

Background for assignment 4

Cloud & Cluster Data Management, Spring 2018

Cuong Nguyen

Modified by Niveditha Venugopal

Overview of this talk

1. Quick introduction to Google Cloud Platform solutions for Big Data
2. Walkthrough example: MapReduce word count using Hadoop

(We suggest you set up your account and try going through this example before Assignment 4 is given on 17 May)

Google Cloud Platform (GCP)

Web-based GUI that provides a suite of cloud computing services which runs on Google's infrastructure

GCP: redeem coupon

Faculty will share the URL and instructions with students (remember to use your @pdx.edu email when redeeming)

After signing up, you'll get \$50 credit 😊

GCP: basic workflow

1. go to the GCP console at <https://console.cloud.google.com>
2. create a new project OR select an existing one
3. set billing for this project *****SUPER IMPORTANT*****
4. enable relevant APIs that you want to use
5. use APIs to manage and process data

GCP: console

The screenshot shows the Google Cloud Platform console interface. At the top, the browser address bar displays `https://console.cloud.google.com/`. Below the address bar is a blue header bar with the Google Cloud Platform logo and the project name `codelab-mapreduce`. A red arrow points from the hamburger menu icon to a callout box. Another red arrow points from the project name dropdown to a callout box. A third red arrow points from the 'Billing' section to a callout box. The main content area is divided into two columns. The left column contains a 'Project info' card with the project name `codelab-mapreduce`, the Project ID `codelab-mapreduce`, and the Project Number `#195383170405`. Below this card is a link to 'Manage project settings'. The right column contains a 'Billing' card showing `$0.00` in approximate charges for the month, with a link to 'View detailed charges'.

Projects: use this to select or create new project

Main Menu: use this to access all components of GCP

Billing information: shows how much you've been charged for the service (more later)

GCP: create a new project

1. Click on the project link in GCP console



2. Make sure organization is set to "pdx.edu", then click on the "plus" button



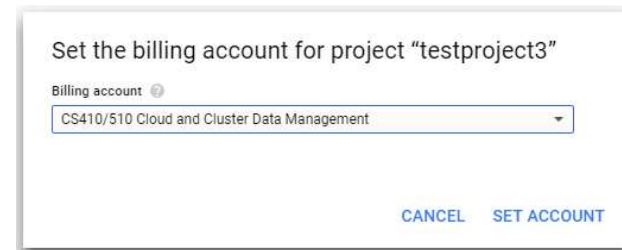
3. Give it a name, then click on "Create". Write down the Project ID, you will need it all the time



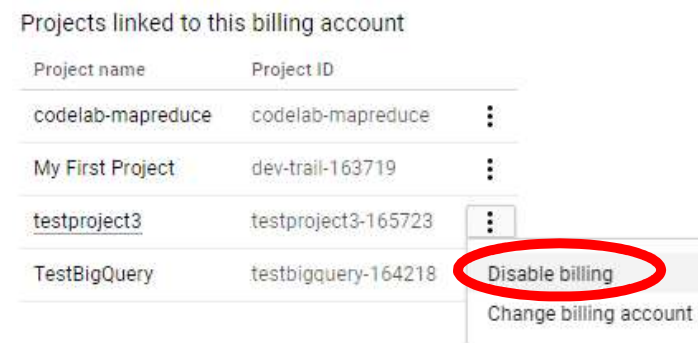
GCP: billing

Go to Main Menu → Billing

Enable billing: if a project's billing has not been enabled, set it to the billing account that you redeemed

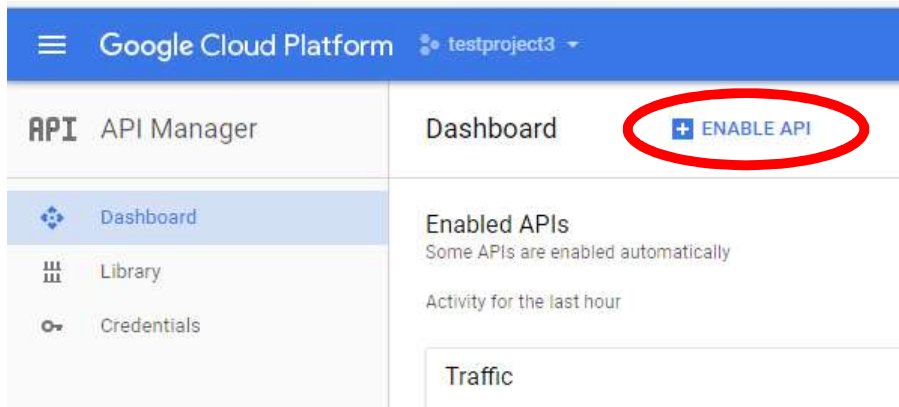


Disable billing: if you are worried about overspending, it's easiest to just disable billing from any project that you don't use



GCP: enable APIs

Go to Main Menu → API Manager, then click on Enable API



Type the name of the API you want to enable in the search box, and click Enable

GCP: learn more

check out the codelabs for GCP

<https://codelabs.developers.google.com/?cat=Cloud>

useful codelabs:

[query the Wikipedia dataset in BigQuery](#)

[introduction to Cloud Dataproc](#)

Instructions for monitoring billing

https://docs.google.com/document/d/1pkMx12gc3uSOdp4nKtTx_DJjY5SlnokwVeN8-D1gTHA/edit

MapReduce word count example

Word count

We will count the frequency of all words in the Shakespeare dataset

The data set is publicly available on Big Query

<https://bigquery.cloud.google.com/table/bigquery-public-data:samples.shakespeare>

APIs

For our word count example, we'll need these APIs enabled:

- BigQuery: query data and store results using SQL
- Google Cloud Dataproc: use Hadoop to run the MapReduce job
- Google Cloud Storage: upload and manage your own data

BigQuery and Cloud Storage are enabled by default. To enable Dataproc, you need to enable the **Google Compute Engine API** first

Basic workflow

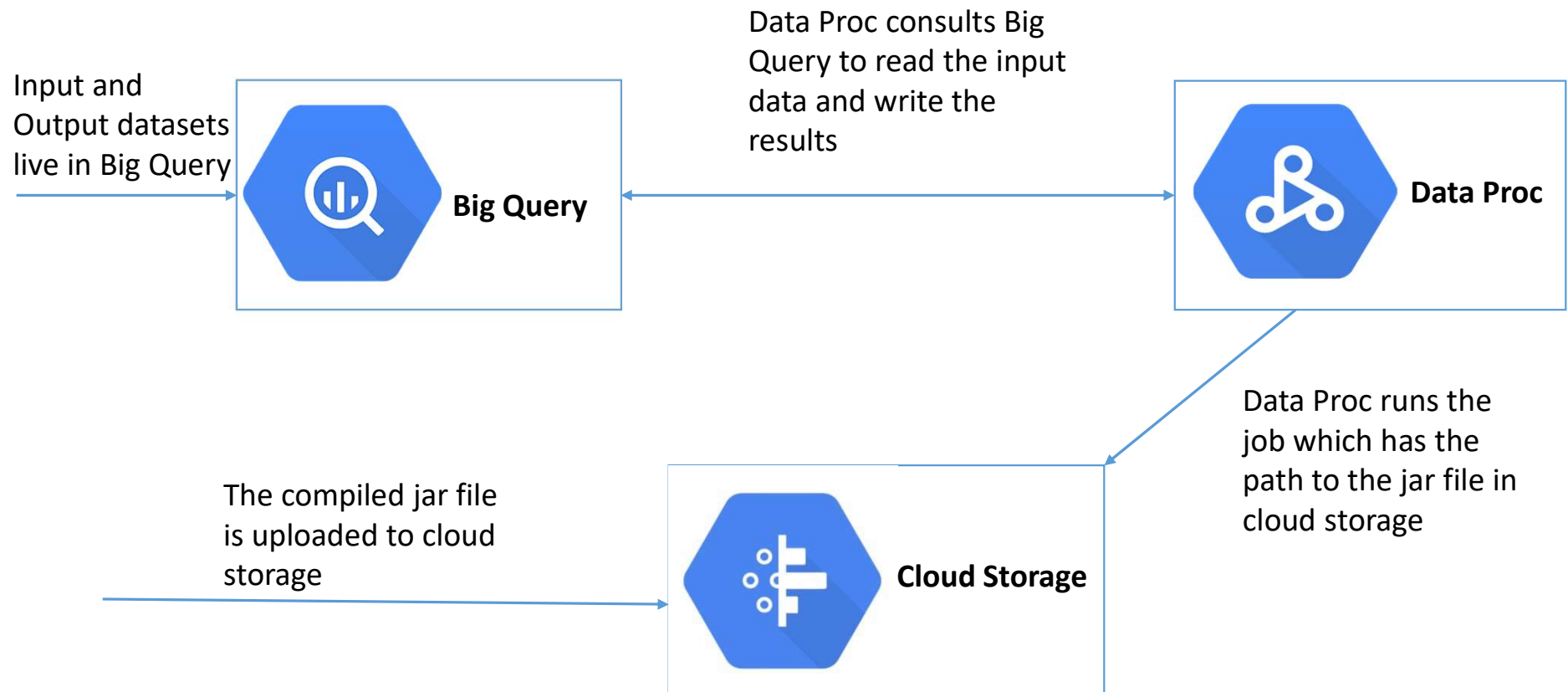
1. Big Query

1. quick start
2. create a table for the output

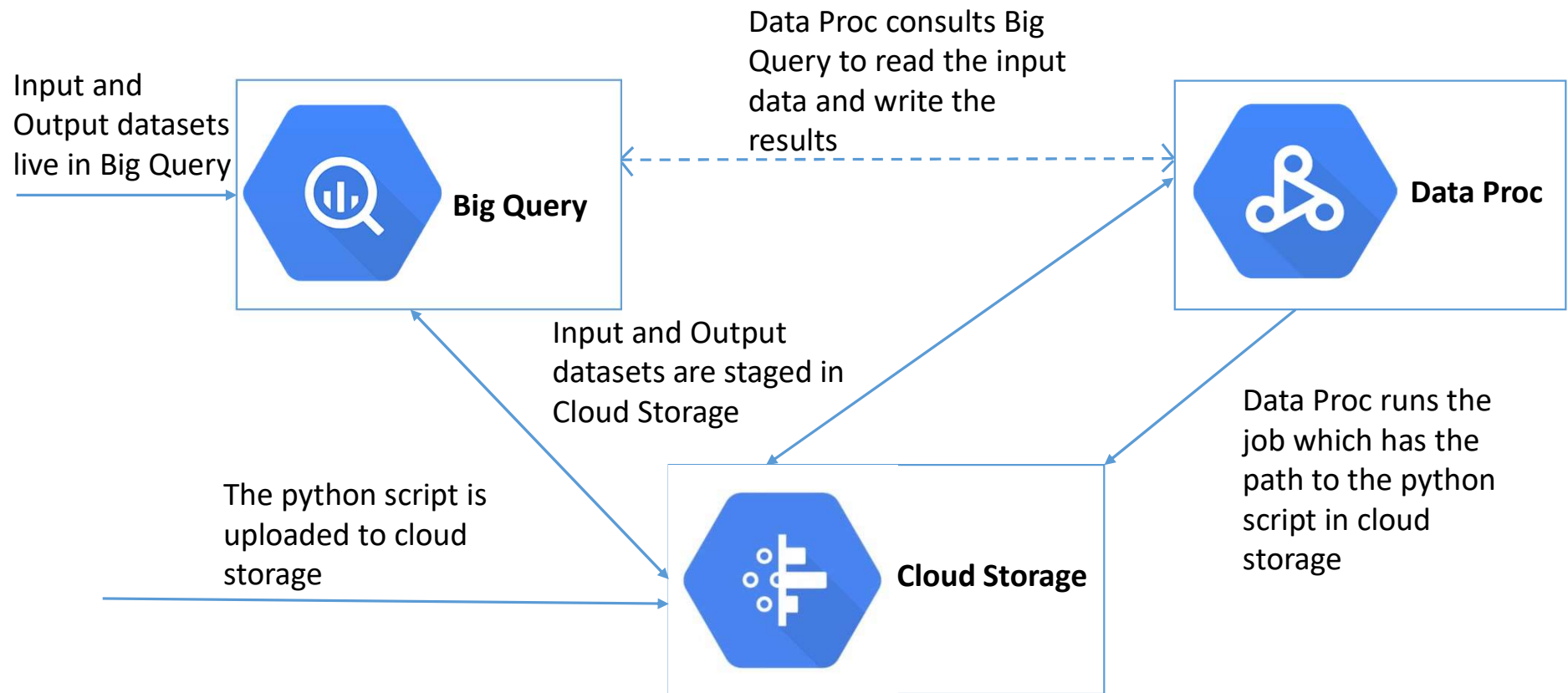
2. Dataproc

1. create clusters to run the MapReduce job using Hadoop
2. write WordCount.java to perform the MapReduce job
3. compile into a .jar file and upload it to Cloud Storage
4. submit a job to the clusters and wait
5. check the results in Big Query

Word count – API Overview Diagram for WordCount.java



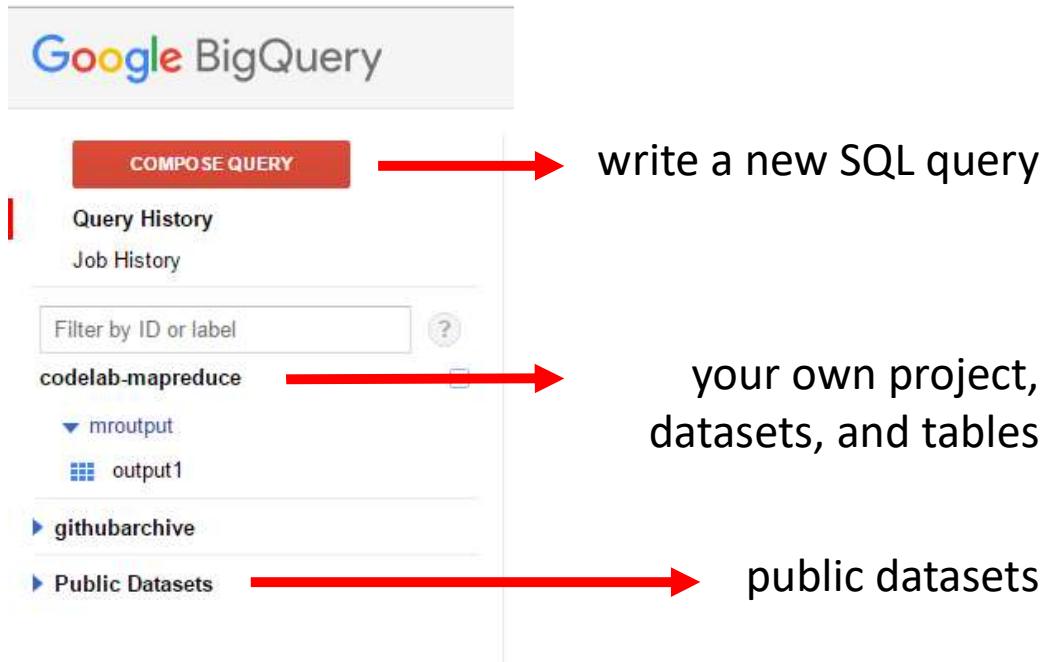
Word count – API Overview Diagram for PySpark Example



Big Query

Big Query: quick start

Go to Main Menu → BigQuery, or <https://bigquery.cloud.google.com/>



The screenshot shows the Google BigQuery interface. At the top is the 'Google BigQuery' header. Below it is a red button labeled 'COMPOSE QUERY'. To the left of this button is a sidebar menu with 'Query History' and 'Job History'. Below the menu is a search bar labeled 'Filter by ID or label'. Under the search bar, there are two sections: 'codelab-mapreduce' and 'githubarchive'. Under 'codelab-mapreduce', there is a dropdown menu showing 'mroutput' and 'output1'. Under 'githubarchive', there is a link to 'Public Datasets'. Three red arrows point from the interface to text labels: one from the 'COMPOSE QUERY' button to 'write a new SQL query', one from the 'codelab-mapreduce' section to 'your own project, datasets, and tables', and one from the 'Public Datasets' link to 'public datasets'.

COMPOSE QUERY → write a new SQL query

Query History
Job History

Filter by ID or label

codelab-mapreduce → your own project, datasets, and tables

mroutput
output1

githubarchive

Public Datasets → public datasets

Big Query: compose query

A table can be queried with the format: **[project id]:[dataset name].[table name]**

The screenshot displays the Google Cloud BigQuery console interface. It features two 'New Query' editors. The top editor contains the SQL query: `select Word, Count from [code-lab-mapreduce:mroutput.output1] order by Count desc`. The bottom editor contains the SQL query: `select word, corpus from [bigquery-public-data:samples.shakespeare]`. Below the bottom editor is a red 'RUN QUERY' button. To the right of the editors is a results table with three tabs: 'Results', 'Explanation', and 'Job Information'. The 'Results' tab is active, showing a table with columns 'Row', 'word', and 'corpus'. The table contains five rows of data from Shakespeare's sonnets. At the bottom of the interface are buttons for 'RUN QUERY', 'Save Query', 'Save View', and 'Format Query'.

New Query ?

```
1 select Word, Count from [code-lab-mapreduce:mroutput.output1] order by Count desc
```

New Query ?

```
1 select word, corpus from [bigquery-public-data:samples.shakespeare]
```

RUN QUERY

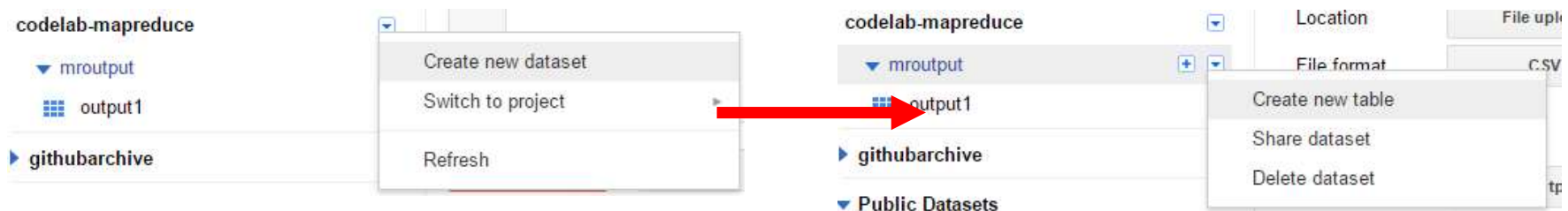
RUN QUERY Save Query Save View Format Query

Row	word	corpus
1	LVII	sonnets
2	augurs	sonnets
3	dimin'd	sonnets
4	plagues	sonnets
5	treason	sonnets

Big Query: create a new table

We will create a new table to store the results of the word count job

1. In your project, create a new dataset and a new table if you haven't had one



Big Query: create a new table

Create an empty table with two fields: Word (type string) and Count (type Integer).
Our MapReduce job will write the results into this table

Create Table

Source Data ☐ Create from source ☒ Create empty table

Destination Table

Table name . ?
Table type ?

Schema

Name	Type	Mode
Word	STRING	NULLABLE X
Count	INTEGER	NULLABLE X

[Edit as Text](#)

Options

Partitioning

Dataprocc

Dataproc: create a new cluster

Go to Main Menu → Dataproc, or
<https://console.cloud.google.com/dataproc/>, then click on “Clusters”

Click on “Create Cluster”

Settings on next slide...

Dataproc: create a new cluster

We will create one master node and three worker nodes

Name ?

cluster-8293

Region ?

us-west1

Zone ?

us-west1-a

Cluster mode ?

Standard (1 master, N workers)

Master node

Contains the YARN Resource Manager, HDFS NameNode, and all job drivers

Machine type ?

4 vCPUs

15 GB memory

[Customize](#)

Primary disk size (minimum 10 GB) ?

32

GB

Worker nodes

Each contains a YARN NodeManager and a HDFS DataNode.
The HDFS replication factor is 2.

Machine type ?

4 vCPUs

15 GB memory

[Customize](#)

Primary disk size (minimum 10 GB) ?

32

GB

Nodes (minimum 2) ?

3

Local SSDs (0-8) ?

0

x 375 GB

YARN cores ?

12

YARN memory ?

36 GB

Dataproc: important

Dataproc is billed **by the minute**

<https://cloud.google.com/dataproc/docs/resources/pricing>

It'll be fine for our example, but if you don't use it: **delete the cluster**

MapReduce word count in Java

Source code is on github

<https://github.com/cuong3/GoogleCloudPlatformExamples/blob/master/dataproc-mapreduce/WordCount.java>

Input arguments

- arg0 projectId = your project id (example: mine is *codelab-mapreduce*)
- arg1 fullyQualifiedInputTableId = bigquery-public-data:samples.shakespeare
- arg2 fieldName = word
- arg3 fullyQualifiedOutputTableId = output table (example: mine is *codelab-mapreduce:mroutput.output1*)

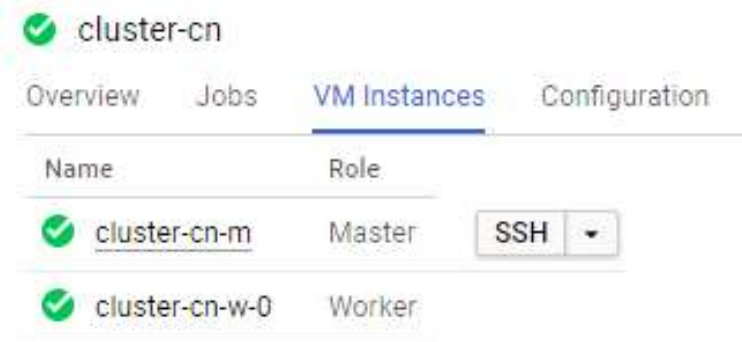
Make .jar file for word count

Go to Main Menu → Dataproc → Clusters

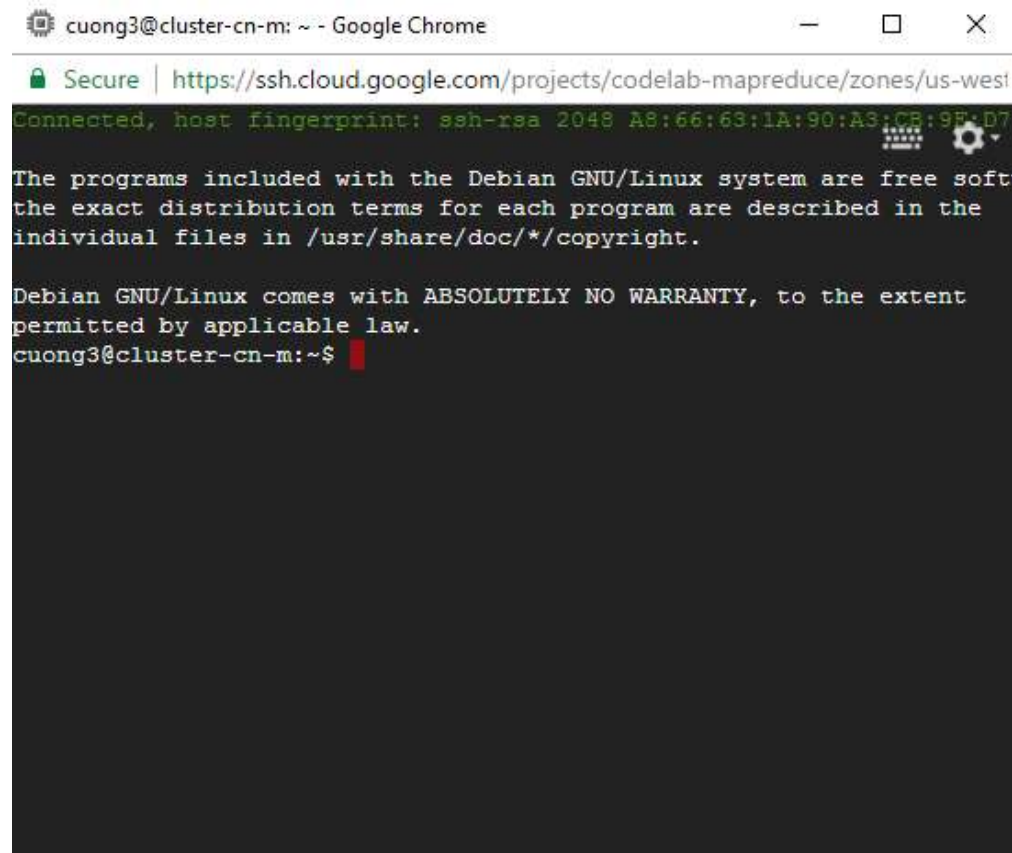
Click on your cluster name

Choose “VM Instances”

Click on SSH. This will open a command line interface that lets you access your cluster.



Make .jar file for word count



```
cuong3@cluster-cn-m: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/codelab-mapreduce/zones/us-west
Connected, host fingerprint: ssh-rsa 2048 A8:66:63:1A:90:A3:CB:9F:D7
The programs included with the Debian GNU/Linux system are free soft
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
cuong3@cluster-cn-m:~$
```

Make .jar file for word count

Setup environment variables (needs to be done every time you open SSH)

Create a home directory

```
hadoop fs -mkdir -p /user/your user name
```

To find your user name, run

```
whoami
```

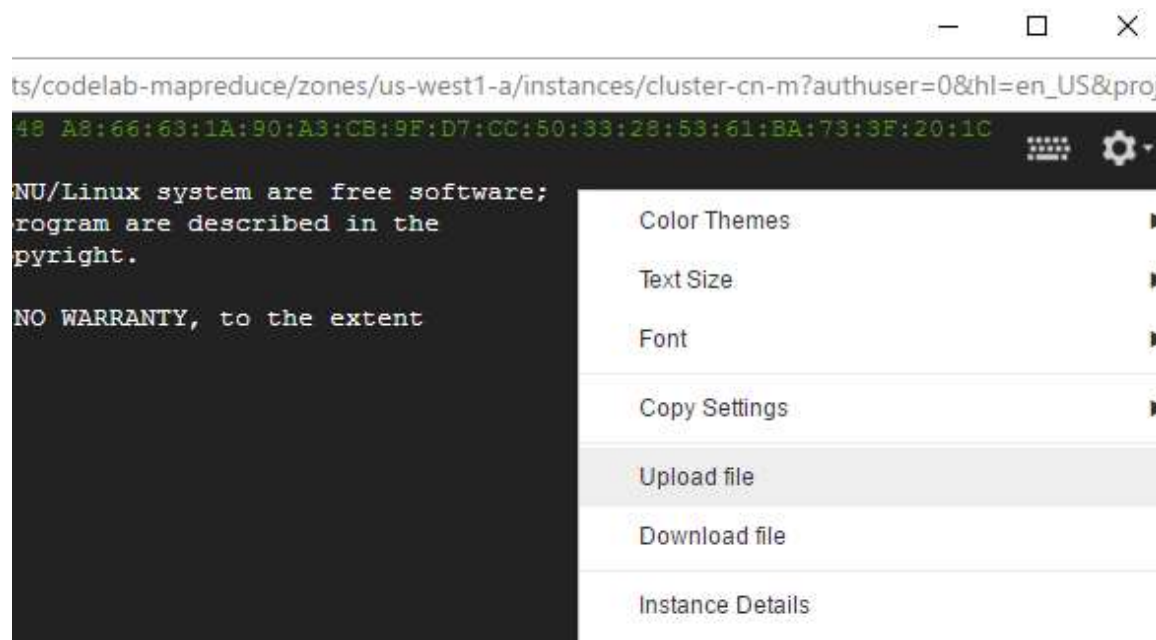
Setup paths

```
export PATH=${JAVA_HOME}/bin:${PATH}
```

```
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```

Make .jar file for word count

Upload your .java file to the cluster: click on the gear icon in the top right corner and selects “Upload file”



Make .jar file for word count

Compile

```
hadoop com.sun.tools.javac.Main ./WordCount.java  
jar cf wordcount.jar WordCount*.class
```

Check if wordcount.jar has been created

```
ls
```

We now have the wordcount.jar in the cluster, we need to move it to a Google Cloud Storage folder so we can submit a MapReduce job with it later

Why store the .jar file in Cloud Storage?

You still have access to the files when you shut down the cluster

You can transfer/access files in Cloud Storage in almost all of GCP's APIs

Cloud Storage

Go to Main Menu → Dataproc → Clusters, click on “Cloud Storage staging bucket”

Search clusters, press Ent

Name ^	Zone	Total worker nodes	Cloud Storage staging bucket	Created	Status
cluster-cn	us-west1-a	3	dataproc-9b19685e-1912-48fa-afab-651b8f92355b-us	Apr 26, 2017, 10:43:29 AM	Running

In the Browser tab, Click on “Create Folder”, give it a name.

Example: if a create a folder called “testMapReduce”, then my Cloud Storage link would be `gs://dataproc-9b19685e-1912-48fa-afab-651b8f92355b-us/testMapReduce/`

Transfer wordcount.jar to Cloud Storage

Back to the SSH interface of our cluster, run these commands:

Copy wordcount.jar to Hadoop file system

```
hadoop fs -copyFromLocal ./wordcount.jar
```

Copy wordcount.jar to folder testMapReduce in Cloud Storage

```
hadoop fs -cp ./wordcount.jar gs://dataproc-9b19685e-1912-48fa-afab-651b8f92355b-us/testMapReduce/
```

[Buckets](#) / [dataproc-9b19685e-1912-48fa-afab-651b8f92355b-us](#) / testMapReduce

☐	Name	Size	Type
☐	 wordcount.jar	4.73 KB	application/octet-stream

Submit the MapReduce job

Go to Main Menu → Dataproc → Jobs

Click on Submit Job

Set Job type to: Hadoop

Set Jar files to your Cloud Storage link (example: my link is `gs://dataproc-9b19685e-1912-48fa-afab-651b8f92355b-us/testMapReduce/wordcount.jar`)

Set Main class of jar to: WordCount

Set arguments (see slide 26)

Click Submit

It should take about 2~10 minutes to finish the job

☐		a423a58b-a8ef-4f61-b140-b8f281c68159	Hadoop	cluster-cn	Apr 25, 2017, 2:45:31 PM	2 min 49 sec	Succeeded
--------------------------	---	--------------------------------------	--------	------------	--------------------------	--------------	-----------

How do we run it?

Go to Main Menu → Dataproc → Jobs, click on Submit a job

← Submit a job

Cluster
cluster-cn

Job type
Hadoop

Jar files (Optional) ?
Enter file path, for example, hdfs://example/example.jar

Main class or jar ?
Enter class name or select a jar file
Main class or jar is required

Arguments (Optional) ?

codelab-mapreduce	×
bigquery-public-data:samples.shakespeare	×
word	×
codelab-mapreduce:mroutput.output1	×
Press <Return> to add more arguments	

Properties (Optional) ?
+ Add item

Labels (Optional) ?
+ Add item

Submit Cancel

To run the code on our clusters, we will need to compile it into a .jar file first

Check the results in BigQuery

New Query ?

```
1 select Word, Count from [code1ab-mapreduce:mroutput.output1] order by Count desc
```

RUN QUERY

Save Query

Save View

Format Query

Show Options

Query complete (0.8s ela

Results			Explanation	Job Information
Row	Word	Count		
1	in	42		
2	down	42		
3	as	42		
4	again	42		
5	How	42		

Note: if you run the MapReduce job one more time, the results will be appended to this table

Important

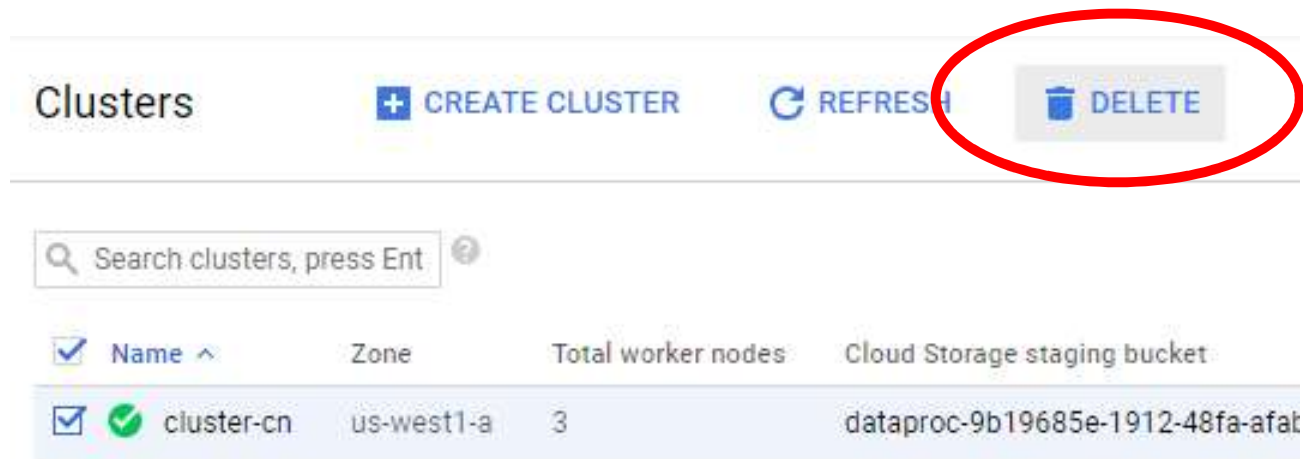
You can choose to stop your clusters if you want to get back to your work after some time

You would be charged much lesser if you do this compared to leaving your clusters running

VM instances						
CREATE INSTANCE ▼ IMPORT VM REFRESH START STOP RESET DELETE						
☐	Name ^	Zone	Recommendation	Internal IP	External IP	Connect
☐	cluster-7361-m	us-west1-a		10.138.0.4	None	SSH ▼ ⋮
☐	cluster-7361-w-0	us-west1-a		10.138.0.2	None	SSH ▼ ⋮
☐	cluster-7361-w-1	us-west1-a		10.138.0.5	None	SSH ▼ ⋮
☐	cluster-7361-w-2	us-west1-a		10.138.0.3	None	SSH ▼ ⋮

Important

Don't forget to delete your clusters when you're done



The screenshot shows the Google Cloud Clusters management interface. At the top, there are four buttons: 'Clusters' (text), '+ CREATE CLUSTER' (blue icon and text), 'REFRESH' (blue circular arrow icon and text), and 'DELETE' (blue trash can icon and text). The 'DELETE' button is circled in red. Below the buttons is a search bar with the placeholder text 'Search clusters, press Ent'. Below the search bar is a table with the following columns: 'Name', 'Zone', 'Total worker nodes', and 'Cloud Storage staging bucket'. The table contains one row with the following data: 'cluster-cn' (with a green checkmark icon), 'us-west1-a', '3', and 'dataproc-9b19685e-1912-48fa-afat'.

Name	Zone	Total worker nodes	Cloud Storage staging bucket
☒ cluster-cn	us-west1-a	3	dataproc-9b19685e-1912-48fa-afat

