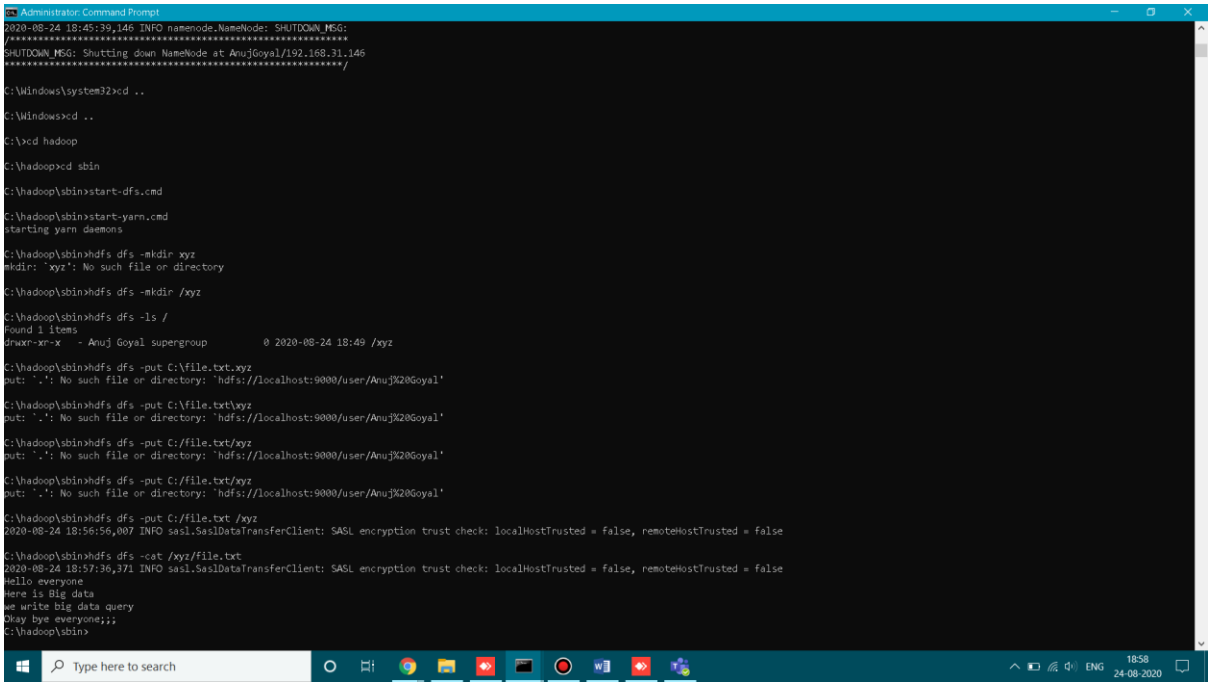


Fig2

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
- $hdfs dfs -chmod [-option] <MODE[,MODE]… | OCTALMODE> URI
- Options: -R : This option will make the change recursively.

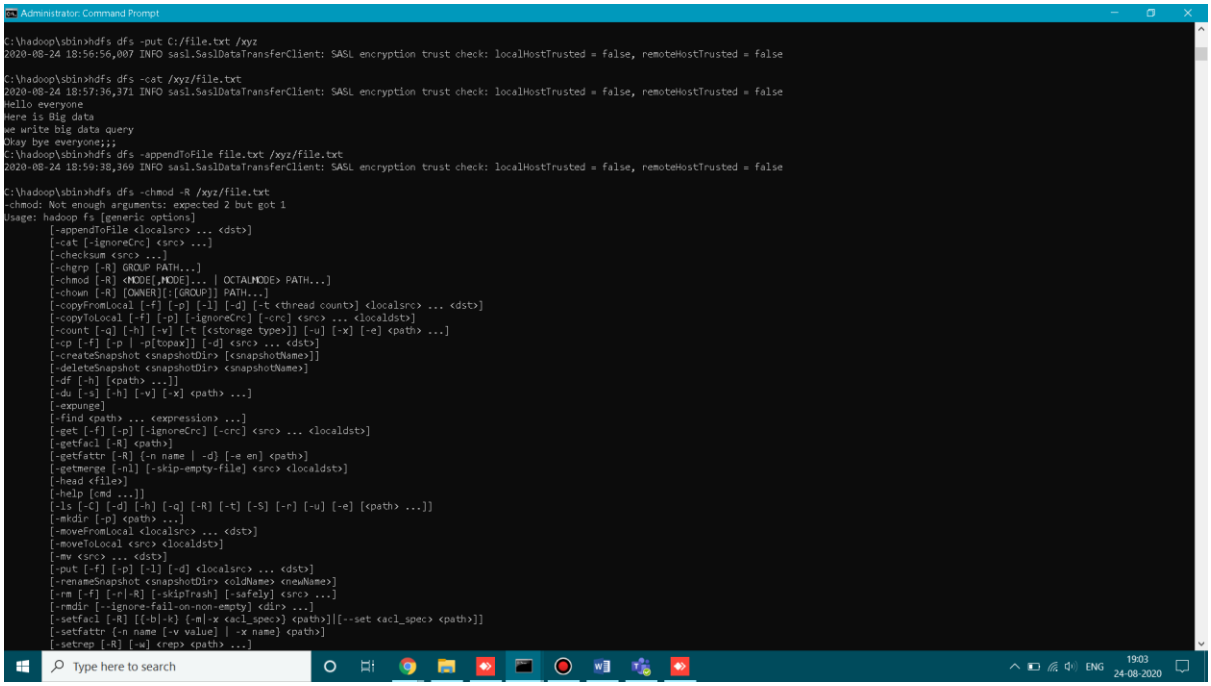


Fig3

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 /

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

- -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 /
```

-q -h -v

```
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

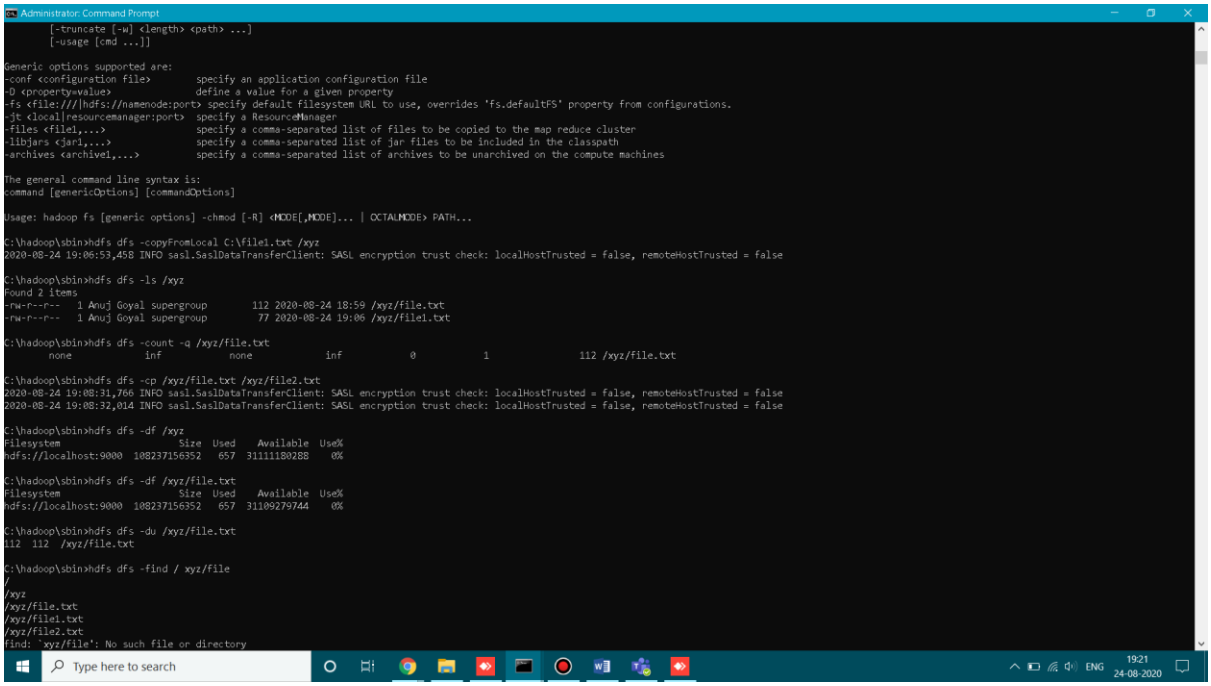
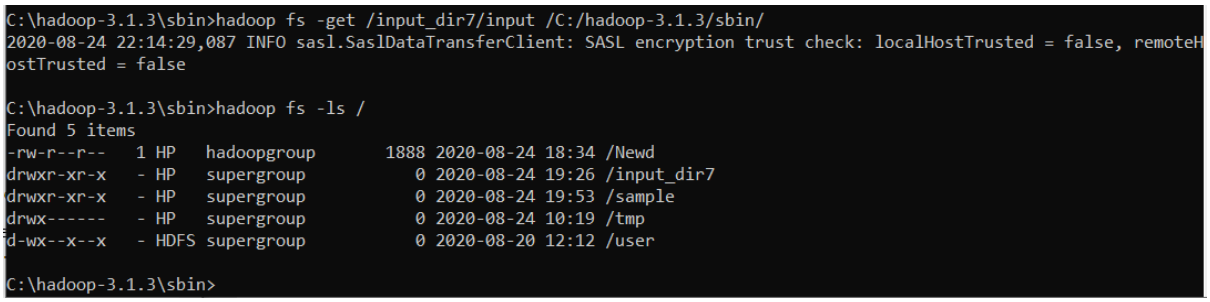


Fig4

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.



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

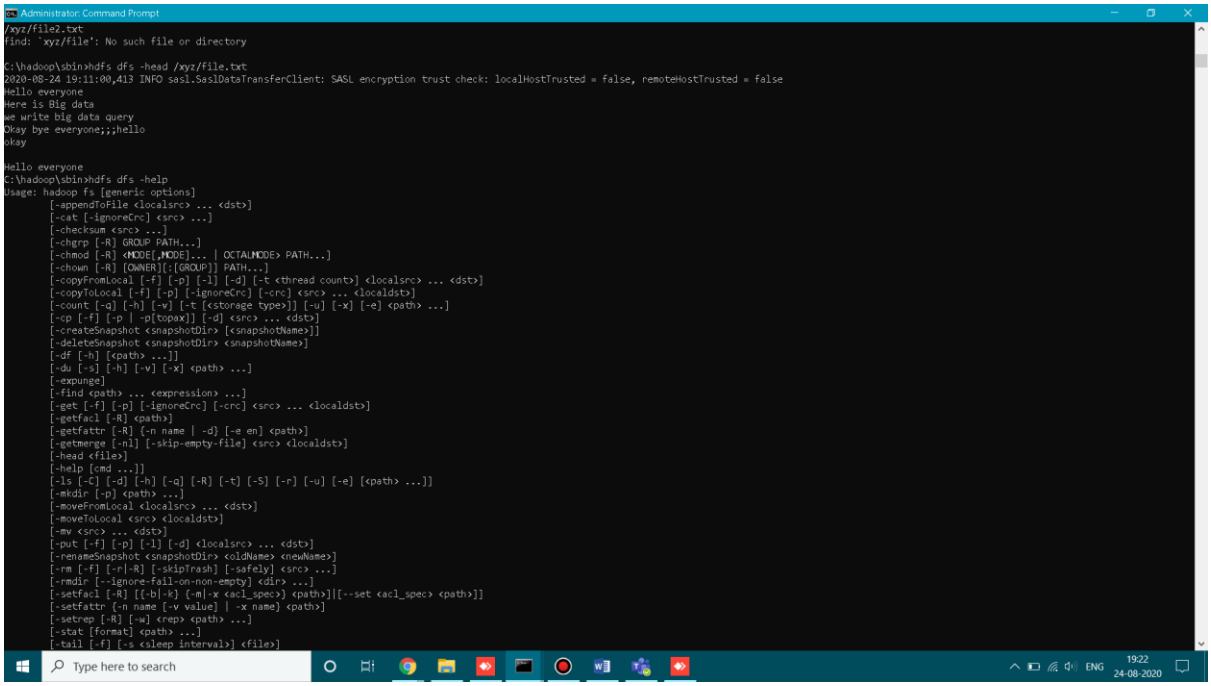


Fig5

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

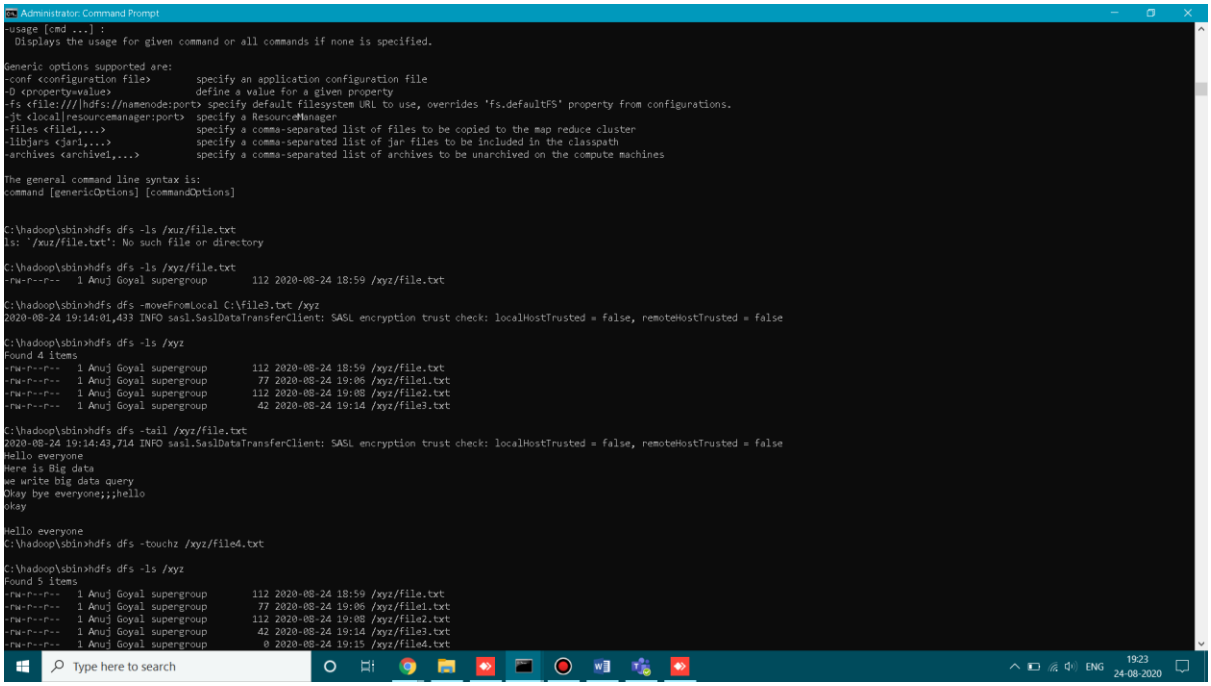


Fig6

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

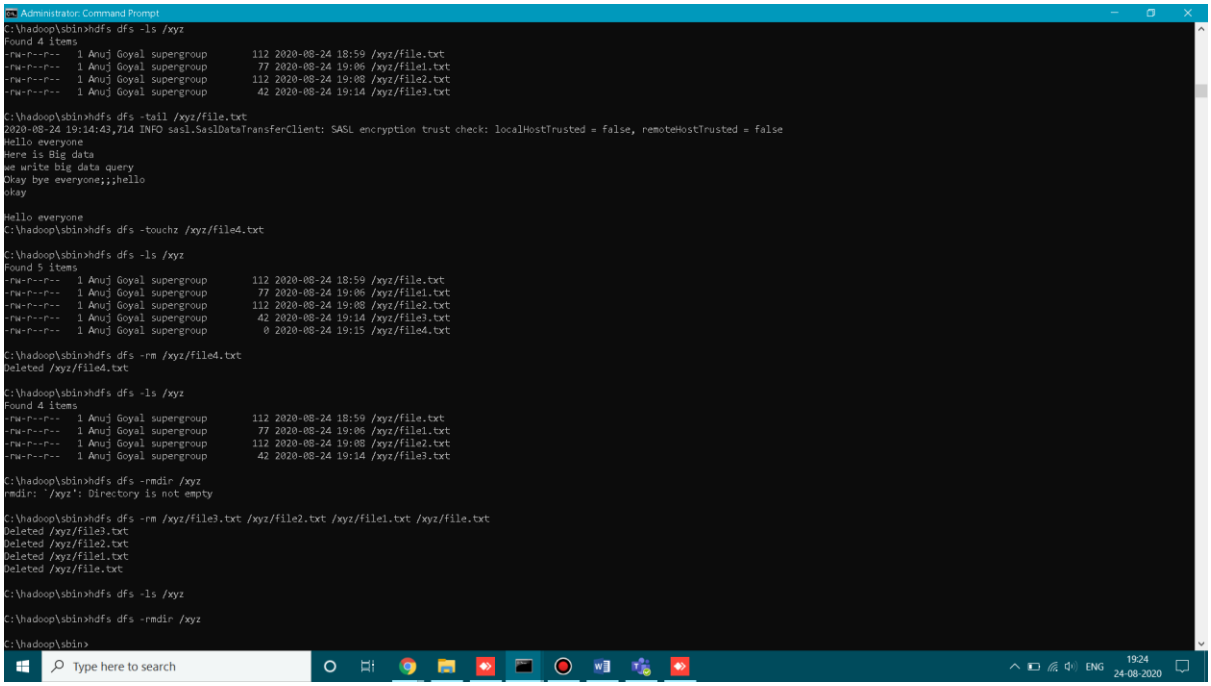

Administrator: Command Prompt
C:\hadoop\sbin>hdfs dfs -ls /xyz
Found 4 items
-rw-r--r--   1 Anuj Goyal supergroup   112 2020-08-24 18:59 /xyz/file.txt
-rw-r--r--   1 Anuj Goyal supergroup   77 2020-08-24 19:06 /xyz/file1.txt
-rw-r--r--   1 Anuj Goyal supergroup   112 2020-08-24 19:08 /xyz/file2.txt
-rw-r--r--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

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

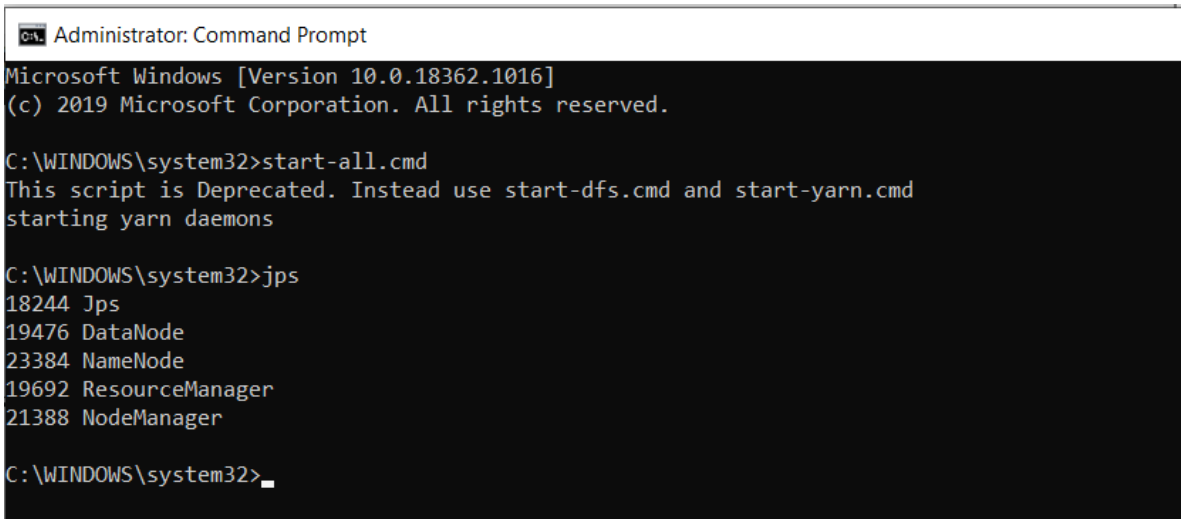


Fig1

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 sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
23  23  27 43  24  25  26  26  26  26  25  26 25
26  27  28 28  28  30  31  31  31  30  30  30 29
31  32  32 32  33  34  35  36  36  34  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  26  25  26 25
26  27  28 28  28  30  31  31  31  30  30  30 29
31  32  32 32  33  34  35  36  36  34  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  26  25  26 25
26  27  28 28  28  30  31  31  31  30  30  30 29
31  32  32 32  33  34  35  36  36  34  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  26  25  26 25
26  27  28 28  28  30  31  31  31  30  30  30 29
31  32  32 32  33  34  35  36  36  34  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  26  25  26 25
26  27  28 28  28  30  31  31  31  30  30  30 29
31  32  32 32  33  34  35  36  36  34  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  26  25  26 25
26  27  28 28  28  30  31  31  31  30  30  30 29
31  32  32 32  33  34  35  36  36  34  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

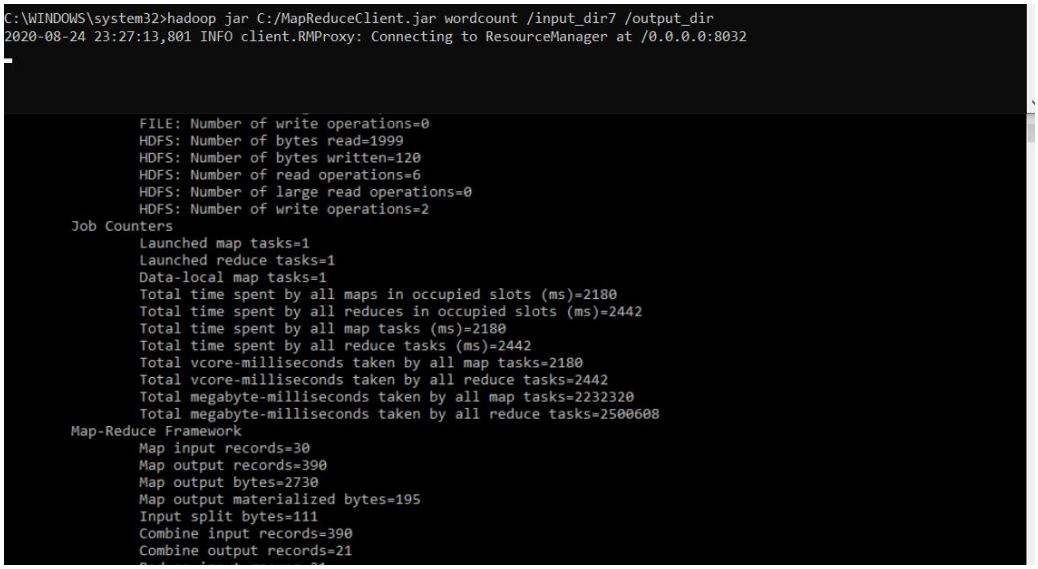


Fig4

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

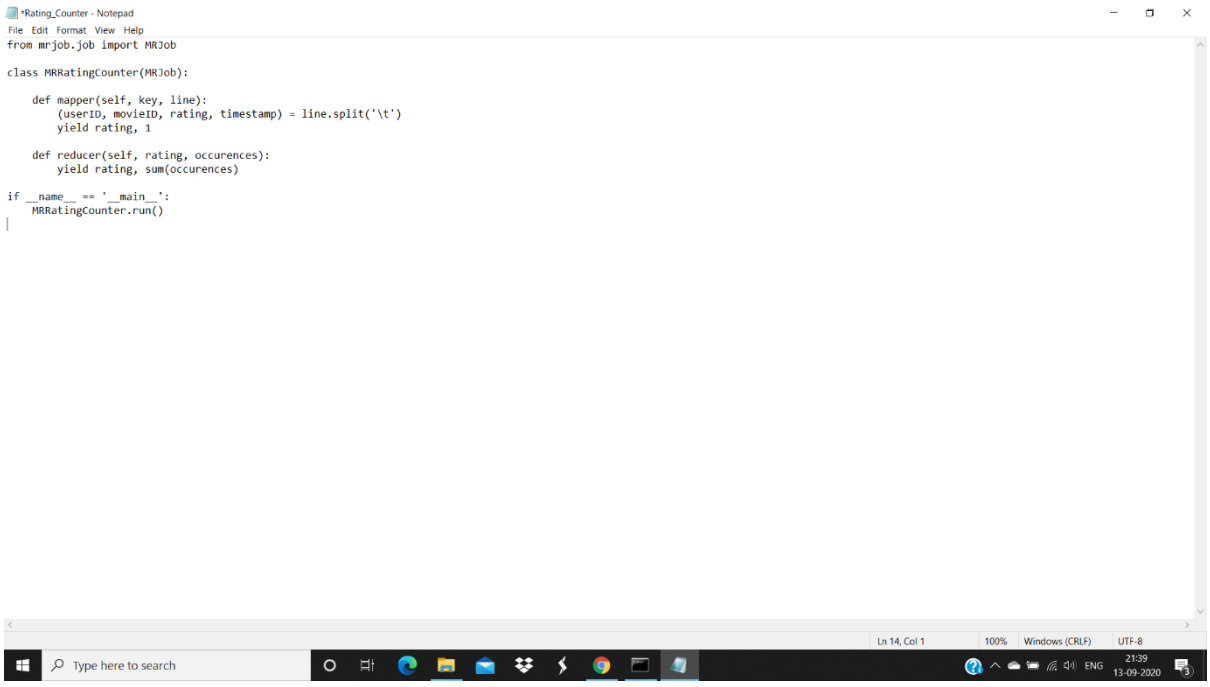


Fig1

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

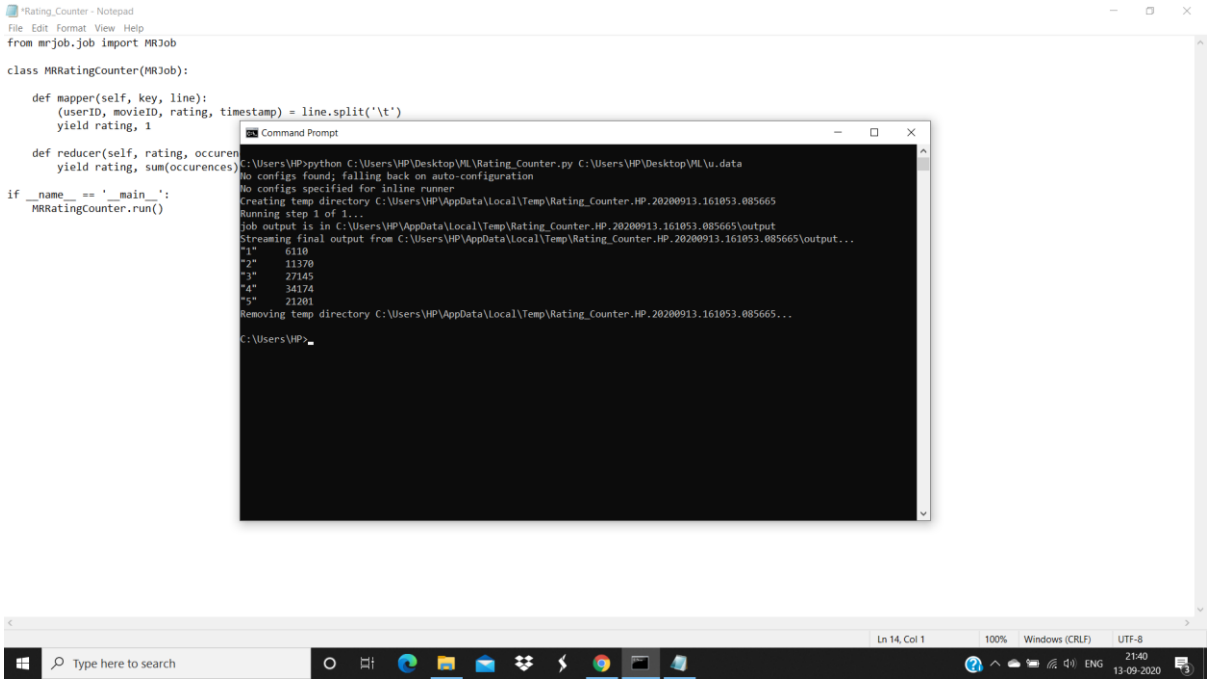


Fig2

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

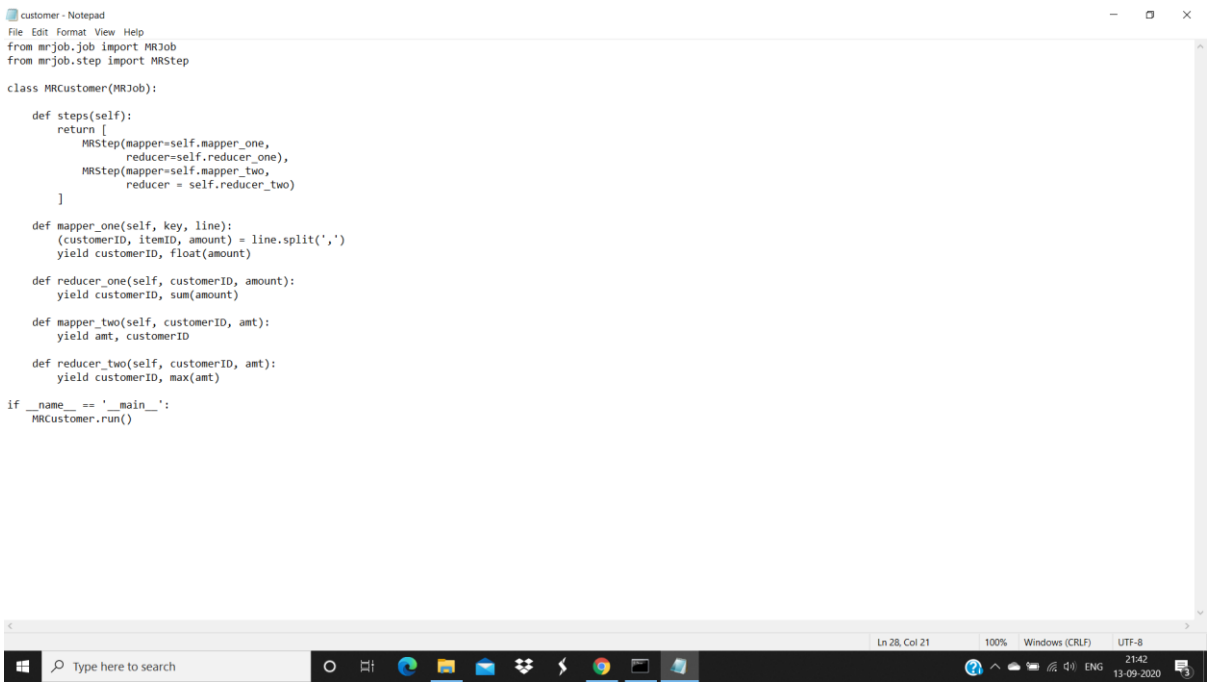


Fig3

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

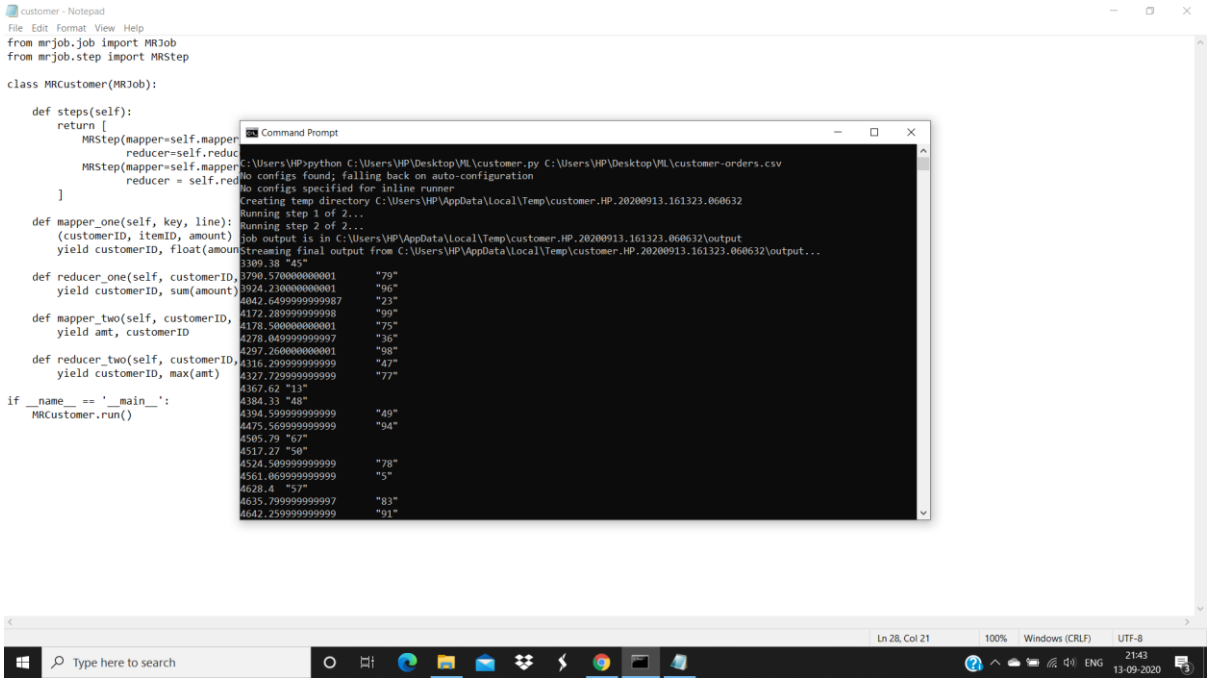

customer - Notepad
File Edit Format View Help
from mrjob.job import MRJob
from mrjob.step import MRStep
class MRCustomer(MRJob):
def steps(self):
return [
MRStep(mapper=self.mapper
reducer=self.reduc
MRStep(mapper=self.mapper
reducer = self.red
]
def mapper_one(self, key, line):
(customerID, itemID, amount)
yield customerID, float(amoun
def reducer_one(self, customerID,
yield customerID, sum(amount)
def mapper_two(self, customerID,
yield amt, customerID
def reducer_two(self, customerID,
yield customerID, max(amt)
if __name__ == '__main__':
MRCustomer.run()
Command Prompt
C:\Users\HP>python C:\Users\HP\Desktop\ML\customer.py C:\Users\HP\Desktop\ML\customer-orders.csv
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory C:\Users\HP\AppData\Local\Temp\customer.HP.20200913.161323.060632
Running step 1 of 2...
Running step 2 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...
3309.38 "45"
3790.570000000001 "79"
3924.230000000001 "96"
4042.6499999999987 "23"
4172.289999999998 "99"
4178.500000000001 "75"
4278.049999999997 "36"
4297.260000000001 "98"
4316.299999999999 "47"
4327.729999999999 "77"
4367.62 "13"
4384.33 "48"
4394.599999999999 "49"
4475.569999999999 "94"
4505.79 "67"
4517.27 "50"
4524.509999999999 "78"
4561.069999999999 "5"
4628.4 "57"
4635.799999999997 "83"
4642.259999999999 "91"
Ln 28, Col 21
100%
Windows (CRLF)
UTF-8
Type here to search
ENG
21:43
13-09-2020

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

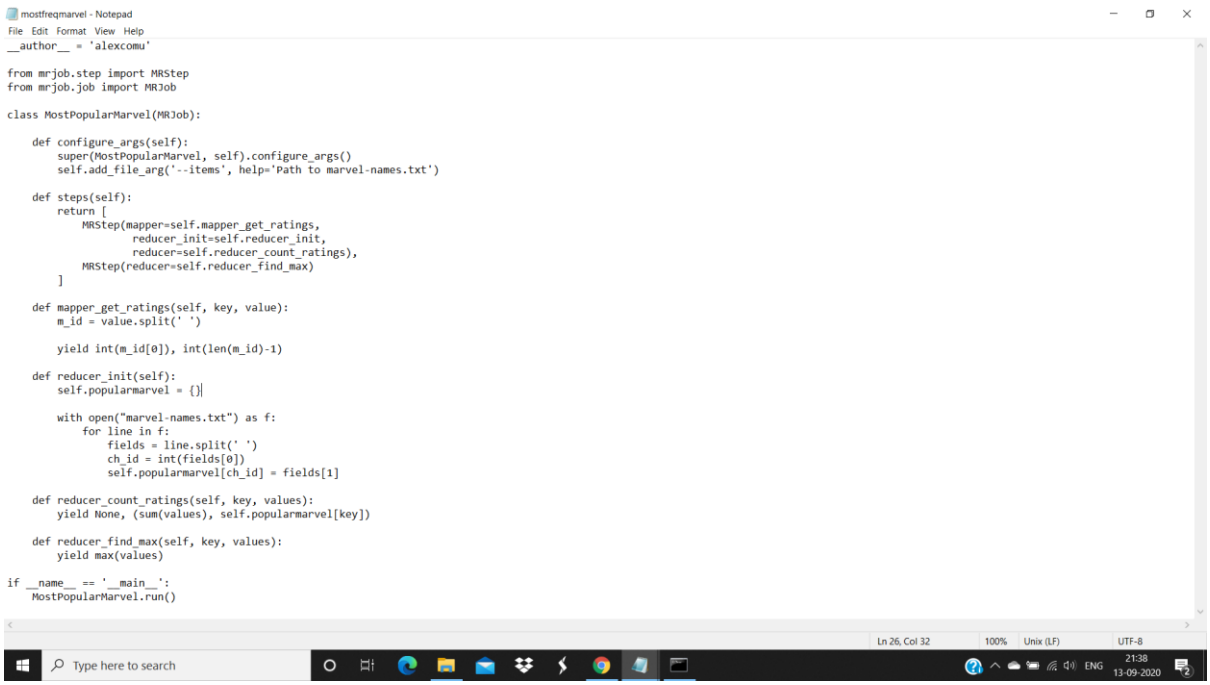


Fig5

- To run MRJob run the following commands:-

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

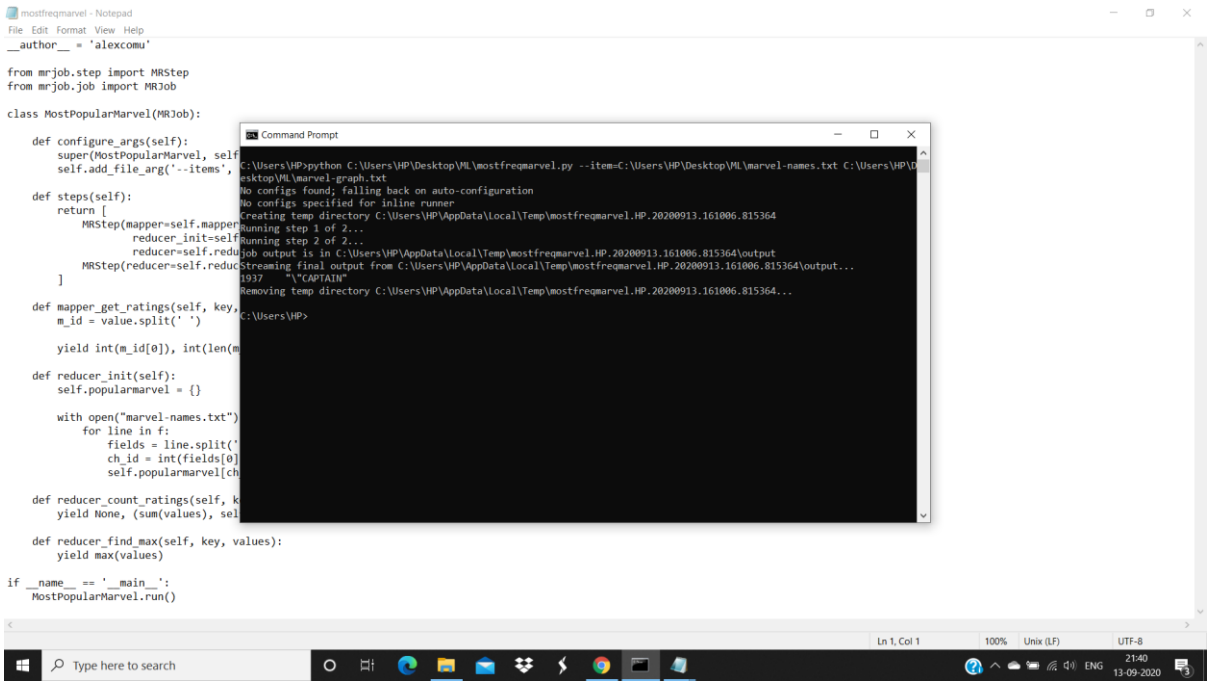


Fig6

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

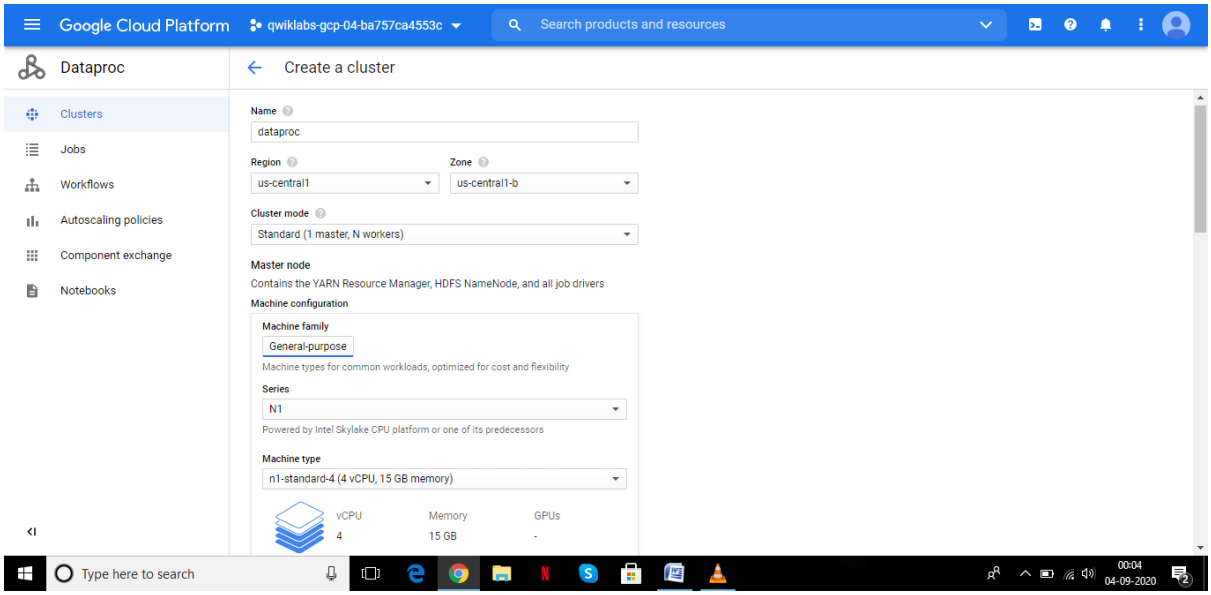


Fig1

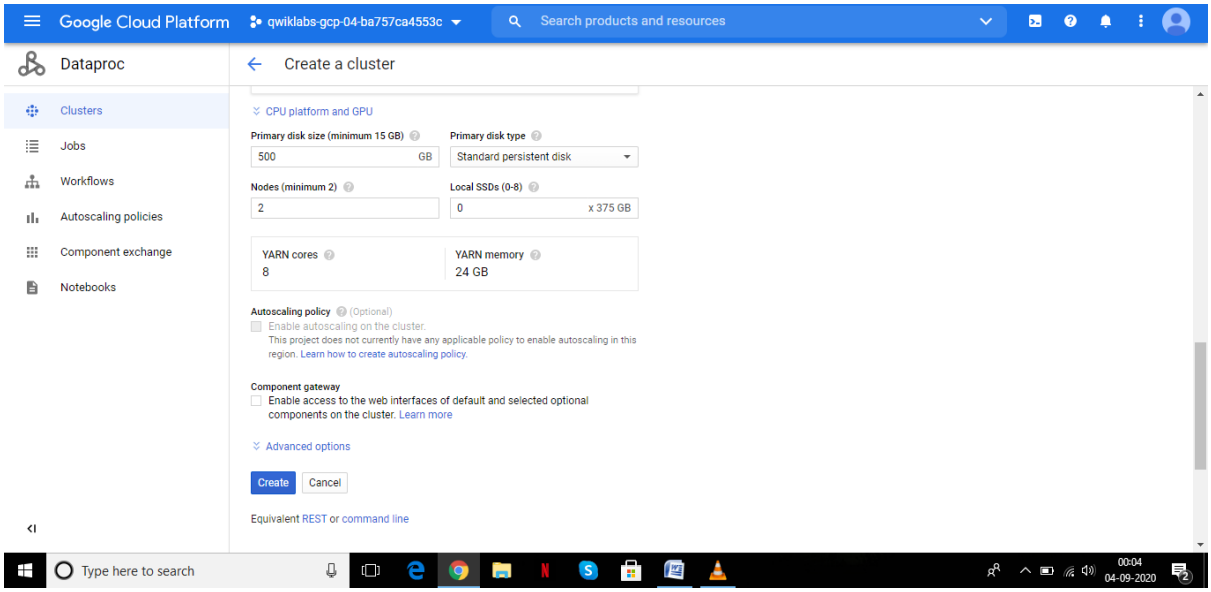


Fig2

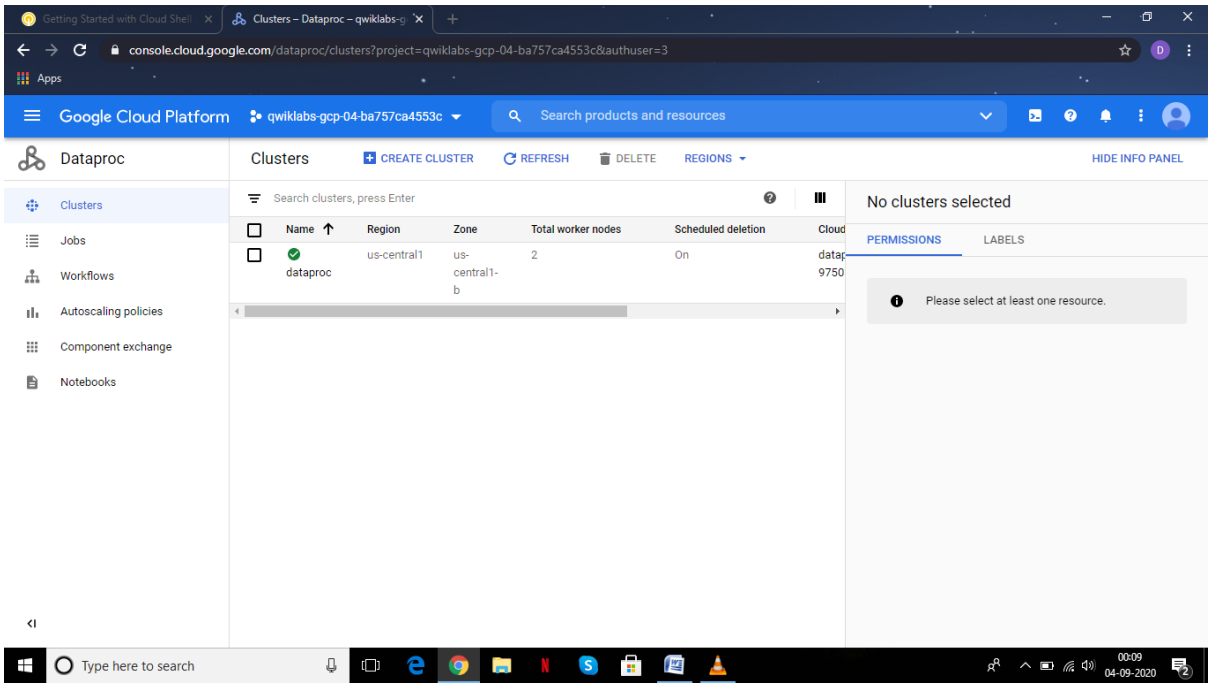


Fig3

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

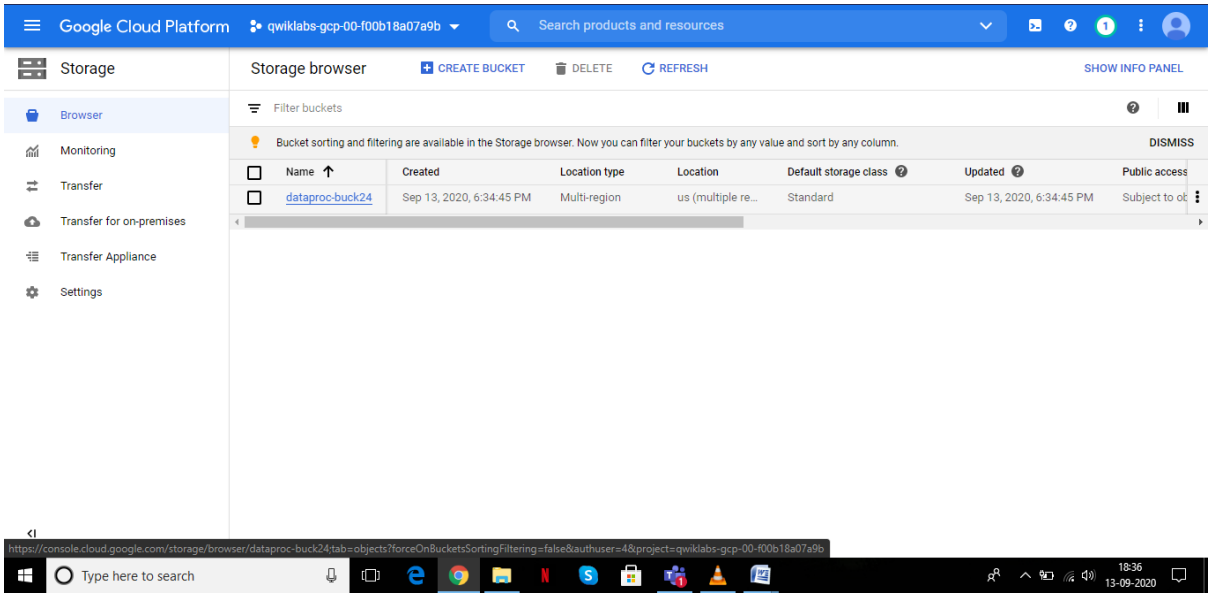


Fig4

- 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.

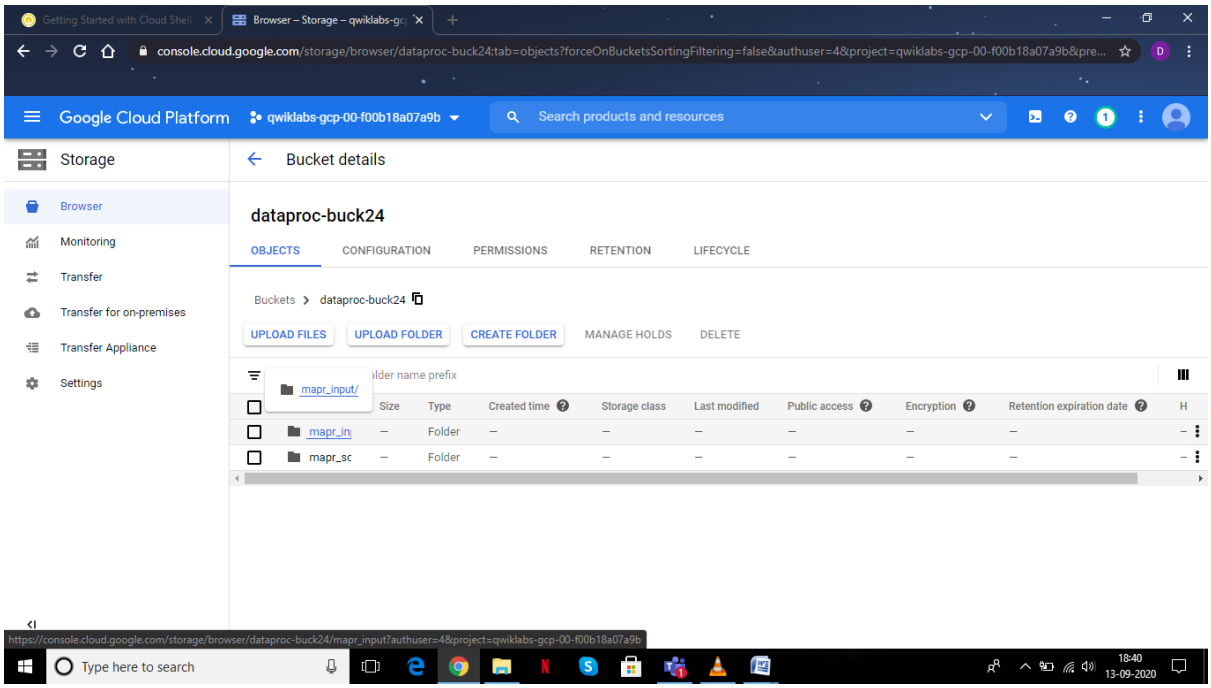


Fig5

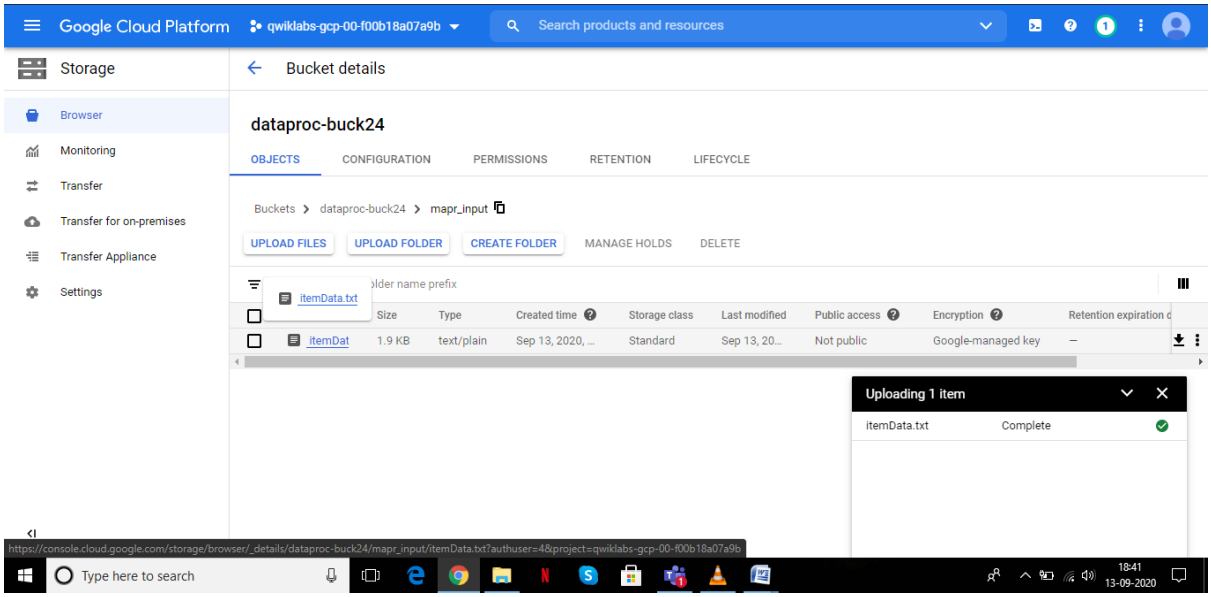


Fig6

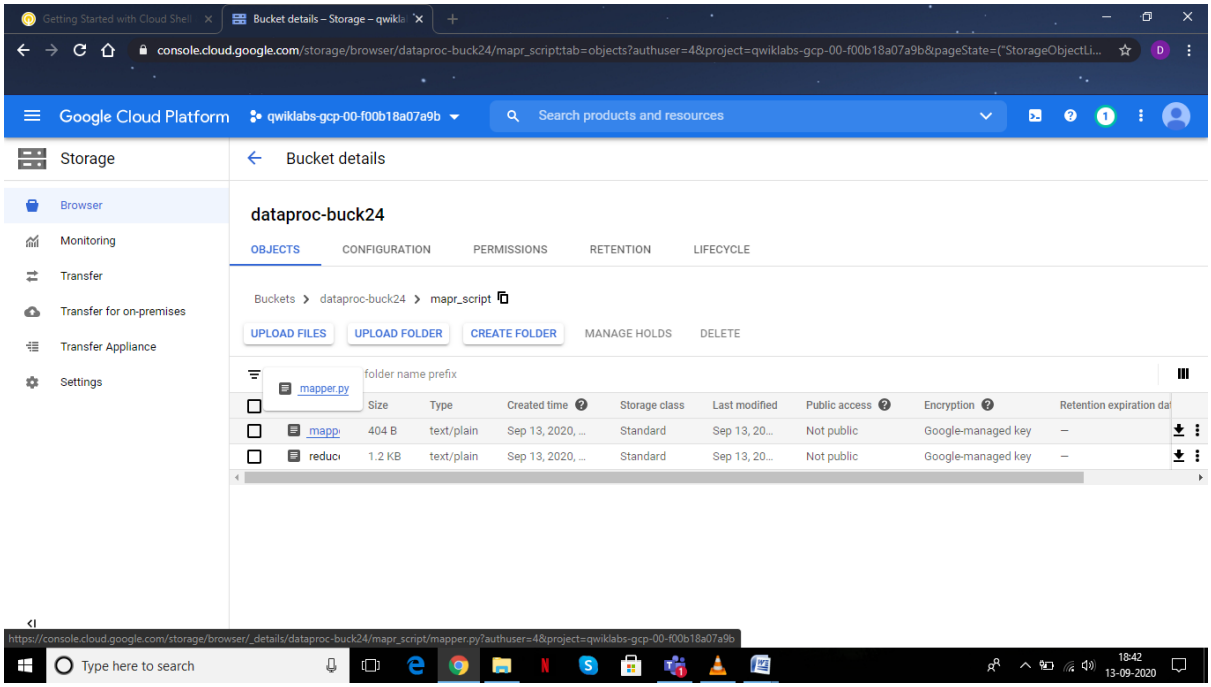


Fig7

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)

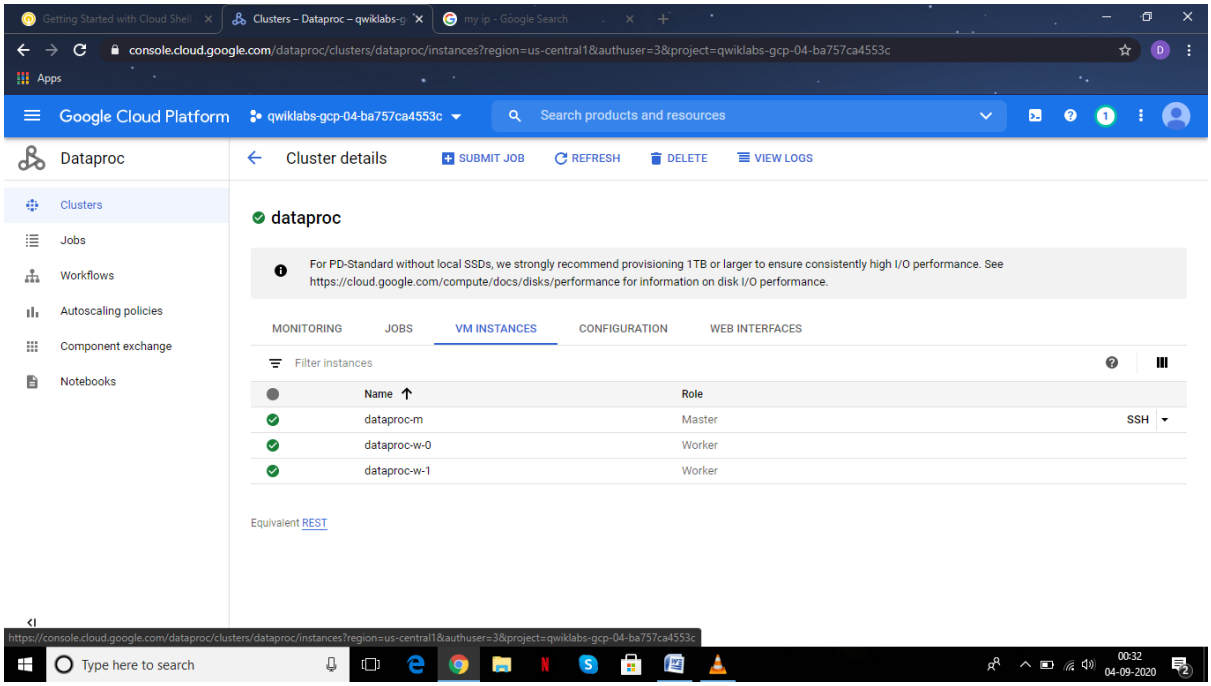


Fig8

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

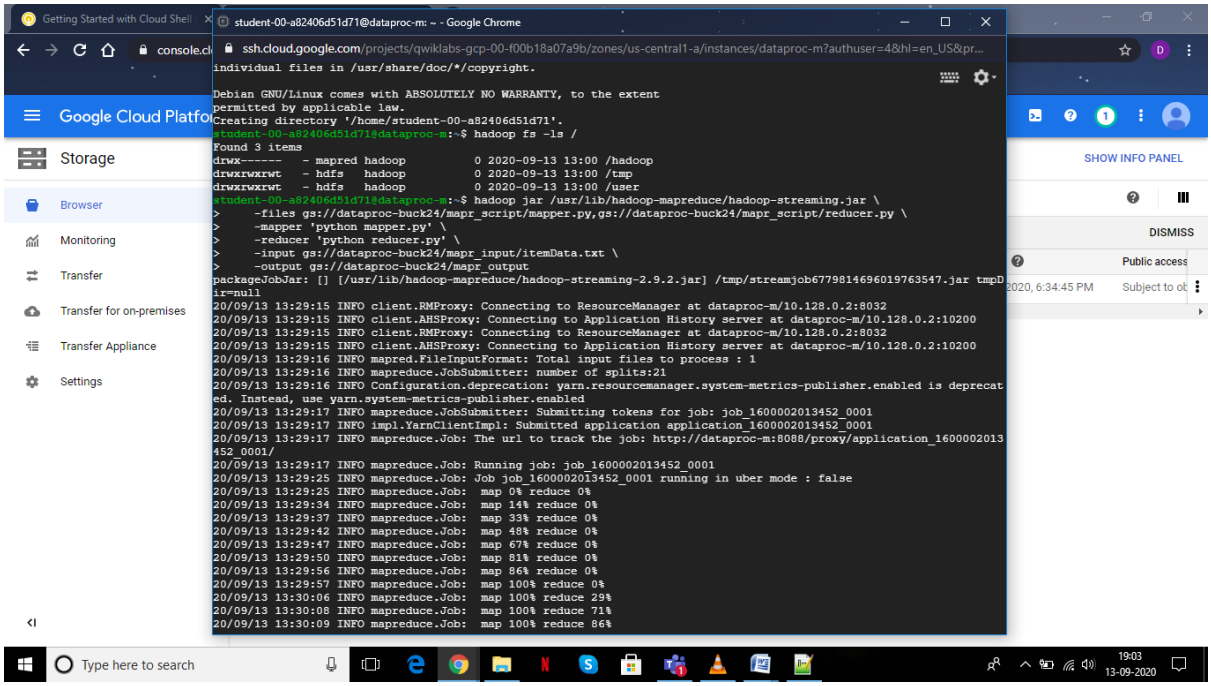


Fig9

- 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

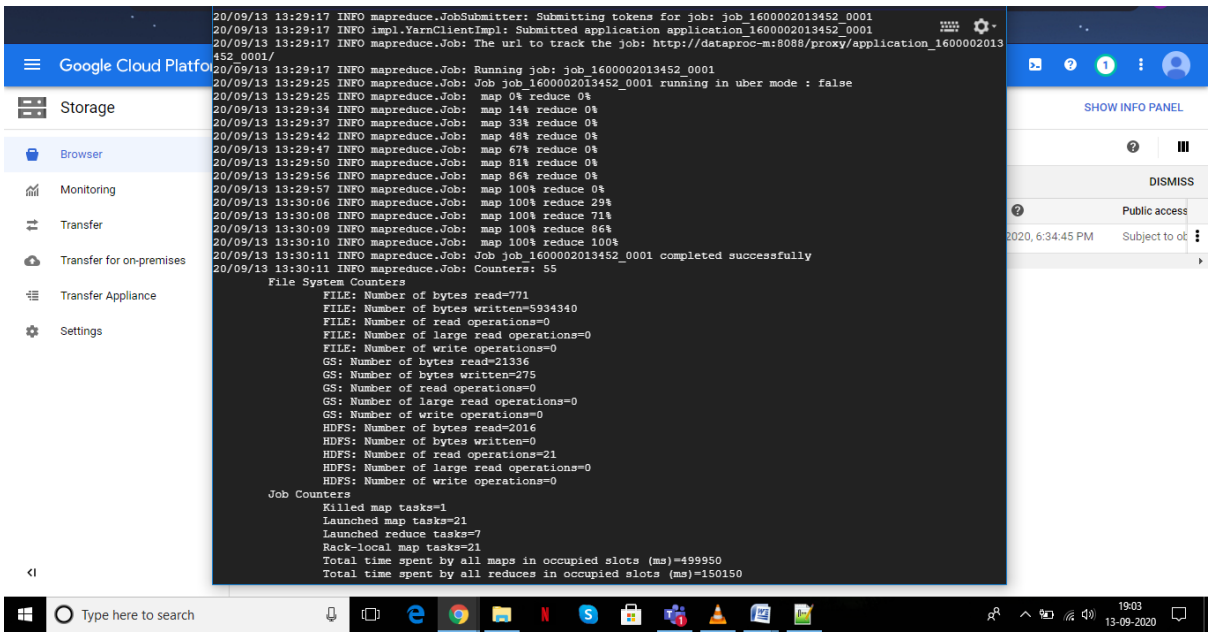


Fig10

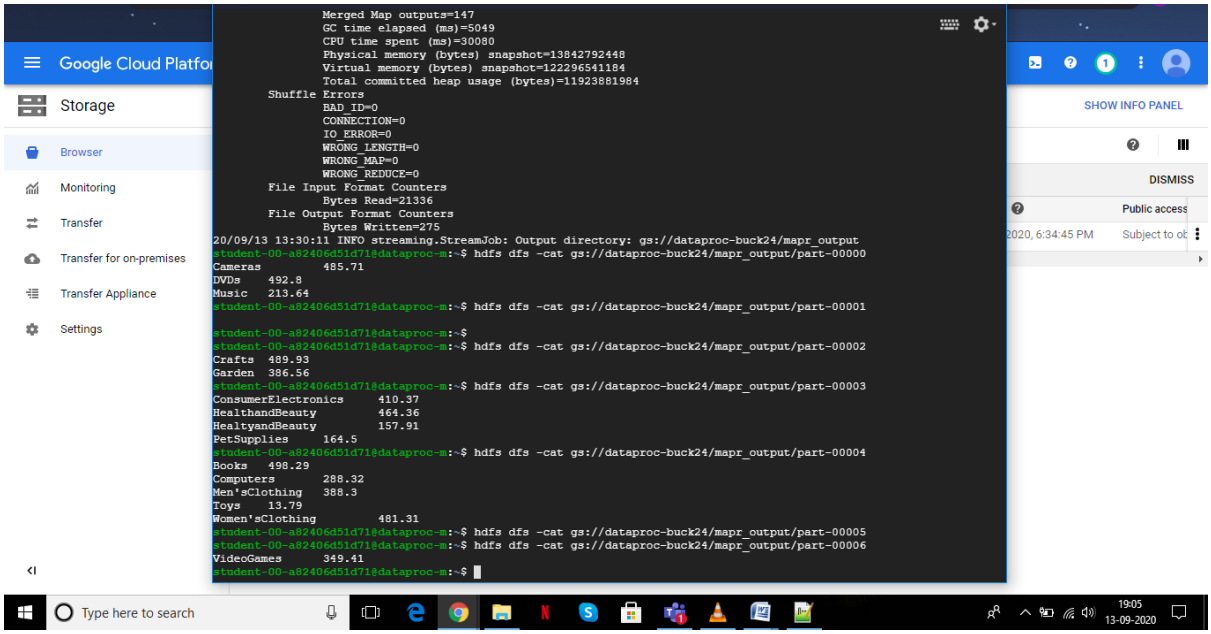


Fig11

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

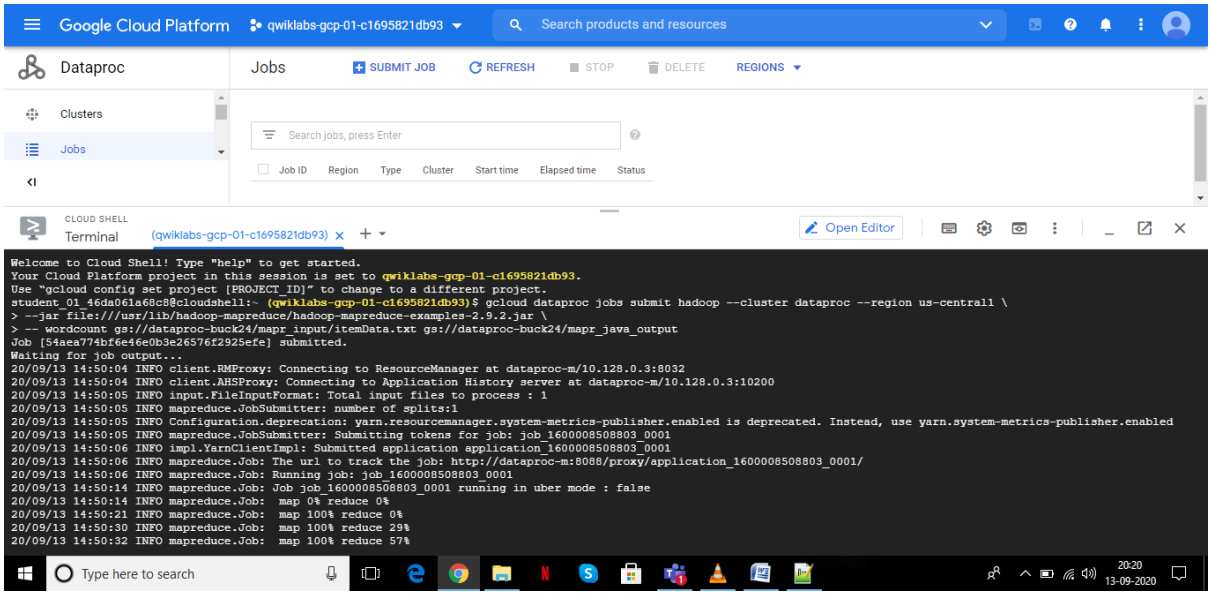


Fig1

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

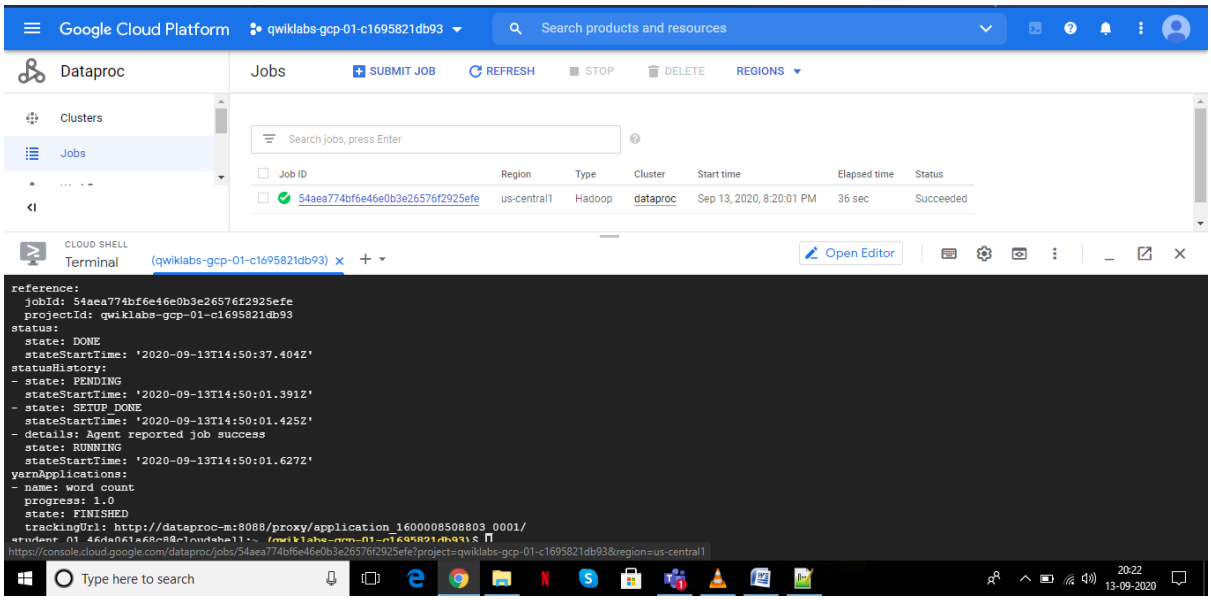


Fig2

- 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

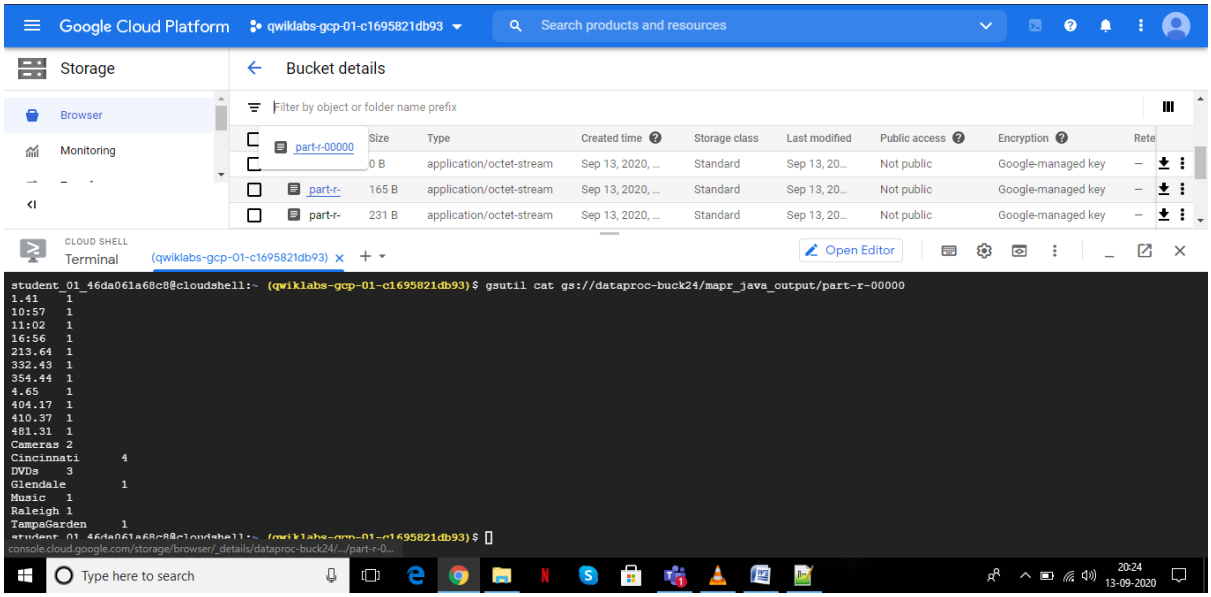

Google Cloud Platform
qwiklabs-gcp-01-c1695821db93
Search products and resources
Storage
Bucket details
Browser
Monitoring
Filter by object or folder name prefix
part-r-00000
Size
Type
Created time
Storage class
Last modified
Public access
Encryption
application/octet-stream
Standard
Not public
Google-managed key
CLOUD SHELL
Terminal
(qwiklabs-gcp-01-c1695821db93)
Open Editor
student_01_46da061a68c8@cloudshell:~ (qwiklabs-gcp-01-c1695821db93)$ gsutil cat gs://dataproc-buck24/mapr_java_output/part-r-00000
1.41 1
10:57 1
11:02 1
16:56 1
213.64 1
332.43 1
354.44 1
4.65 1
404.17 1
410.37 1
481.31 1
Cameras 2
Cincinnati 4
DVDs 3
Glendale 1
Music 1
Raleigh 1
TampaGarden 1
Type here to search
20:24
13-09-2020

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

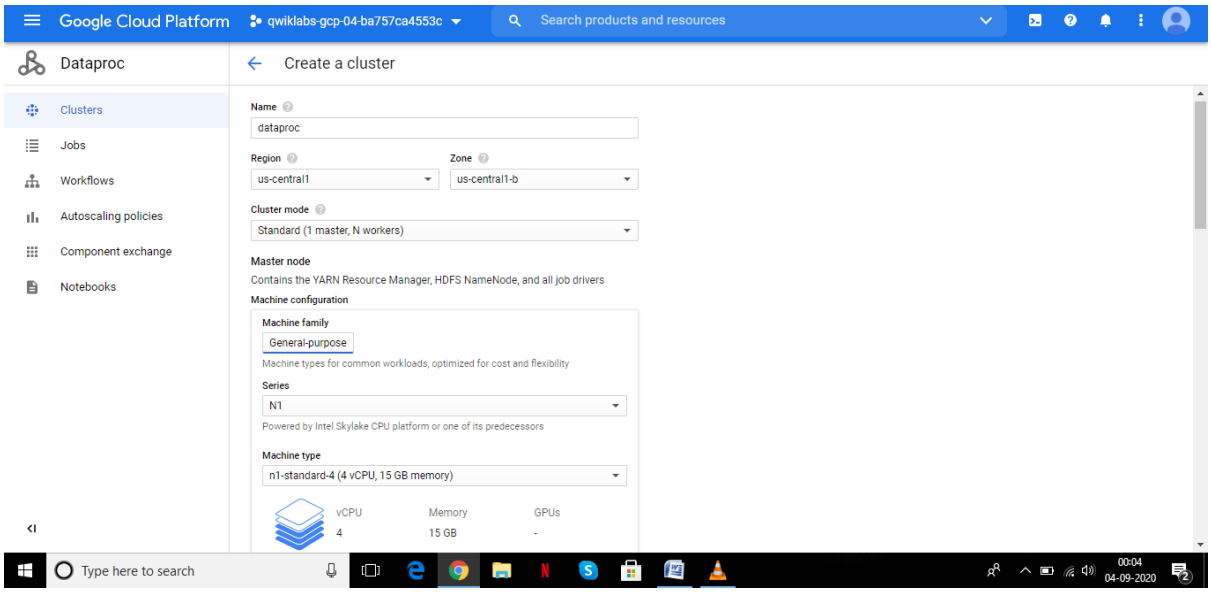


Fig1

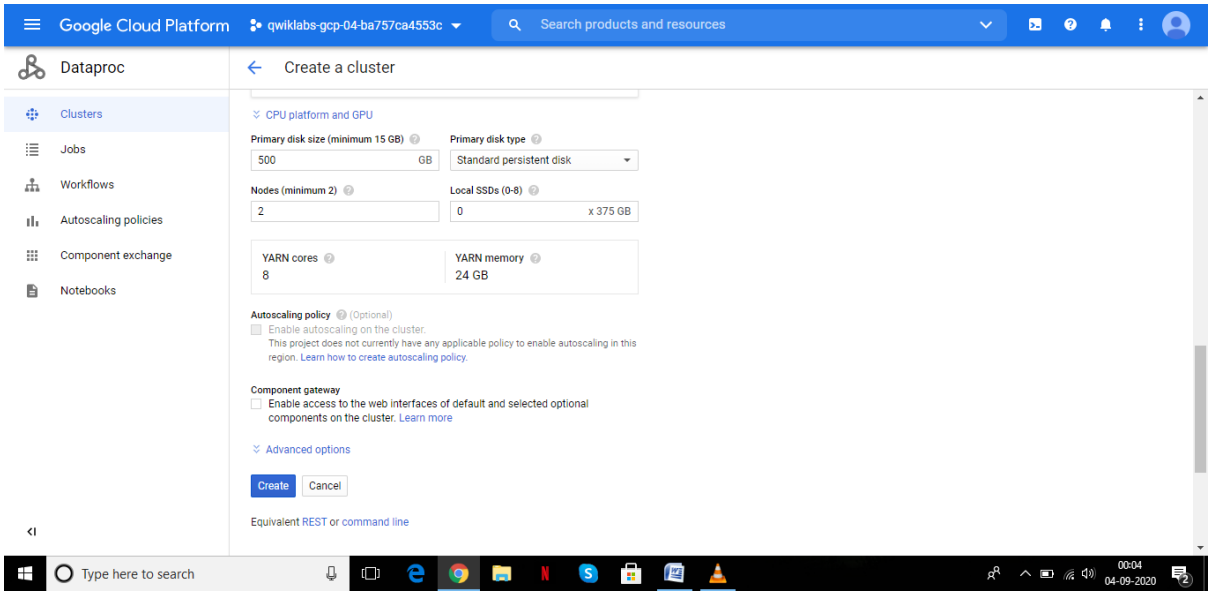


Fig2

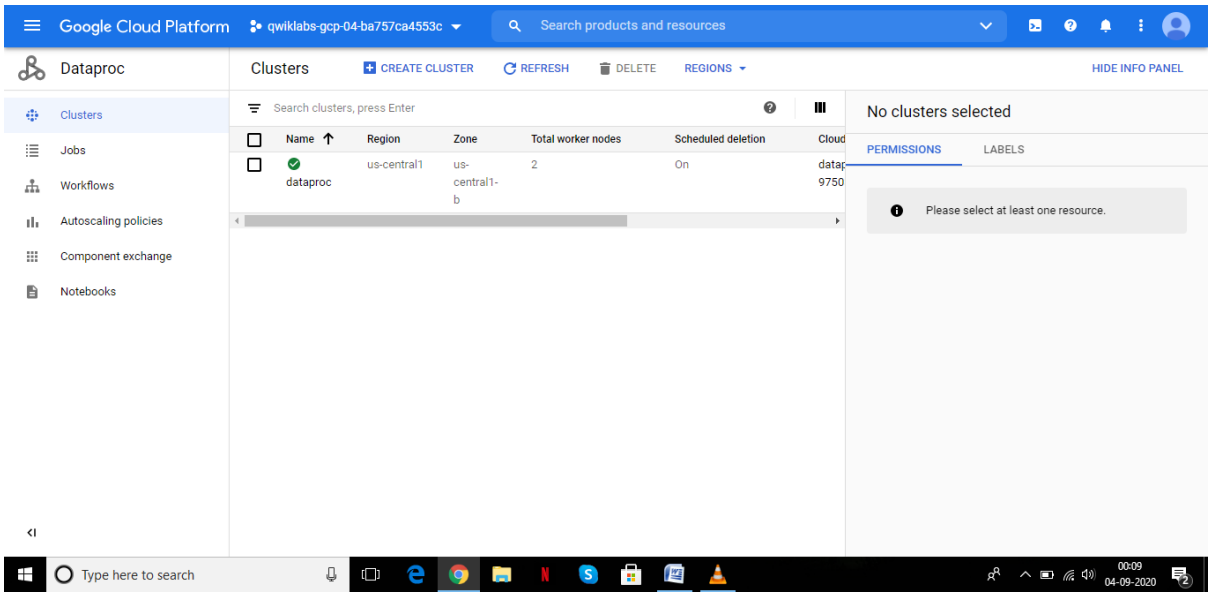


Fig3

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

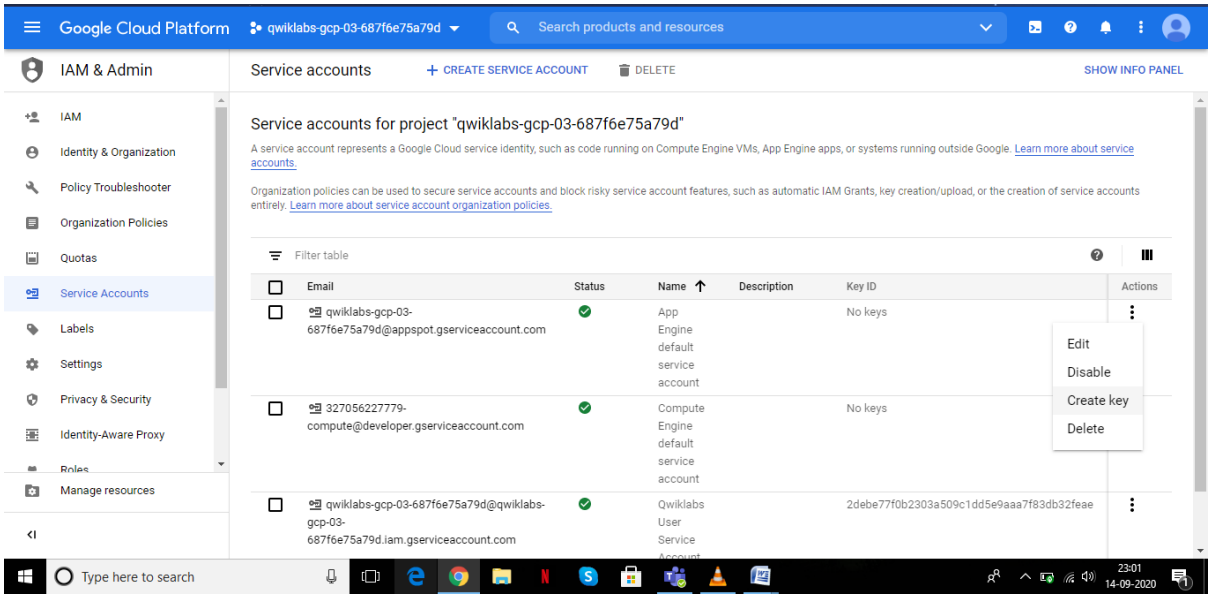


Fig4

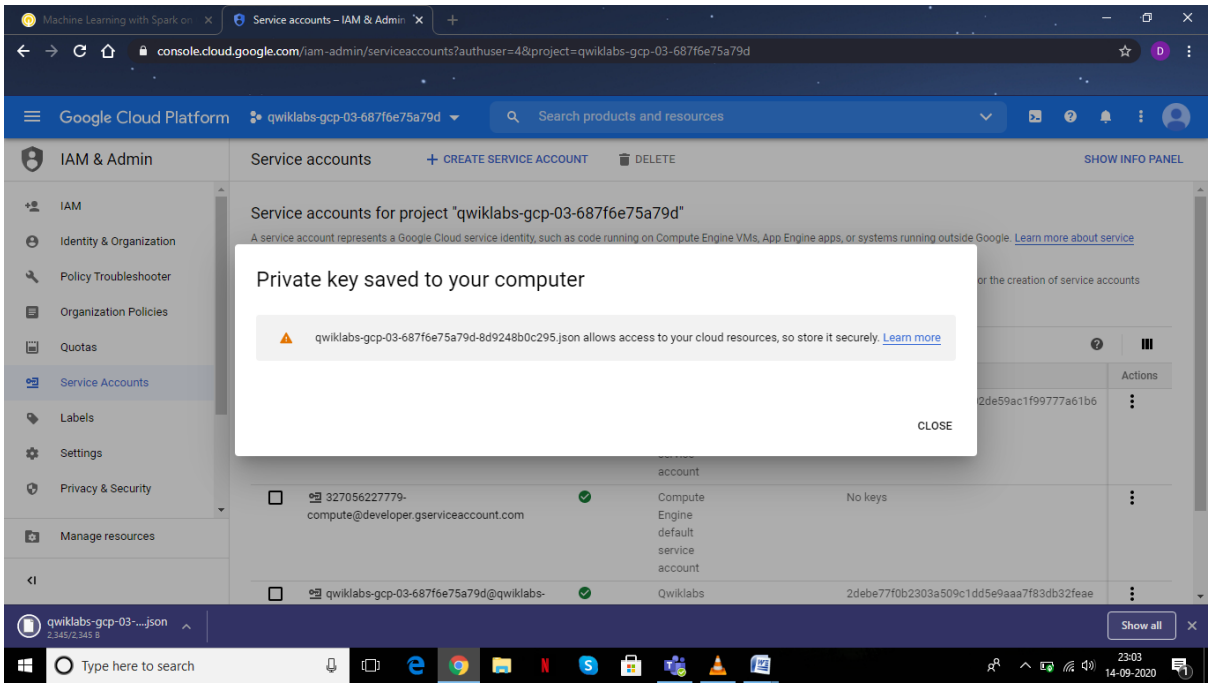


Fig5

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)

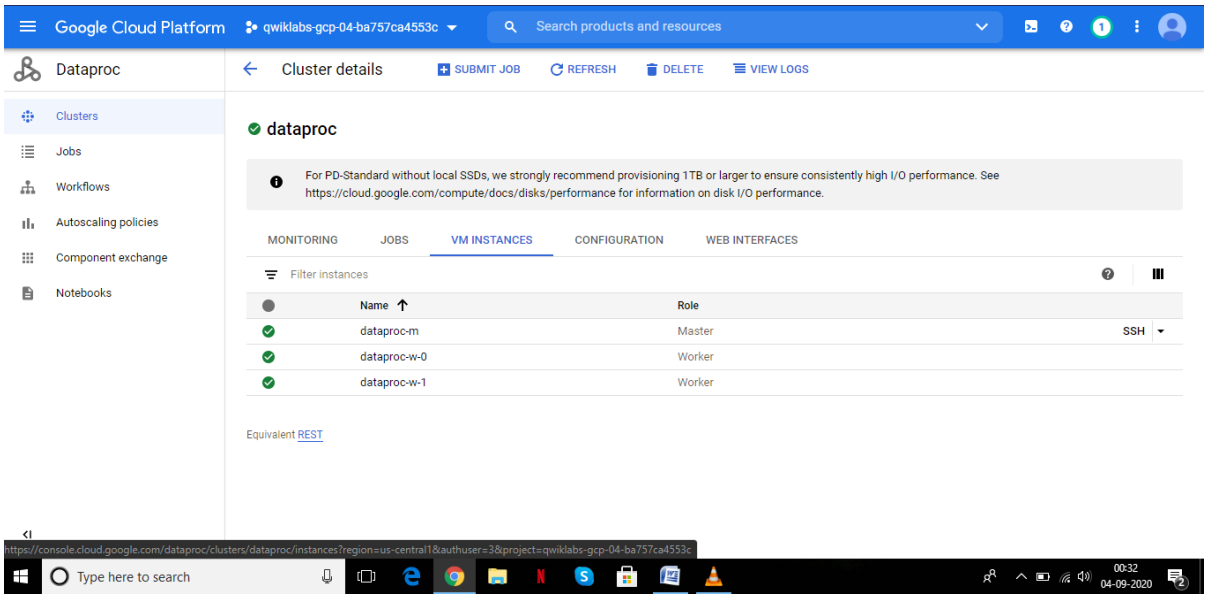


Fig6

Go to settings icon and click on upload a file

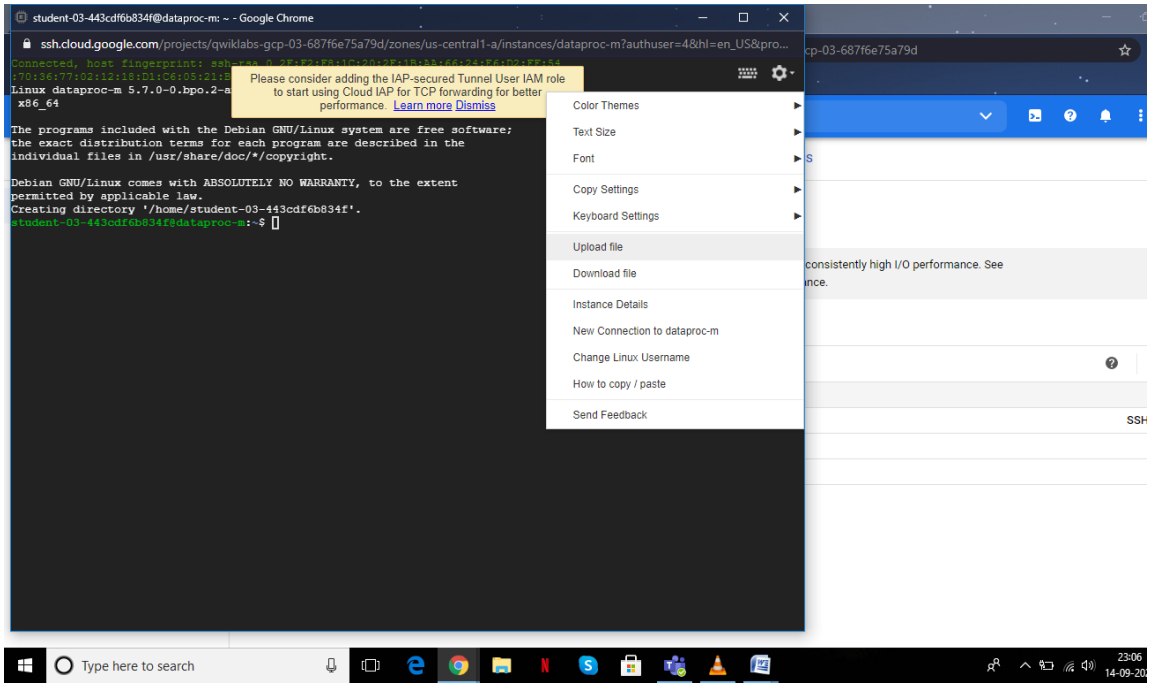


Fig7

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

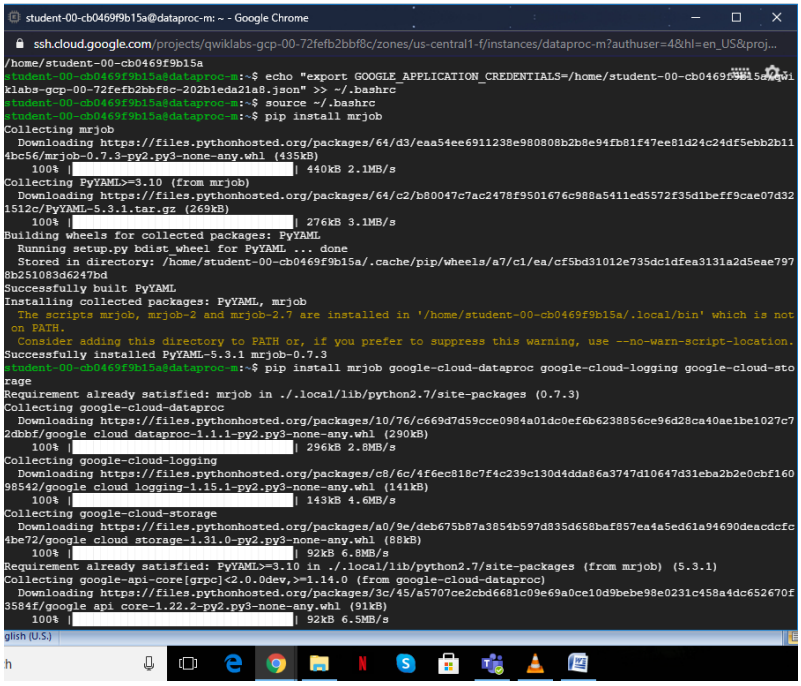


Fig8

4. Now upload the file and dataset similarly using the upload file option and type ‘ls’ to see the content

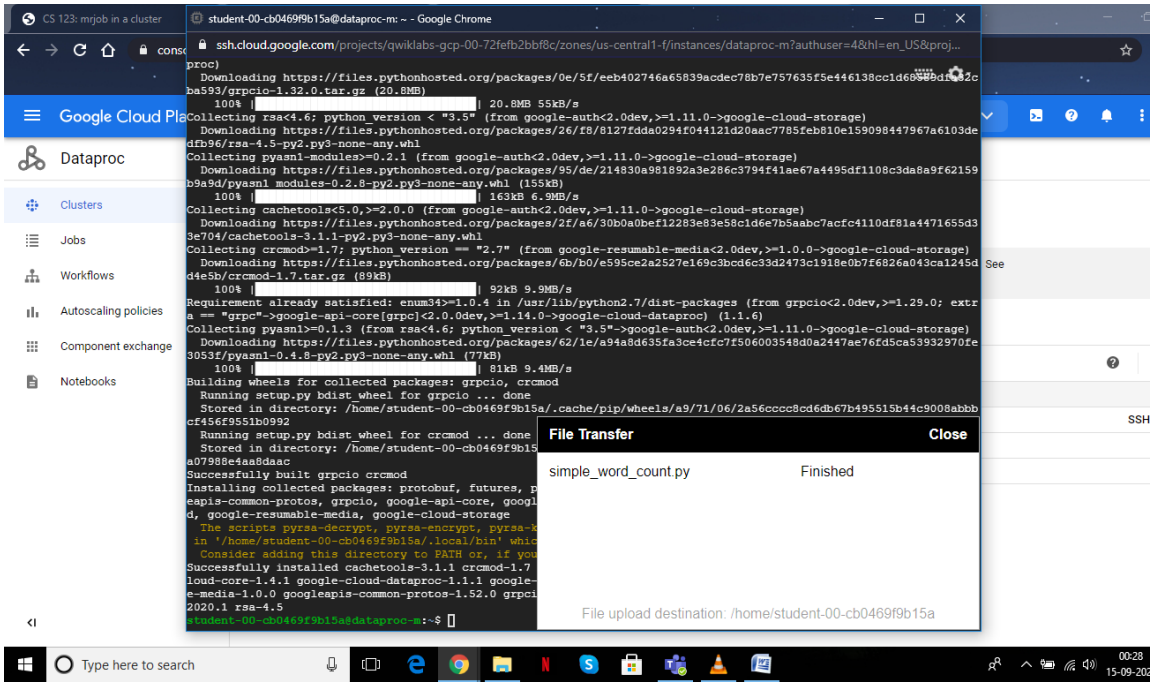


Fig9

5. To execute the script type the following command:

- python simple_word_count.py -r dataproc adventures.txt

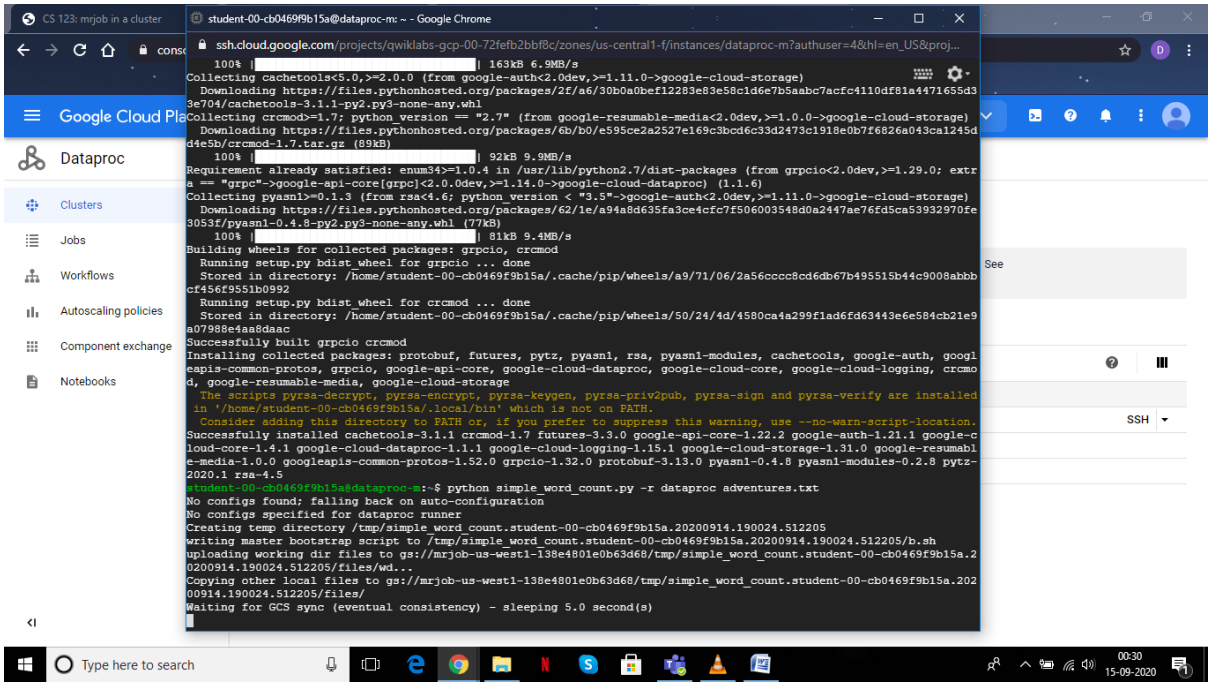


Fig10

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

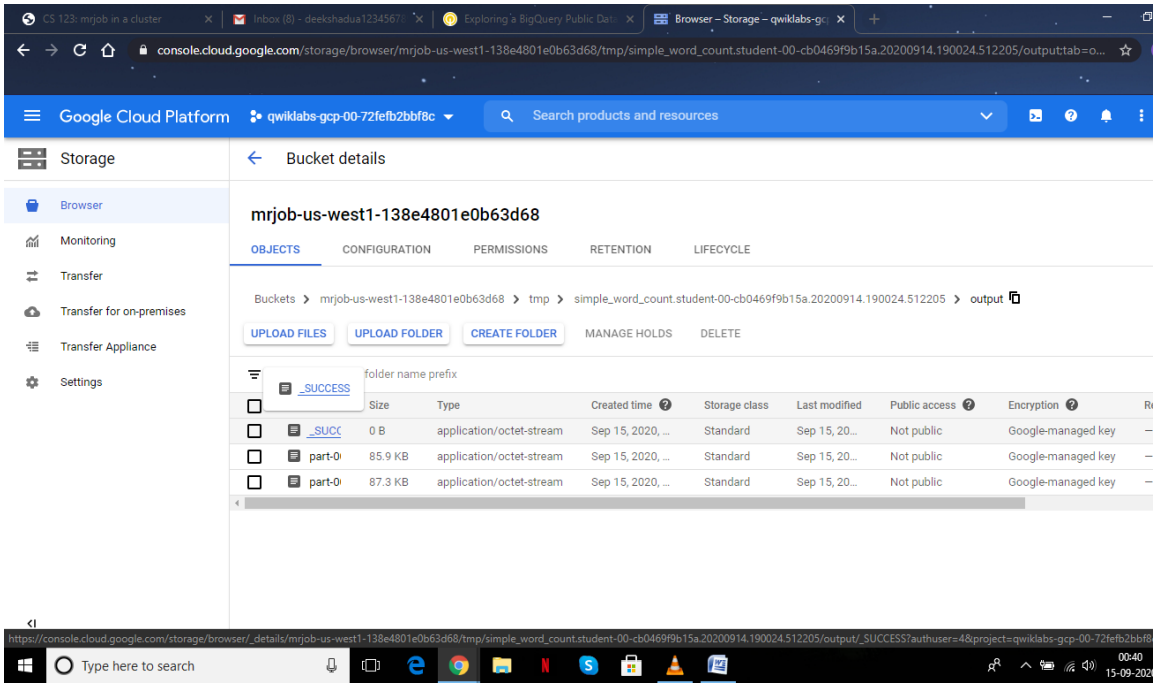


Fig11

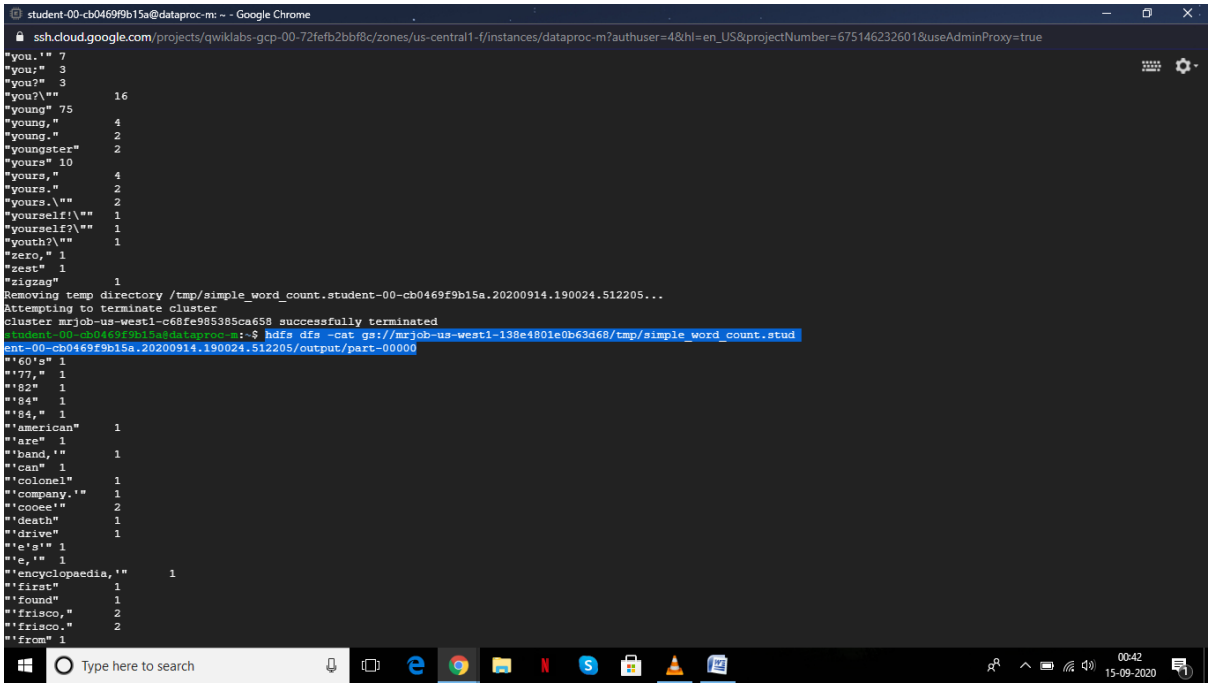


Fig12

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**

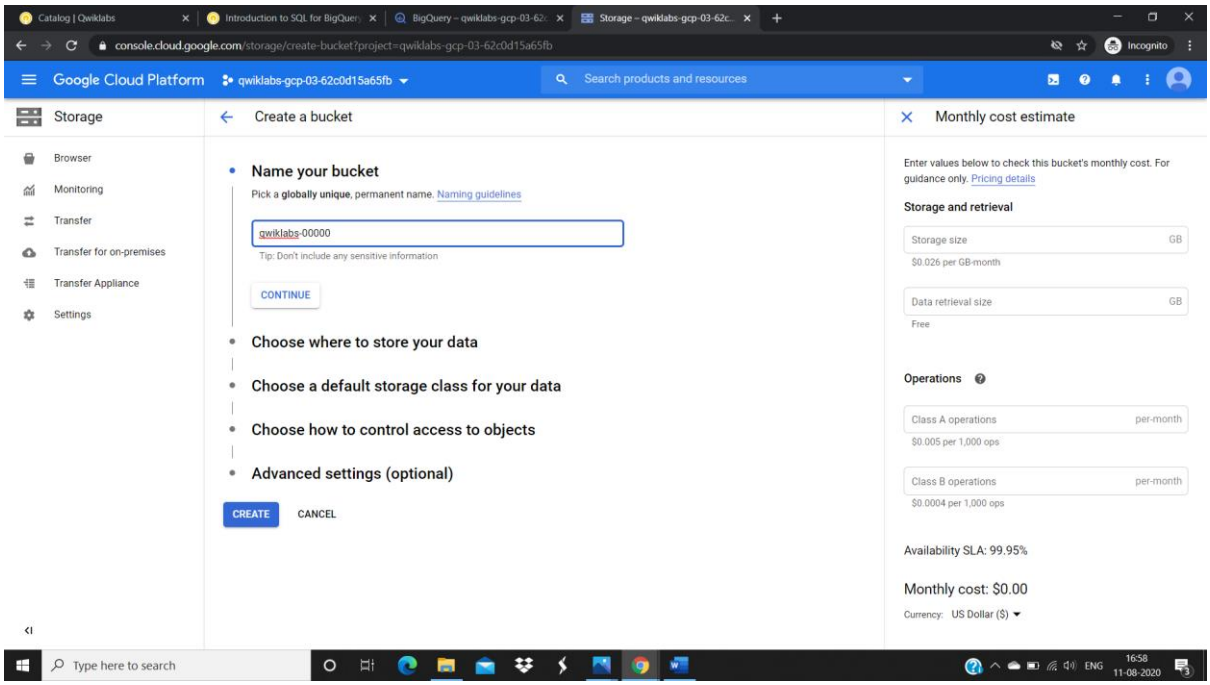


Fig1

After creating a cluster, Click on the cluster name

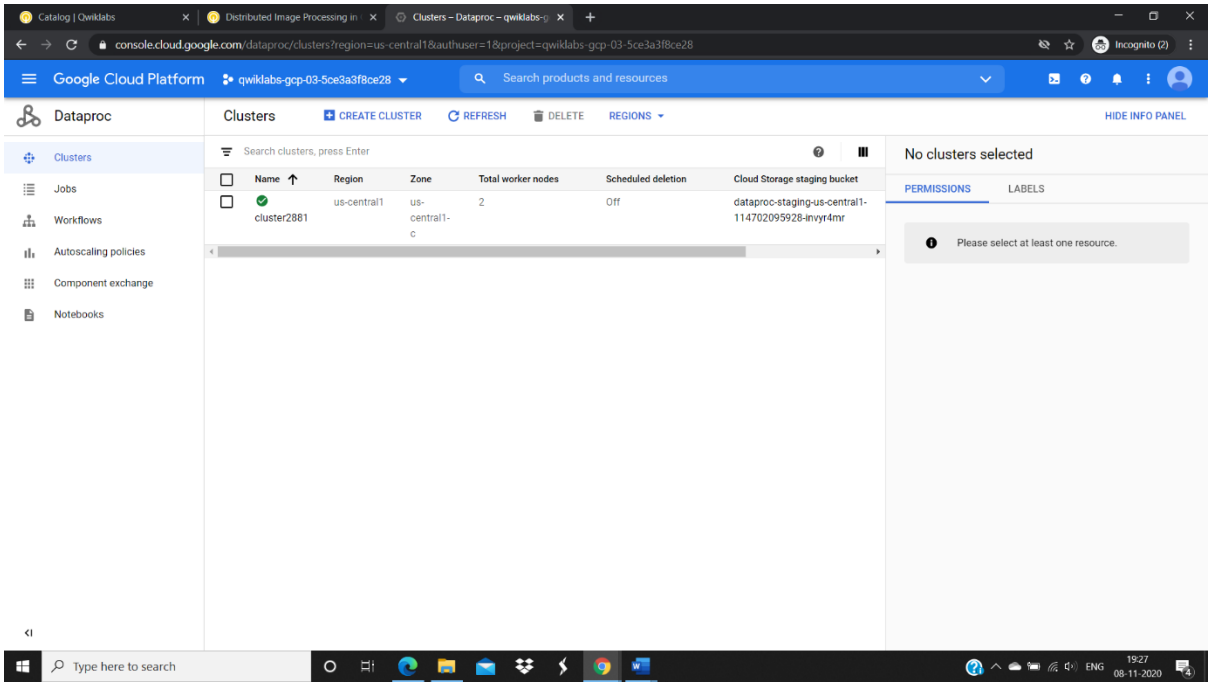


Fig2

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

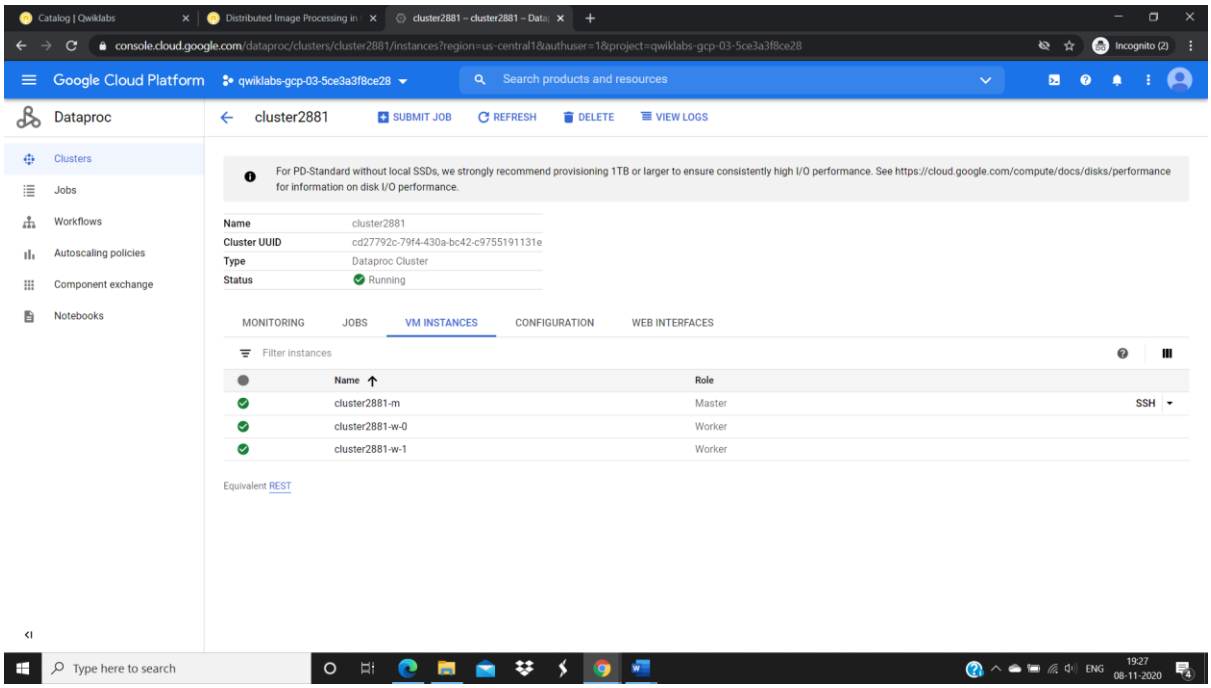


Fig3

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;

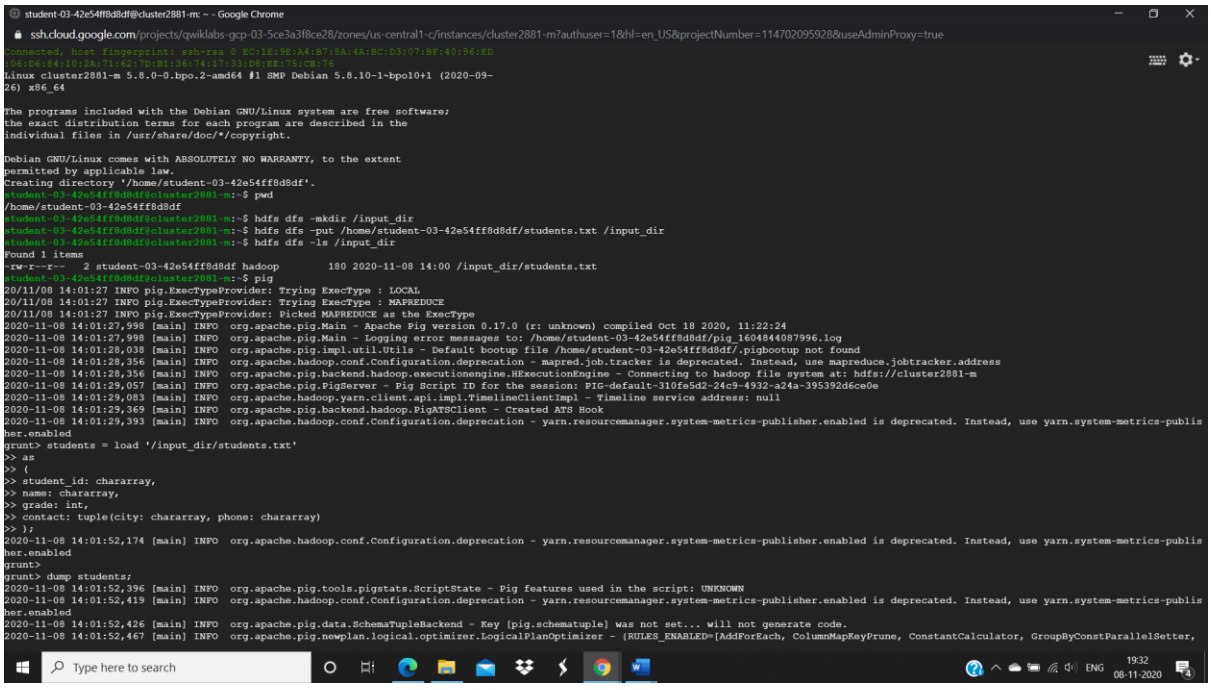


Fig4

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

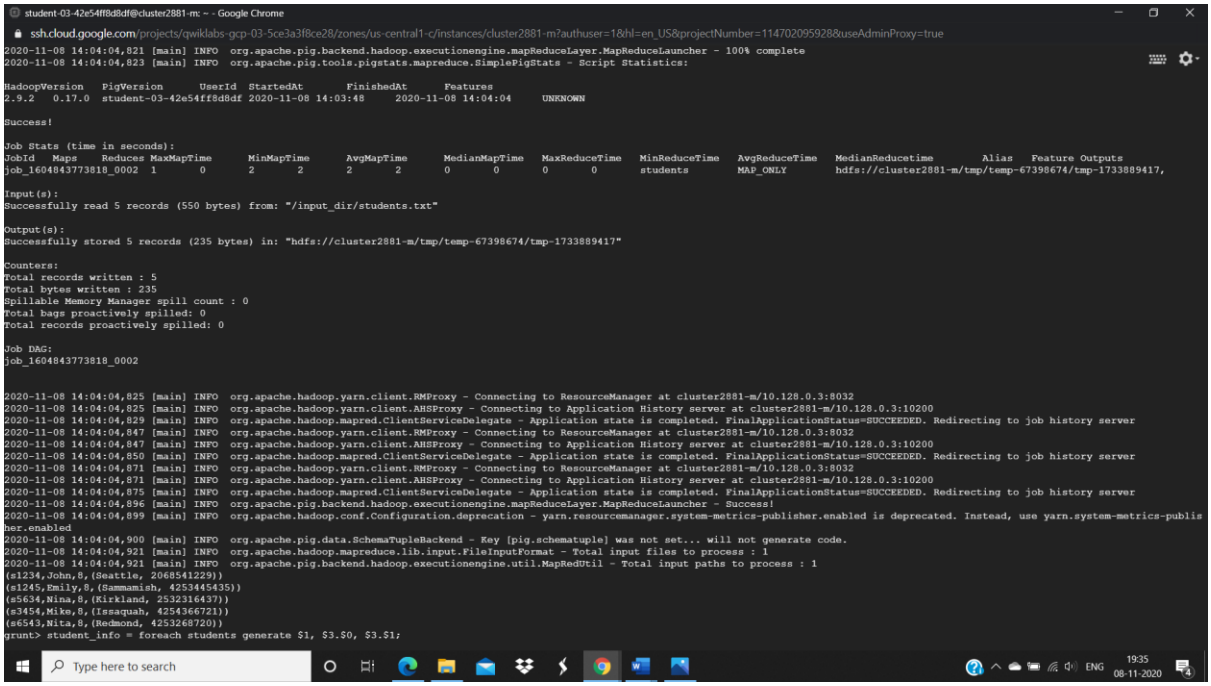


Fig5

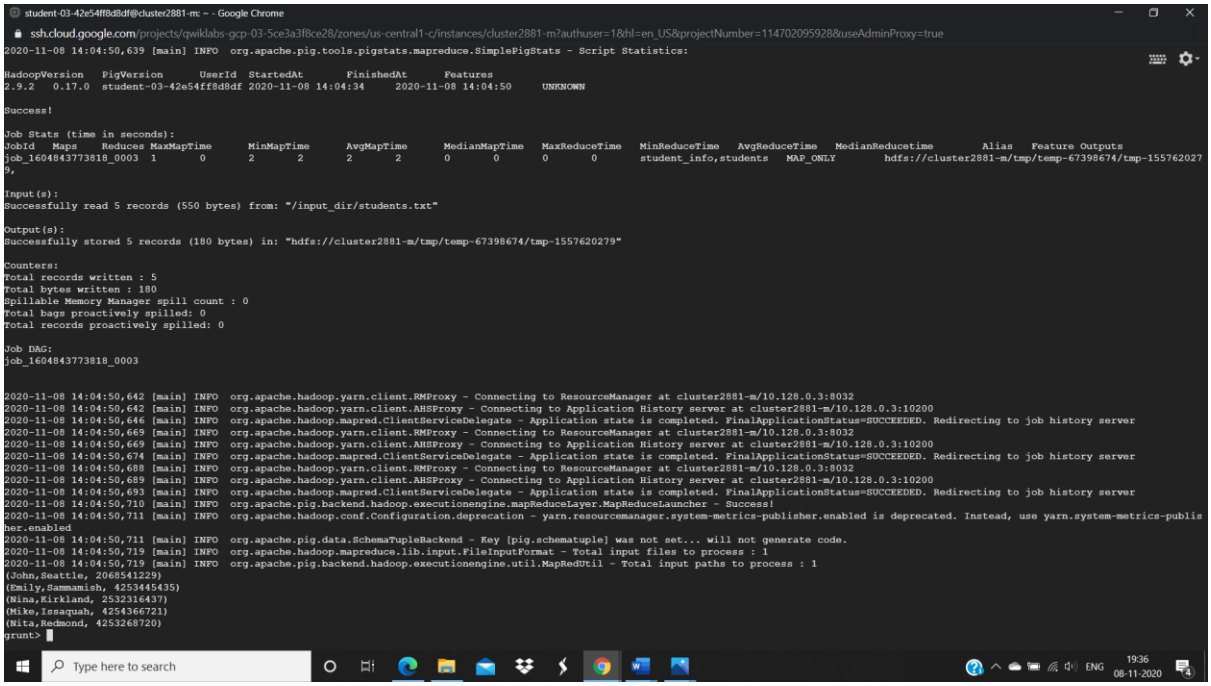


Fig6

- 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;

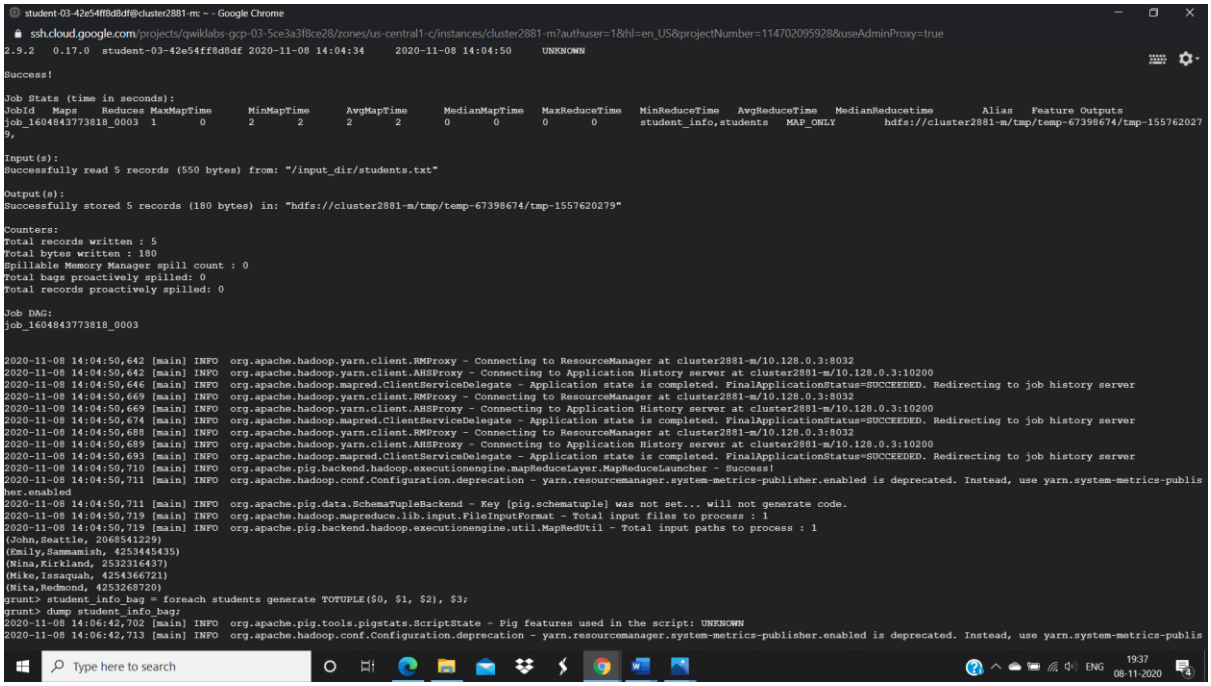


Fig7

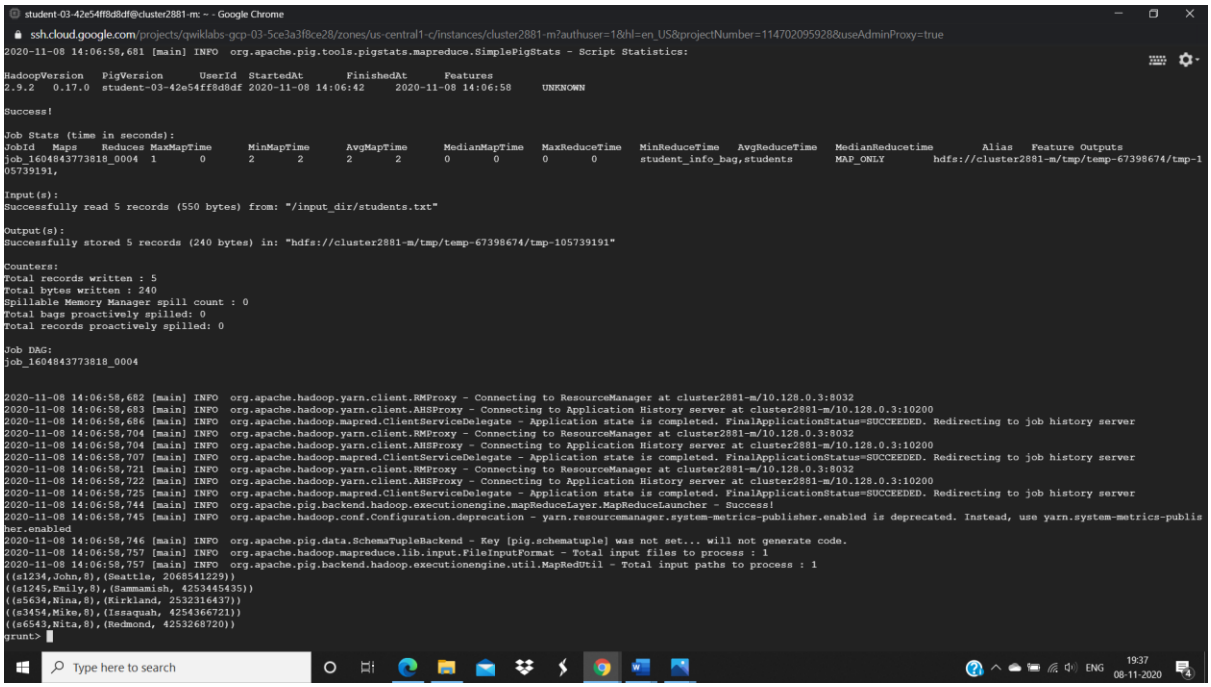


Fig8

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;

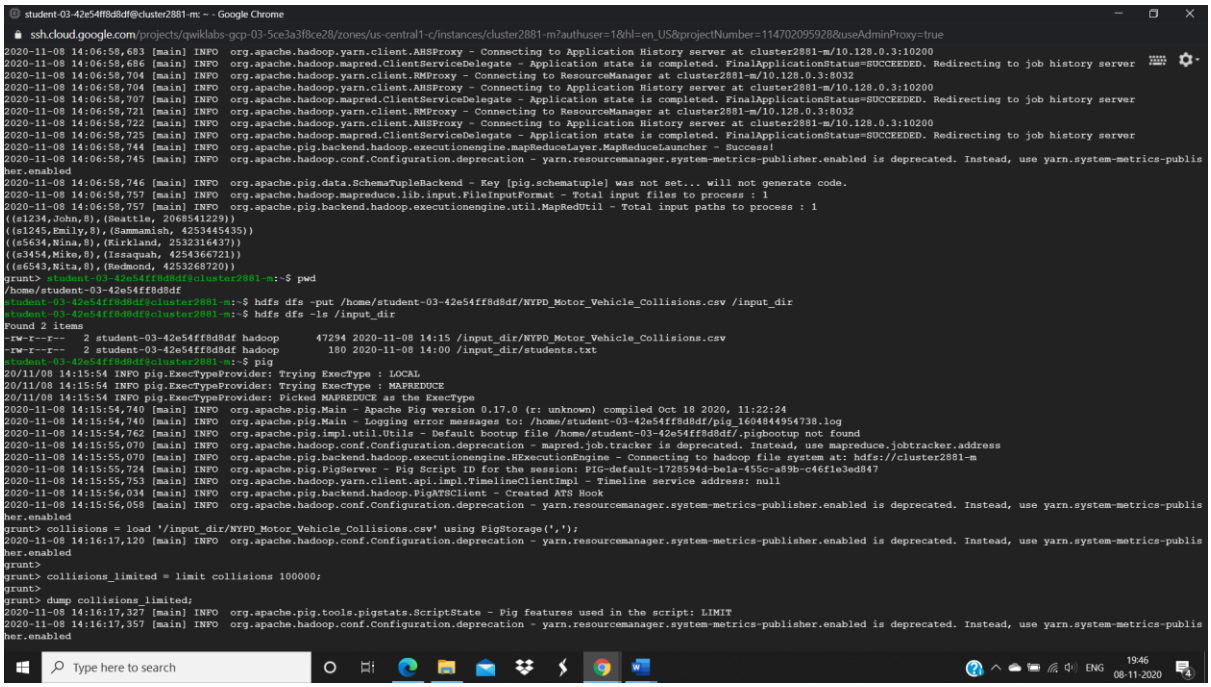


Fig9

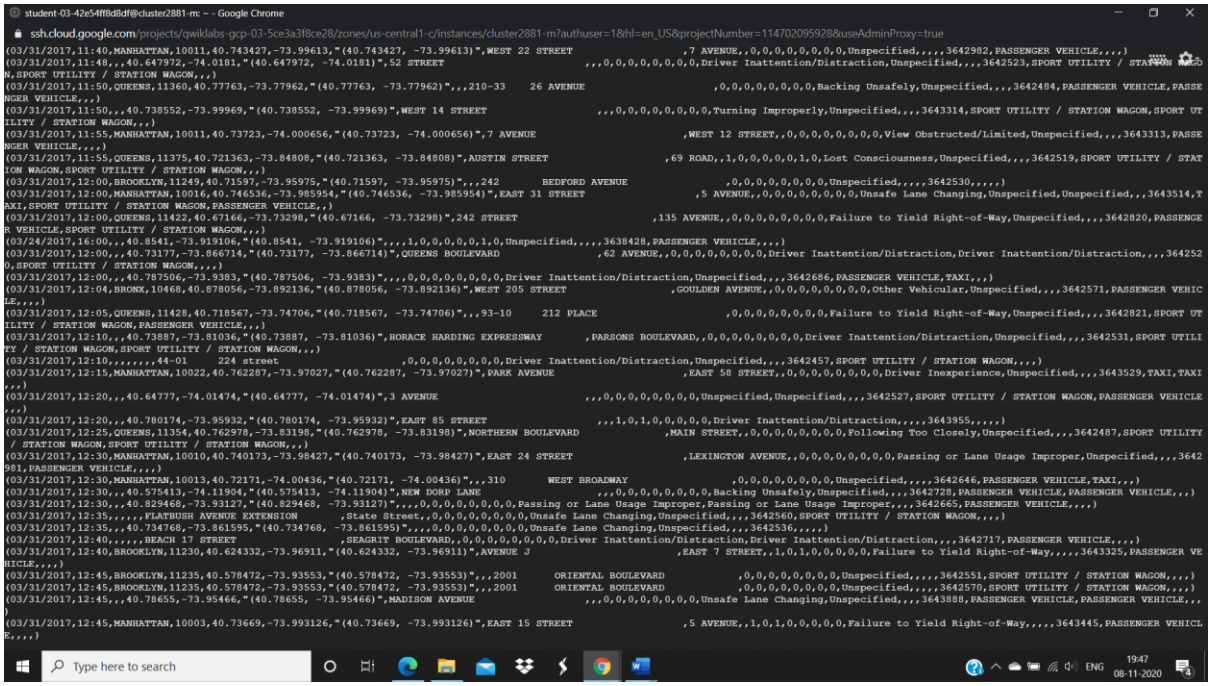


Fig10

- 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;

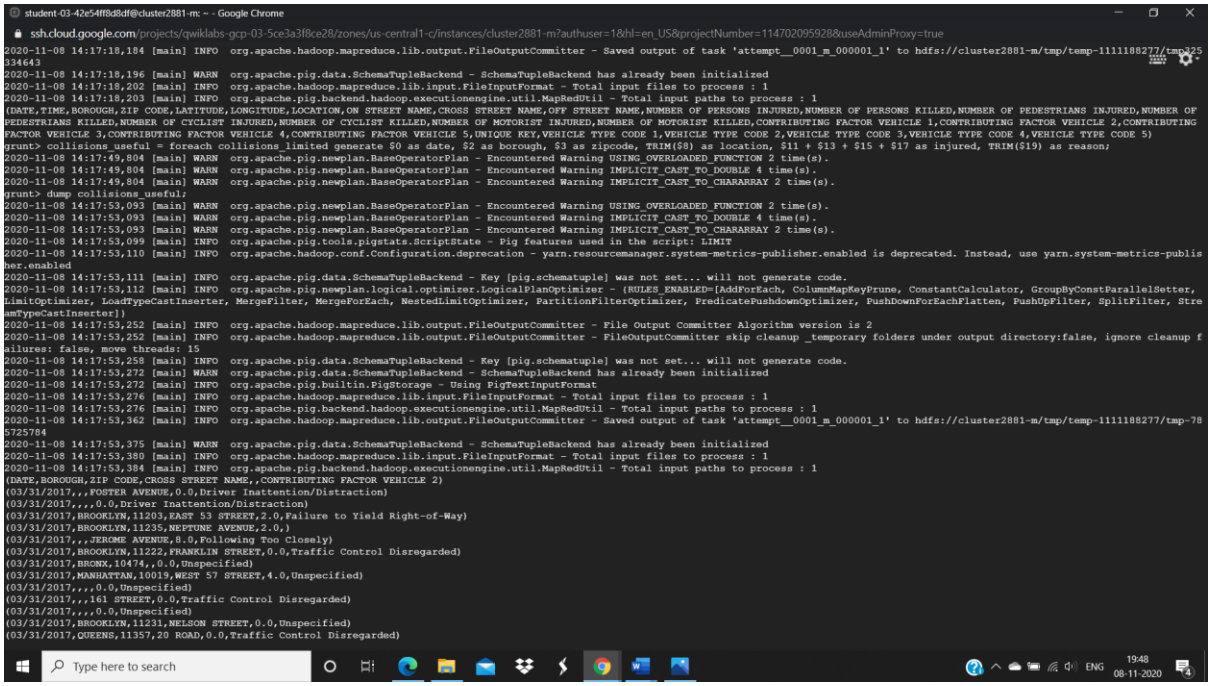


Fig11

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

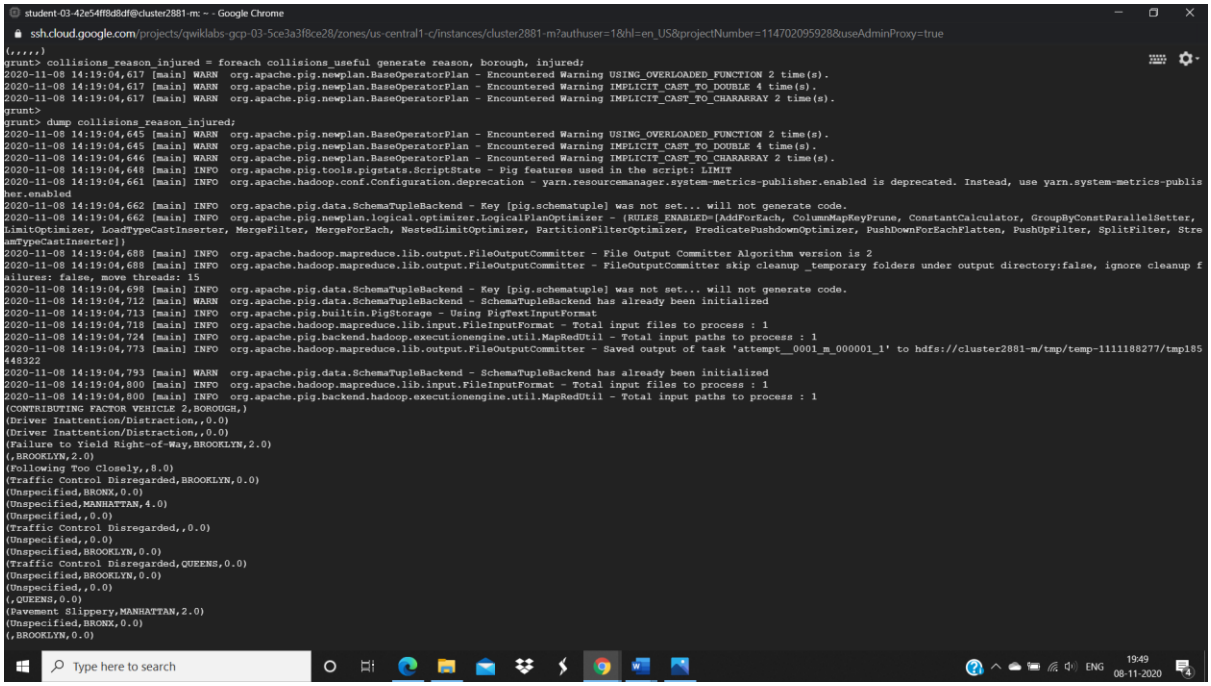


Fig12

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**

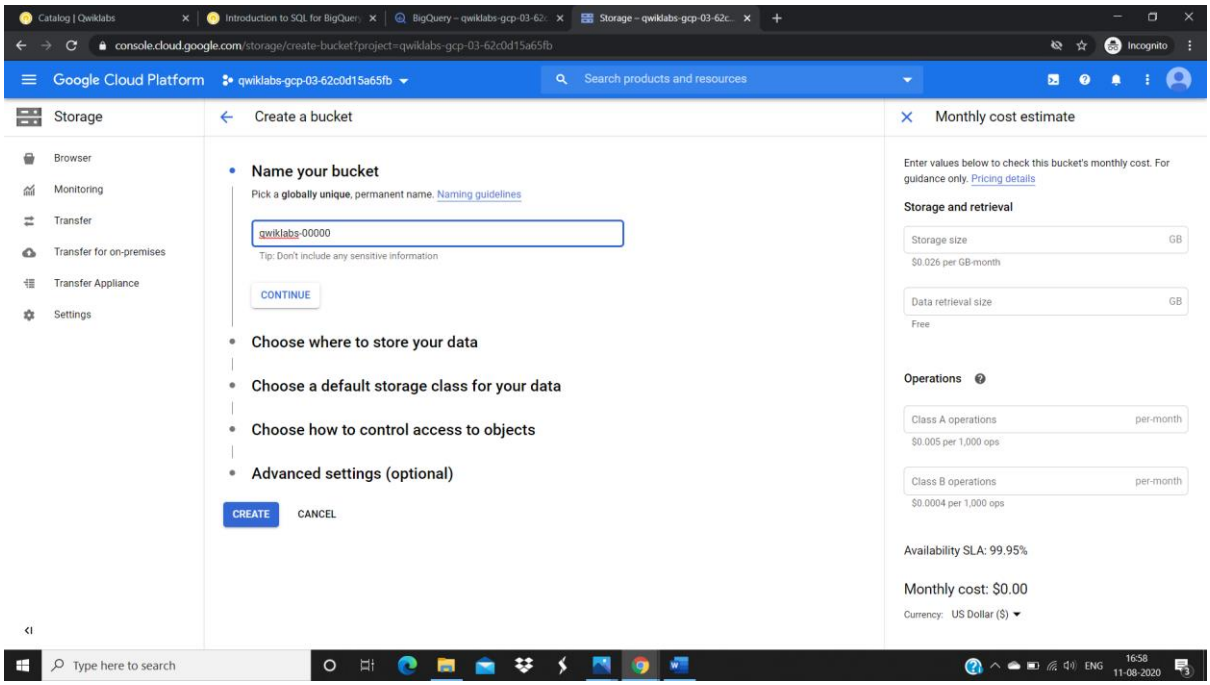


Fig1

After creating a cluster, Click on the cluster name

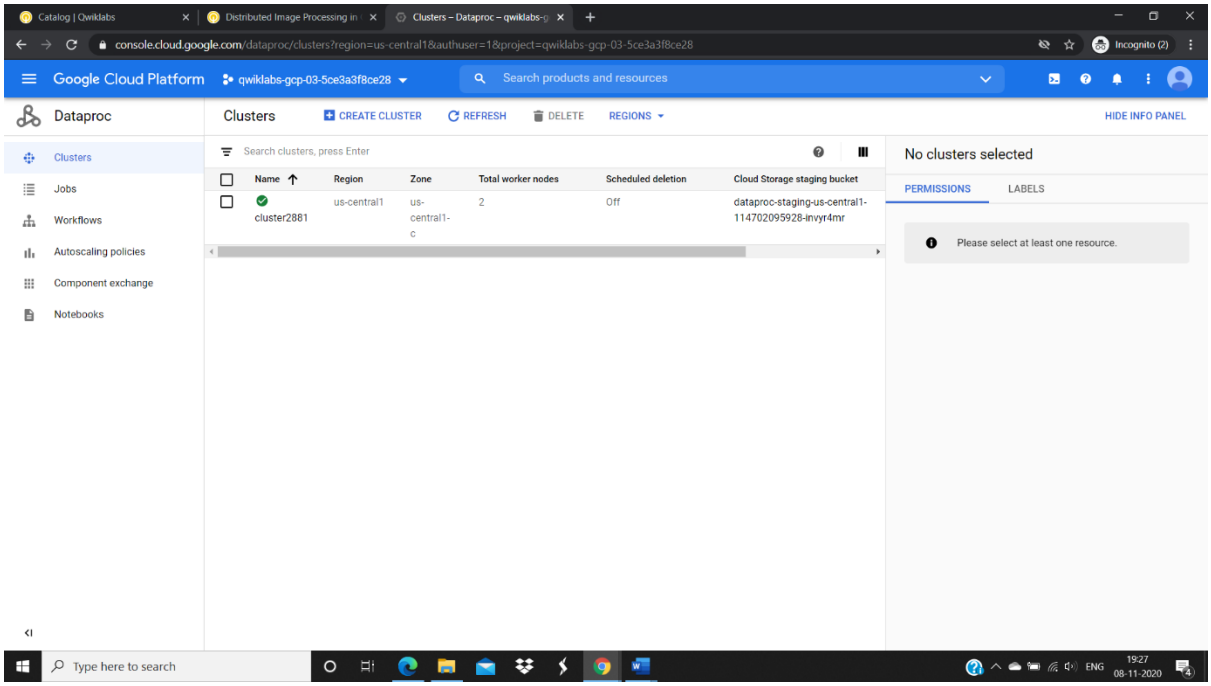


Fig2

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

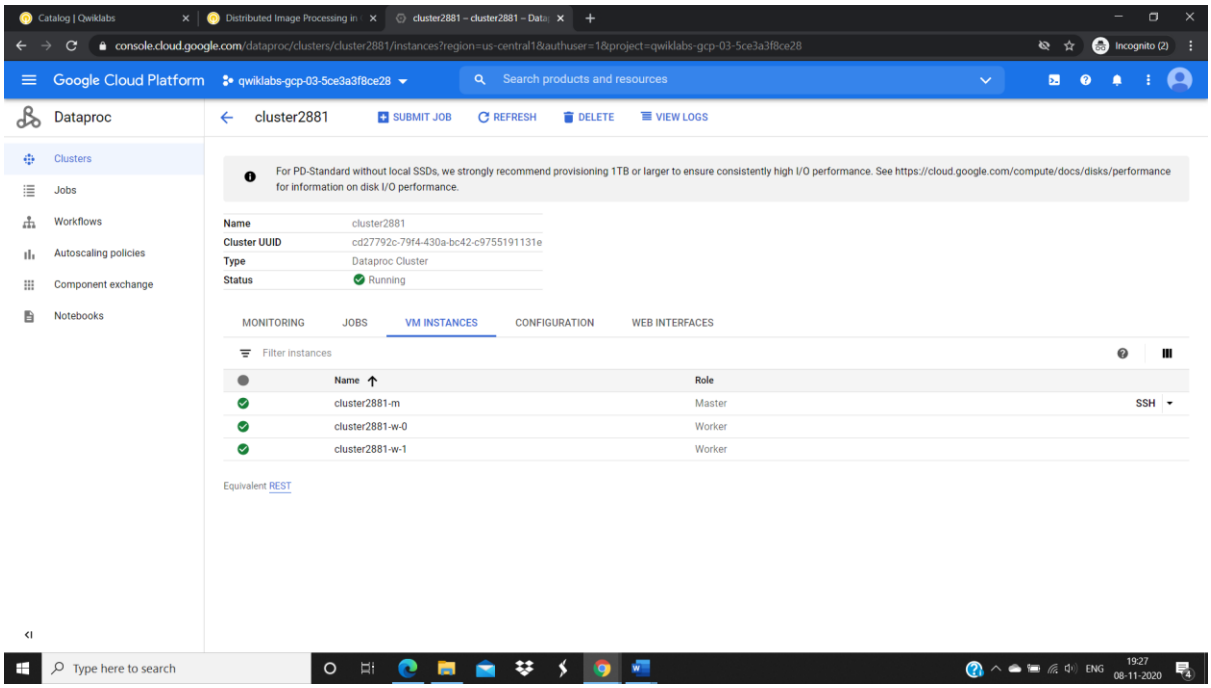


Fig3

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;

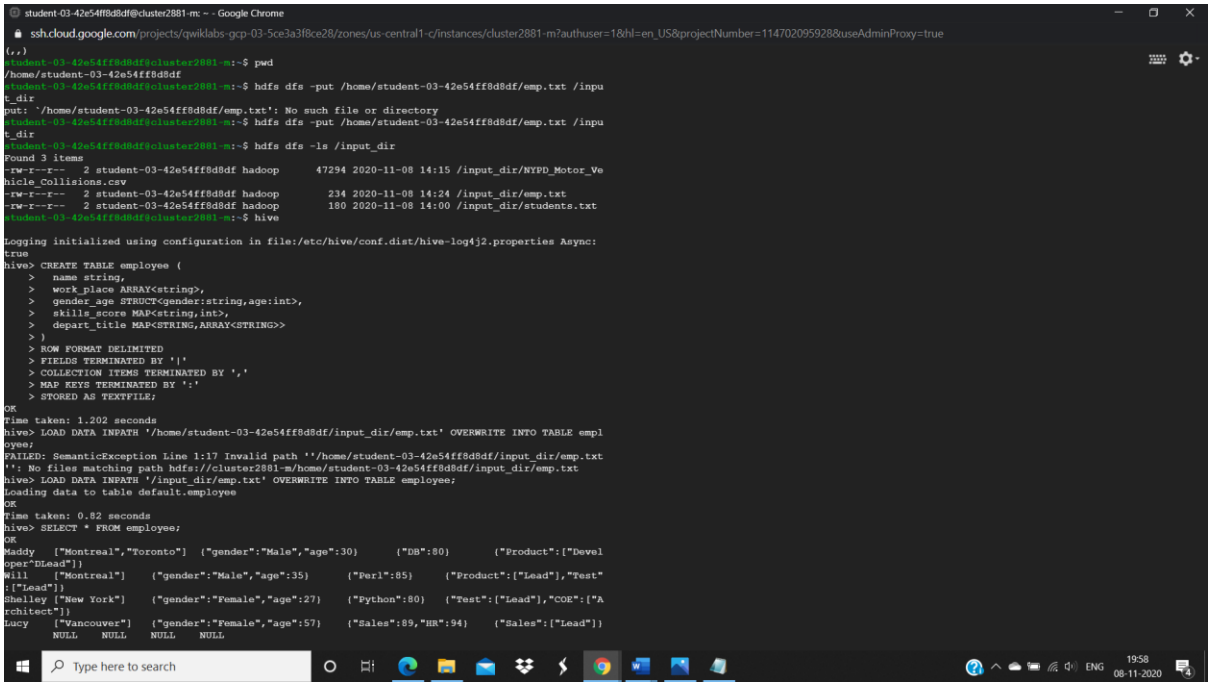


Fig4

--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;

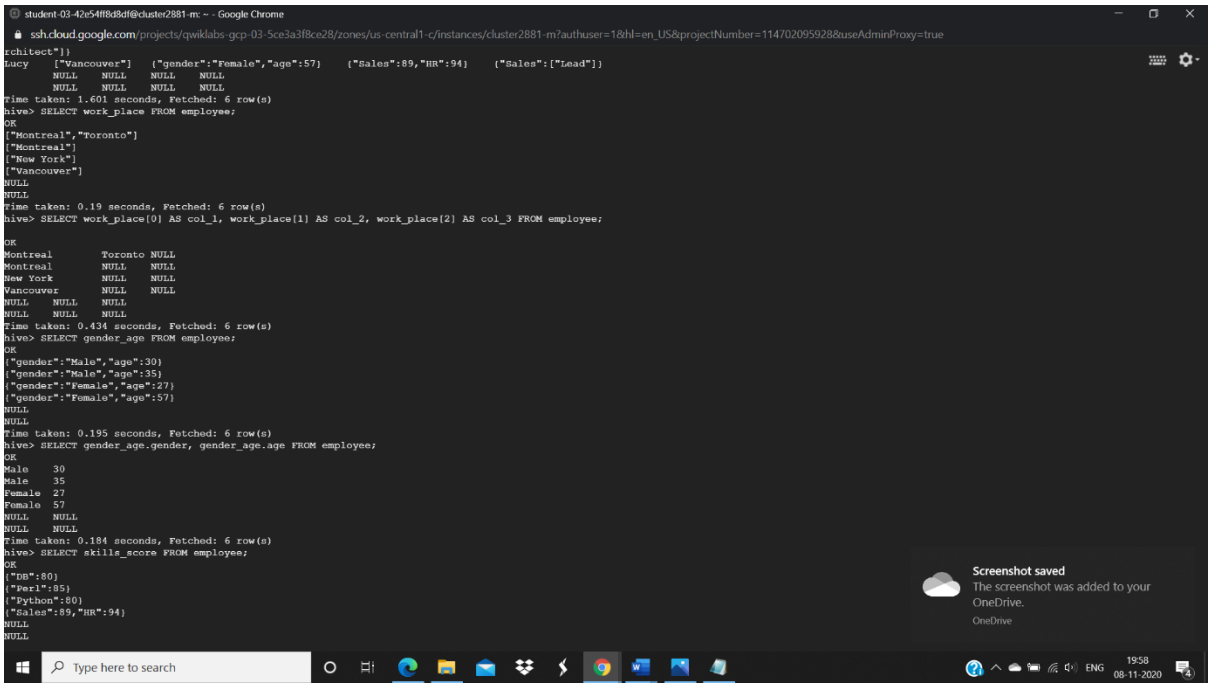


Fig5

- 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;

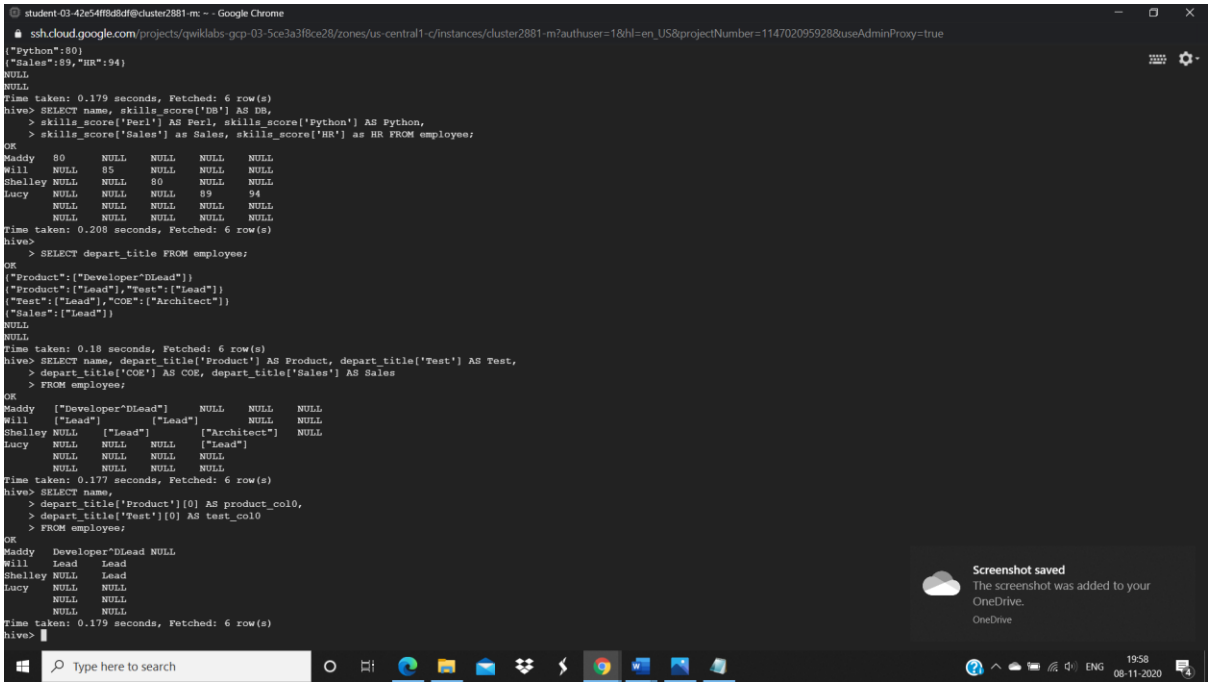


Fig6

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**

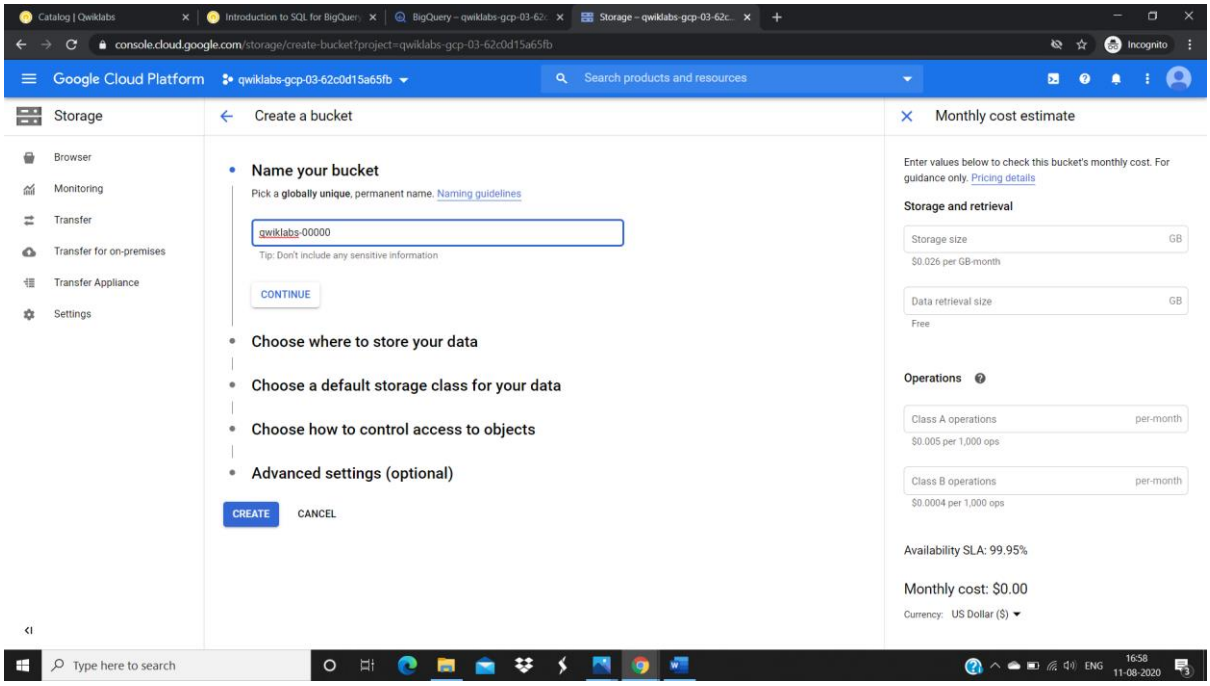


Fig1

After creating a cluster, Click on the cluster name

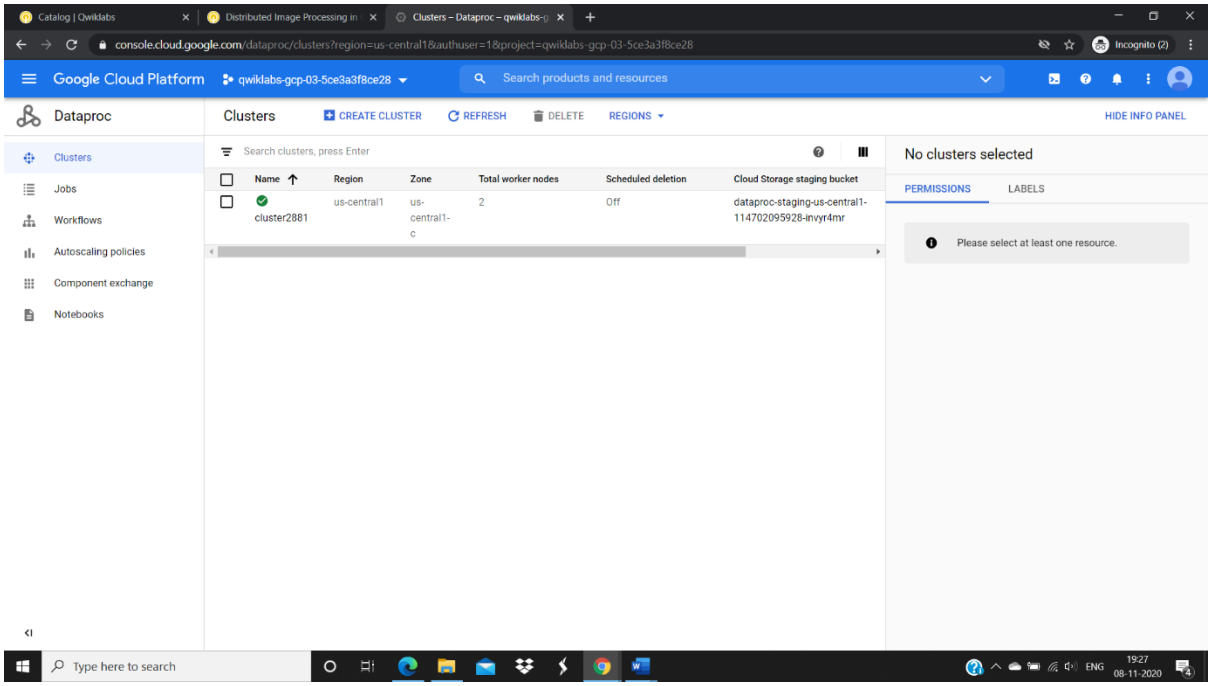


Fig2

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

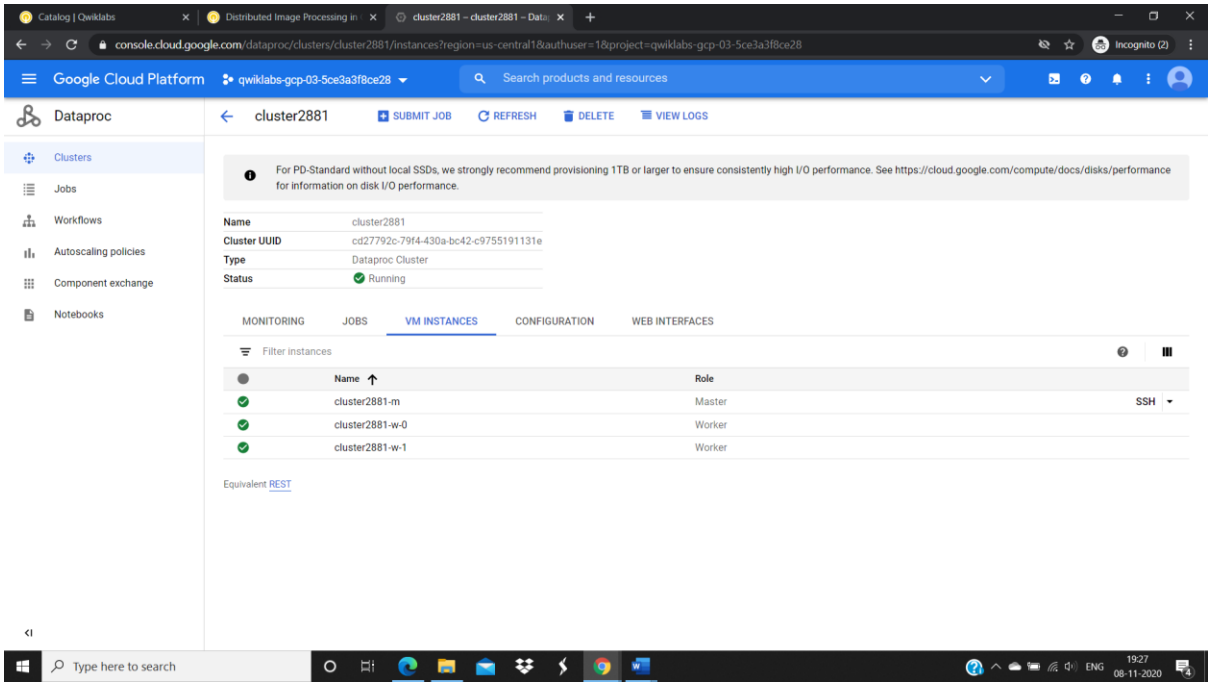


Fig3

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

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();

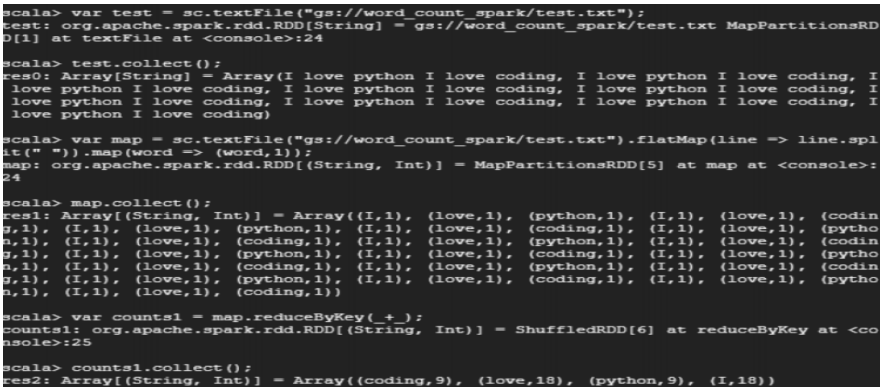


Fig5

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(_._2,true);
- Sort_result.collect();

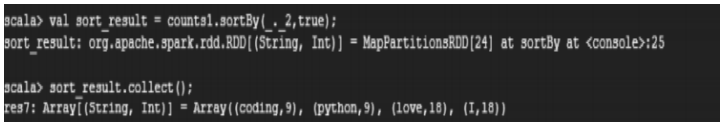


Fig6

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.

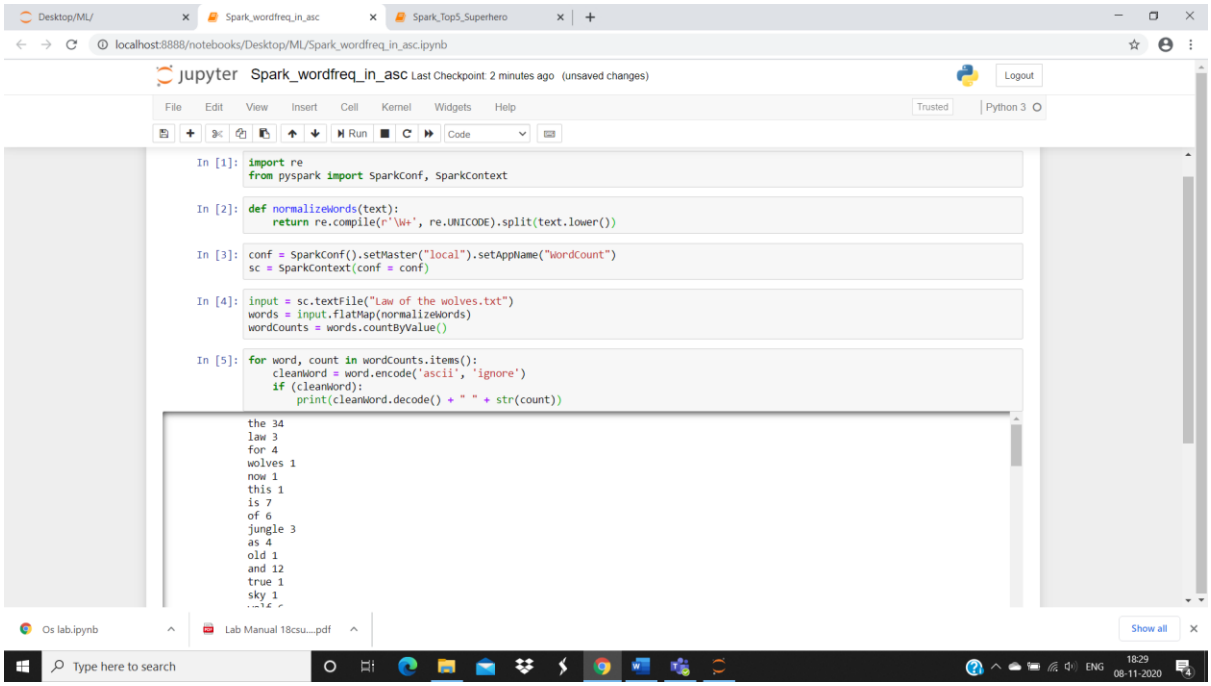


Fig1

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

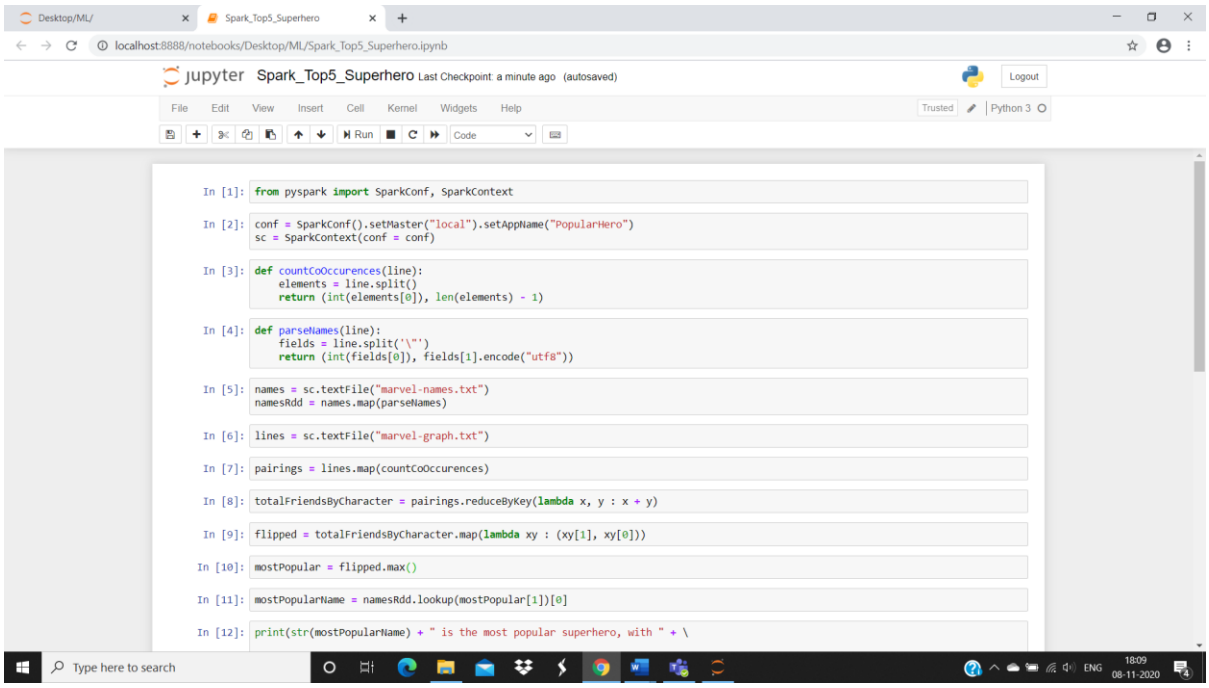


Fig1

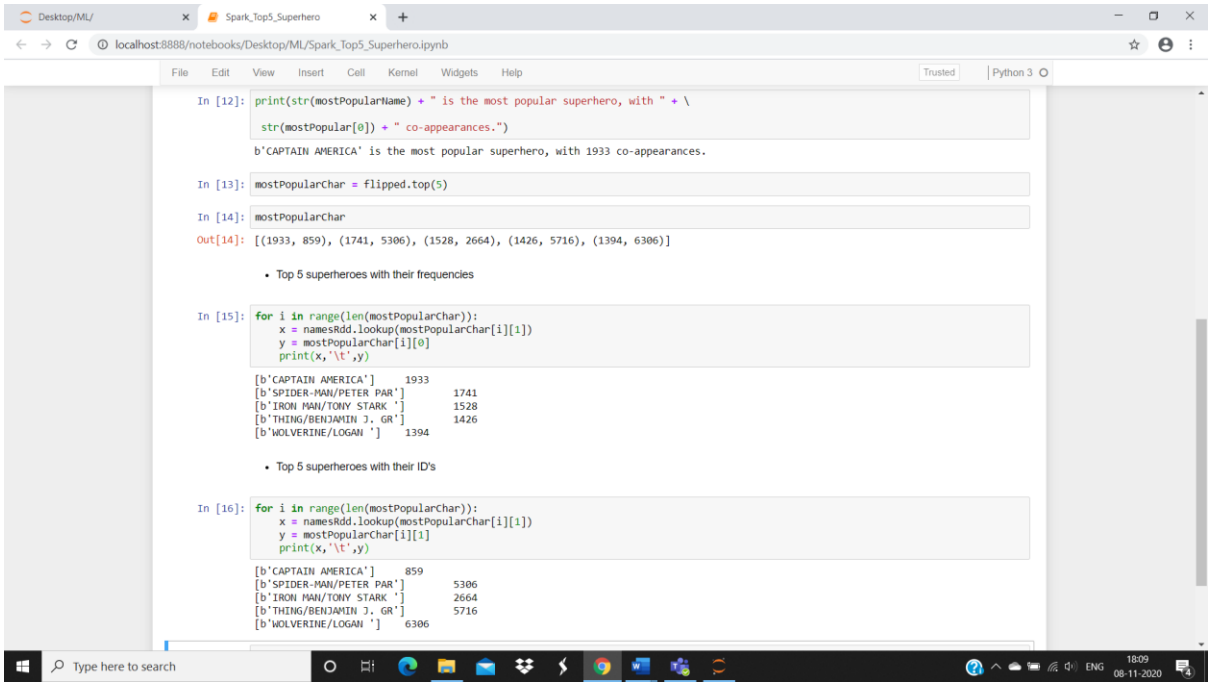

Desktop/ML/
Spark_Top5_Superhero
localhost:8888/notebooks/Desktop/ML/Spark_Top5_Superhero.ipynb
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
In [12]:
print(str(mostPopularName) + " is the most popular superhero, with " + \
str(mostPopular[0]) + " co-appearances.")
b'CAPTAIN AMERICA' 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)
[b'CAPTAIN AMERICA'] 1933
[b'SPIDER-MAN/PETER PAR'] 1741
[b'IRON MAN/TONY STARK '] 1528
[b'THING/BENJAMIN J. GR'] 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)
[b'CAPTAIN AMERICA'] 859
[b'SPIDER-MAN/PETER PAR'] 5306
[b'IRON MAN/TONY STARK '] 2664
[b'THING/BENJAMIN J. GR'] 5716
[b'WOLVERINE/LOGAN '] 6306
Type here to search
ENG
18:09
08-11-2020

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]:

```
from pyspark import SparkConf, SparkContext
def loadMovieNames():
    movieNames = {}
    with open("u.item") as f:
        for line in f:
            fields = line.split('|')
            movieNames[int(fields[0])] = fields[1]
    return movieNames
def parseInput(line):
    fields = line.split()
    return (int(fields[1]), (float(fields[2]), 1.0))

if __name__ == "__main__":
    conf = SparkConf().setAppName("WorstMovies")
    sc = SparkContext(conf = conf)
    movieNames = loadMovieNames()
    lines = sc.textFile("u.data")
    movieRatings = lines.map(parseInput)
    ratingTotalsAndCount = movieRatings.reduceByKey(lambda movie1, movie2: ( movie1[0] + movie2[0], movie1[1] + movie2[1] )
    popularTotalsAndCount = ratingTotalsAndCount.filter(lambda key_val: key_val[1][1] > 10)
    averageRatings = popularTotalsAndCount.mapValues(lambda totalAndCount : totalAndCount[0] / totalAndCount[1])
    sortedMovies = averageRatings.sortBy(lambda x: x[1])
    results = sortedMovies.take(1)
    for result in results:
        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 Algorihms

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.

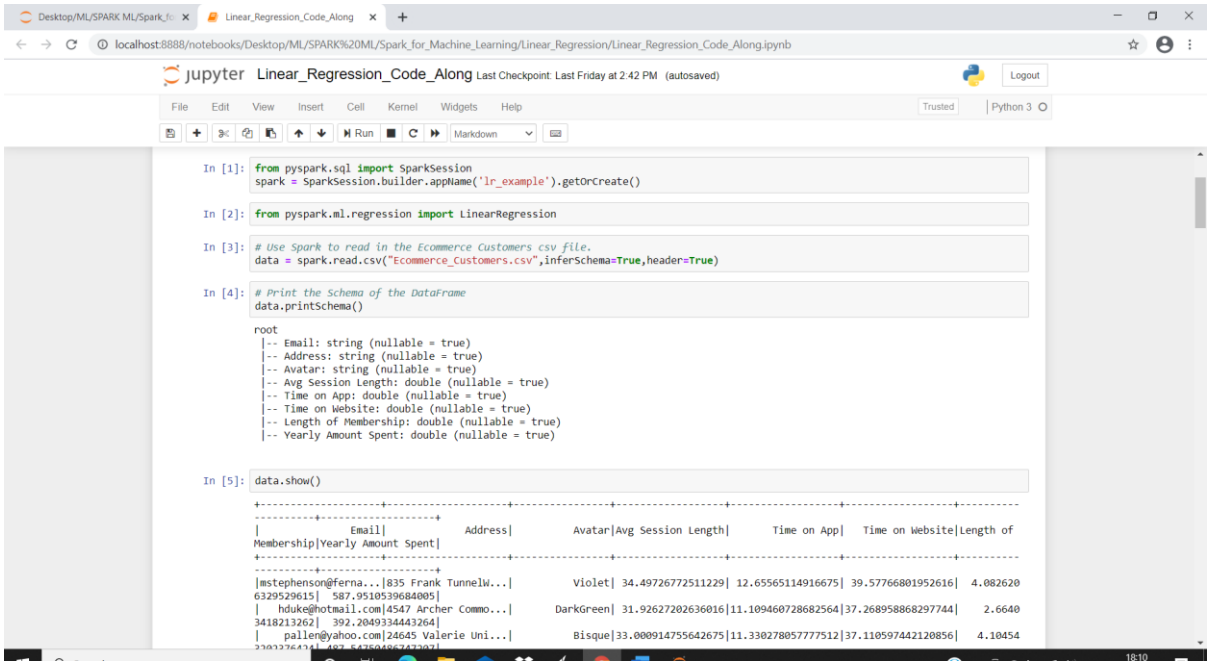


Fig1

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.

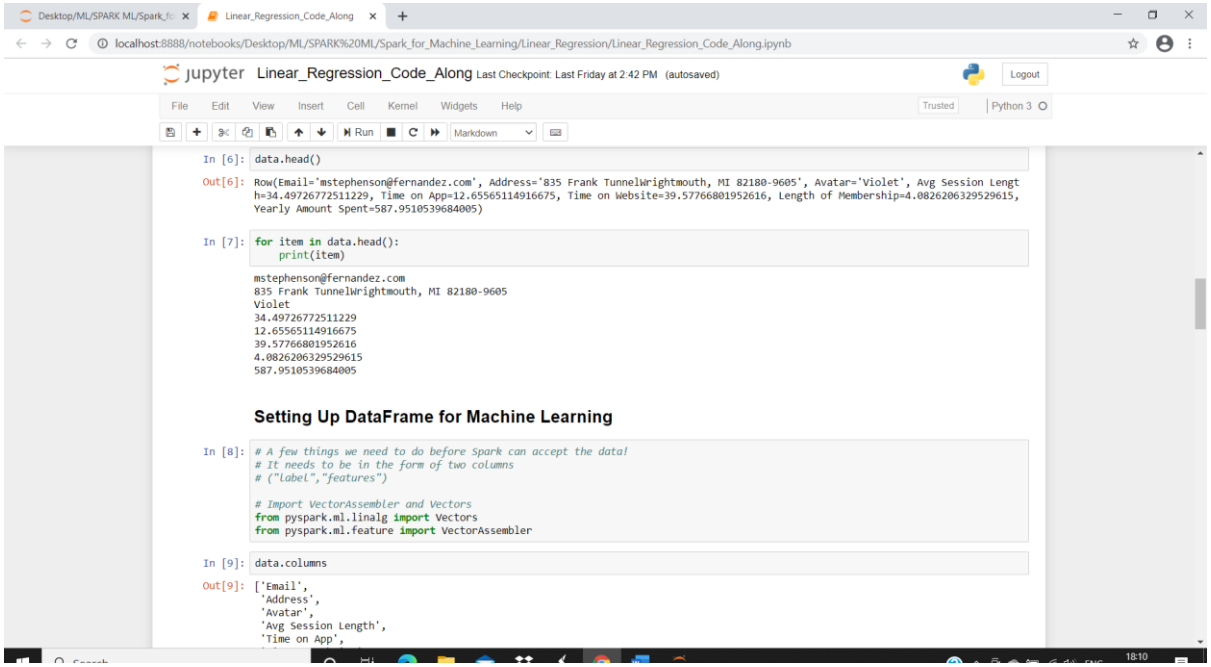


Fig 2

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

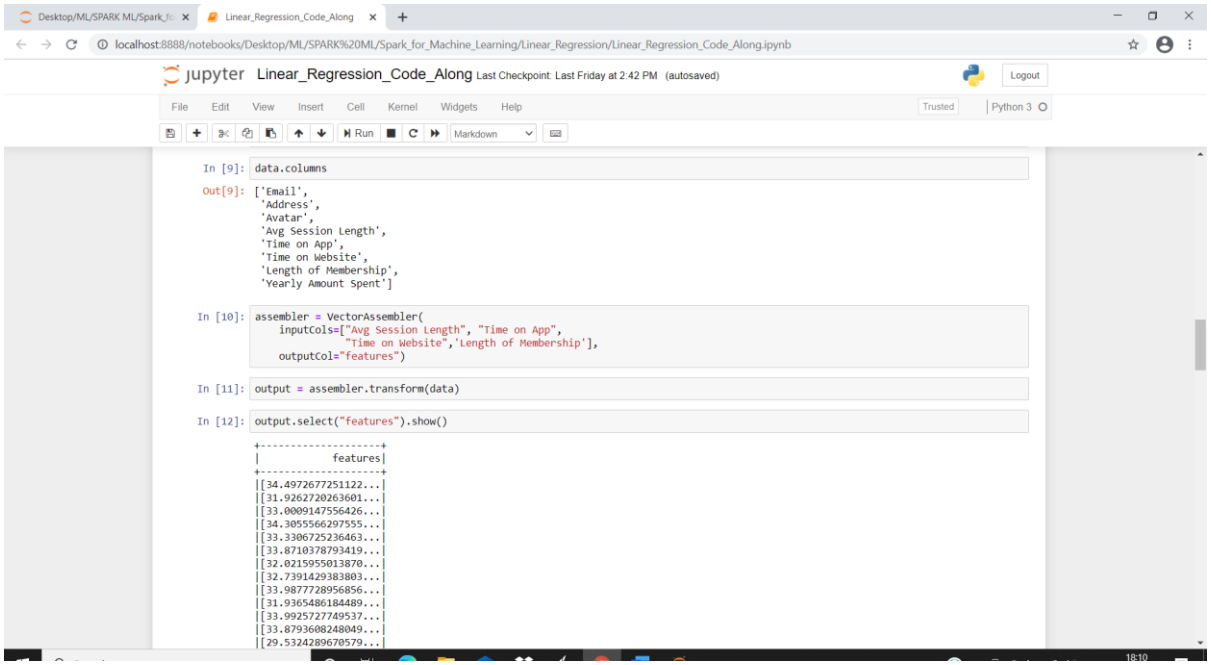


Fig 3

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

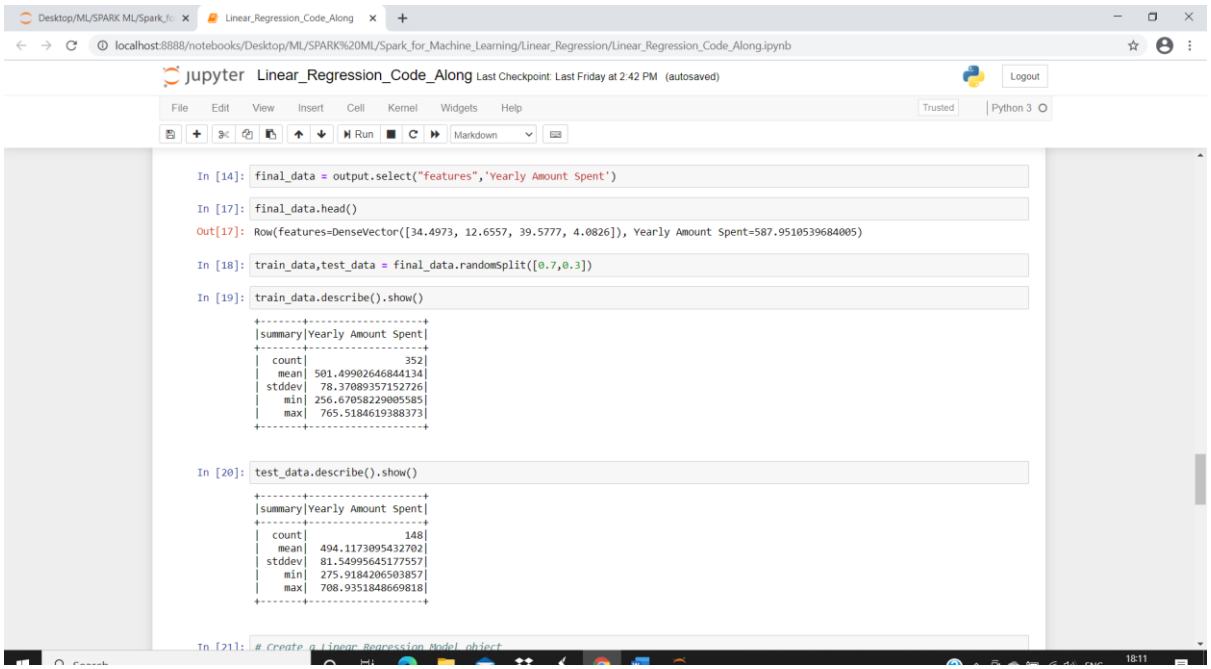


Fig 4

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.

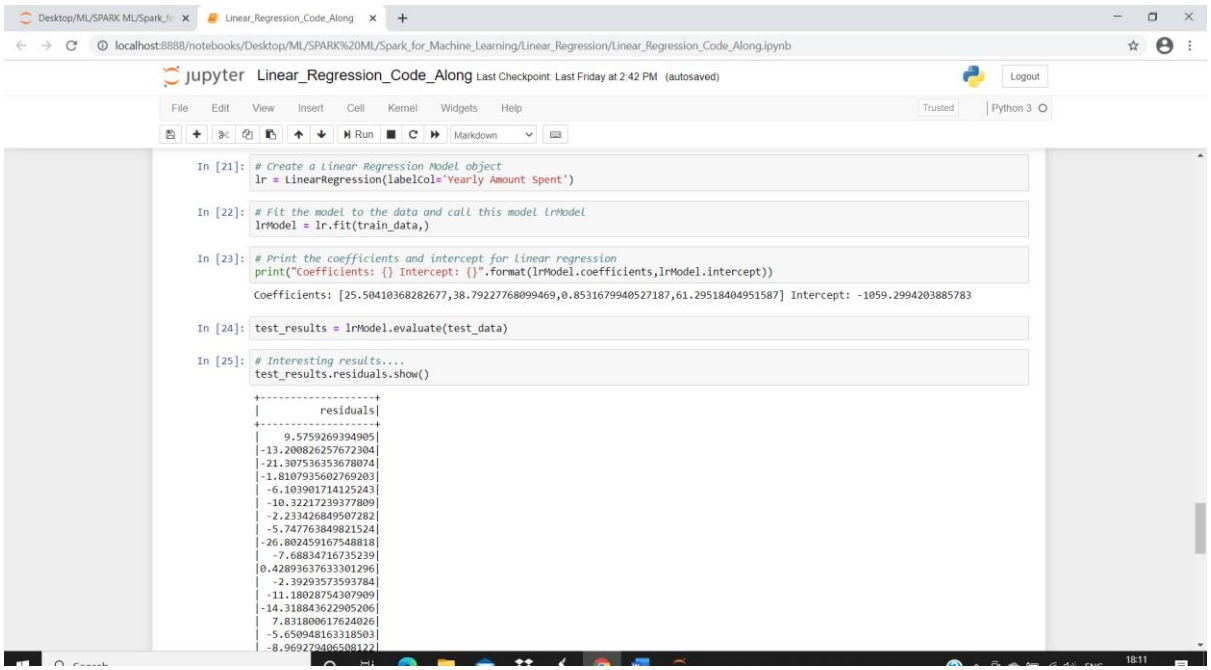


Fig 5

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.

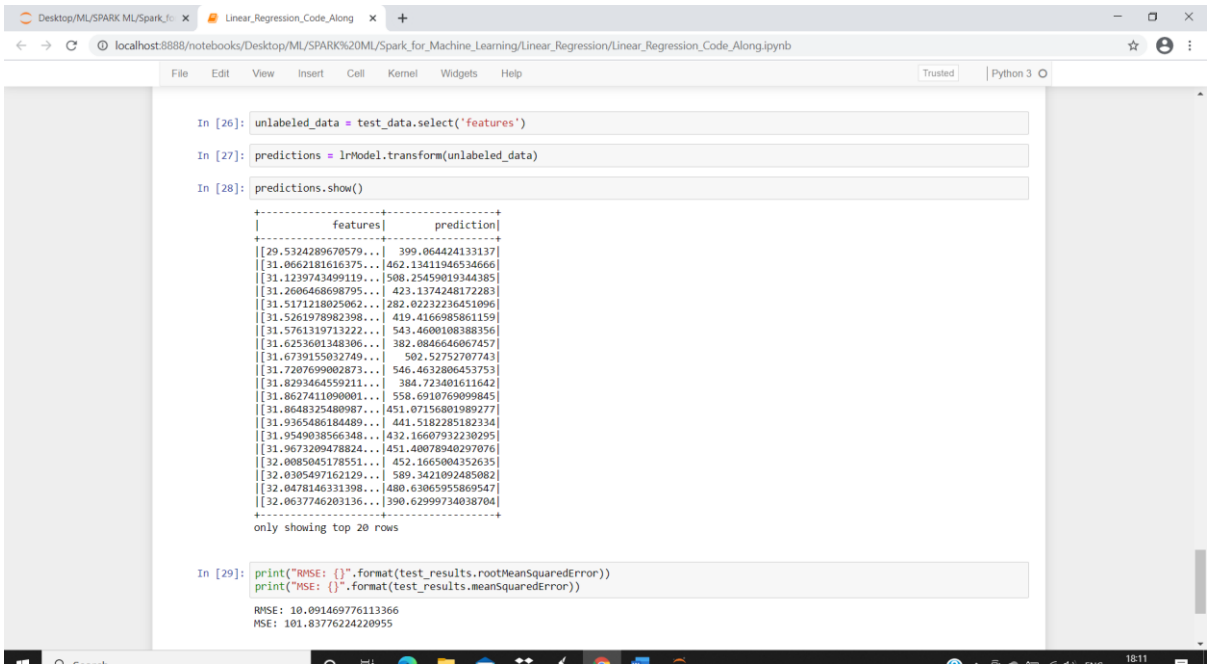


Fig 6