

Background for assignment 4

Cloud & Cluster Data Management, Spring 2017

Cuong Nguyen

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 18 May)

Google Cloud Platform (GCP)

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, there's a blue header bar with the Google Cloud Platform logo and the project name 'codelab-mapreduce'. Below this, there's a navigation bar with 'DASHBOARD' and 'ACTIVITY' tabs. The main content area displays 'Project info' for 'codelab-mapreduce' with its Project ID and a link to 'Manage project settings'. On the right, there's a 'Billing' section showing '\$0.00' and a link to 'View detailed charges'. Three red arrows point from text boxes to specific UI elements: one to the main menu icon, one to the project name dropdown, and one to the 'View detailed charges' link.

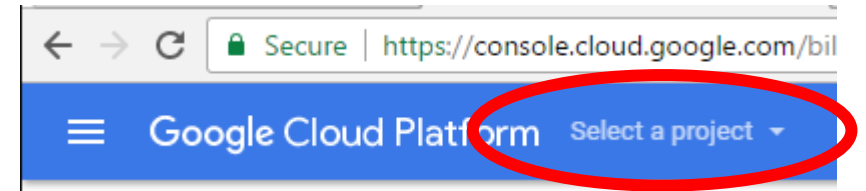
Projects: use this to select or create new project

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

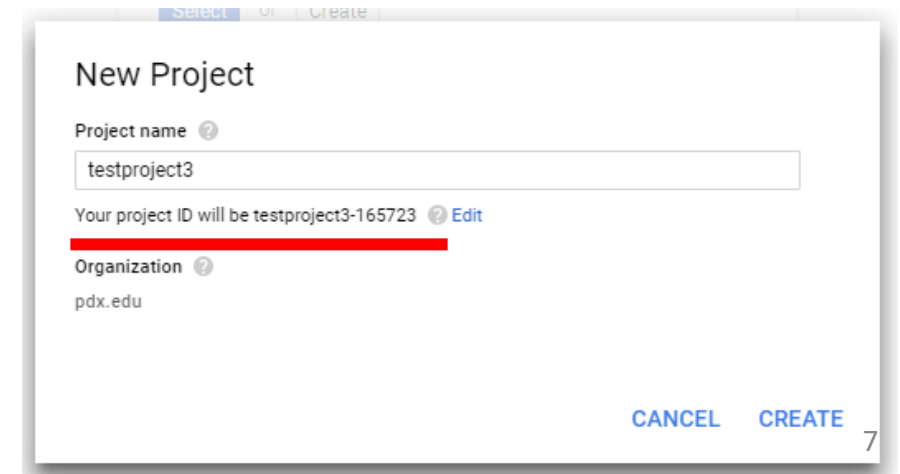
1. Click on the project link in GCP console



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



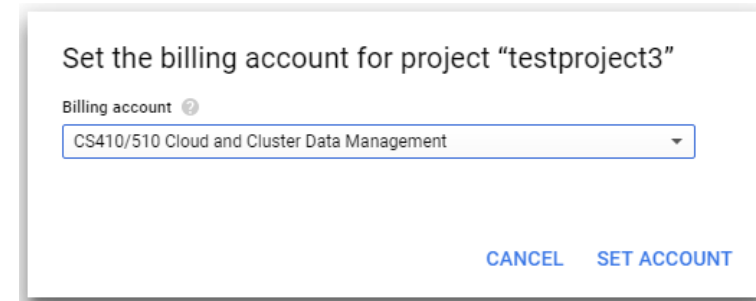
3. Give it a name, then click on “Create”. Write down the Project ID, you will need it all the time



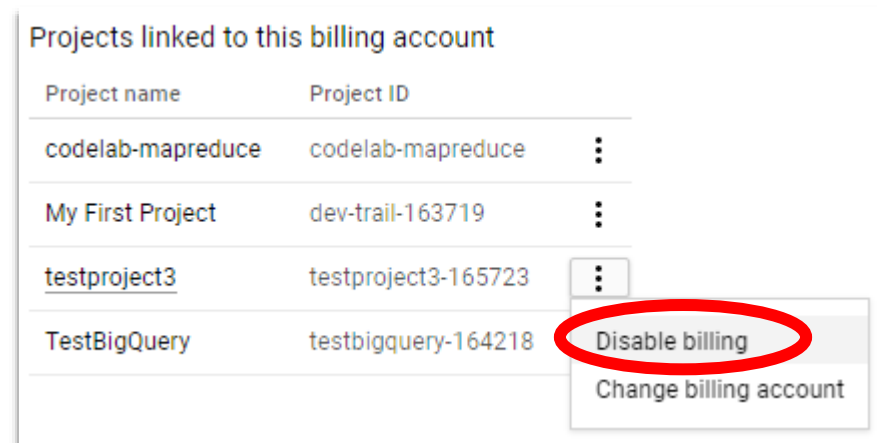
GCP: billing

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

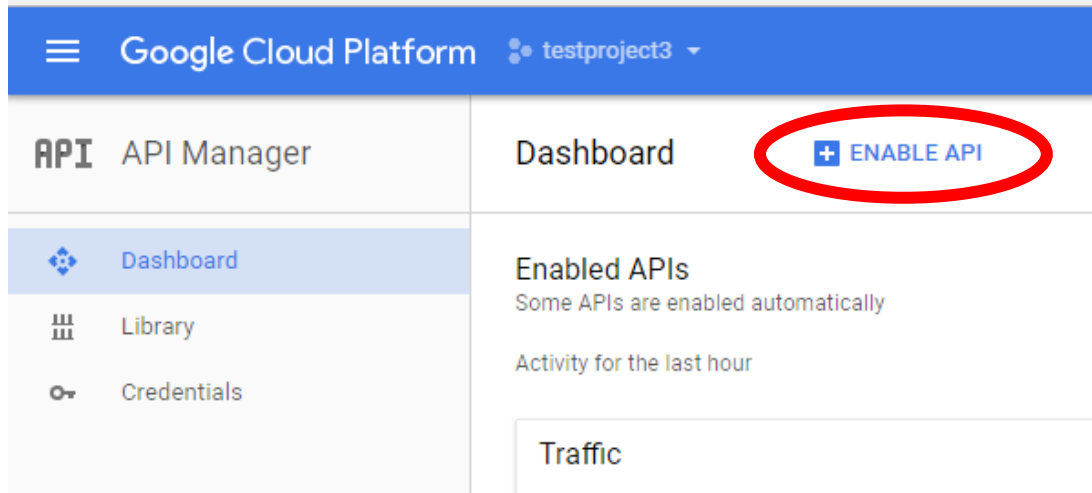


Disable billing: if you are worried about overspending, it's easiest to just disable billing from any project that you don't use



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

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 the main content area is a sidebar menu with 'Query History' and 'Job History'. Below these is a search bar labeled 'Filter by ID or label'. Under the search bar, there are several project and dataset entries: 'codelab-mapreduce', 'mroutput', 'output1', 'githubarchive', and 'Public Datasets'. Red arrows point from these elements to explanatory text on the right: 'COMPOSE QUERY' points to 'write a new SQL query', 'codelab-mapreduce' points to 'your own project, datasets, and tables', and 'Public Datasets' points 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]**

New Query ?

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

RUN QUERY

New Query ?

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

RUN QUERY Save Query Save View Format Query

Results				Explanation	Job Information
Row	word	corpus			
1	LVII	sonnets			
2	augurs	sonnets			
3	dim'm'd	sonnets			1,
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 mroutput . mytablename ?
Table type Native table ?

Schema

Name	Type	Mode
<input type="text" value="Word"/>	STRING	NULLABLE ×
<input type="text" value="Count"/>	INTEGER	NULLABLE ×

[Add Field](#)

[Edit as Text](#)

Options

Partitioning None

[Create Table](#)

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-1

Zone ?

us-west1-a

Master node

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

Machine type ?

n1-standard-4 (4 vCPU, 15.0 GB ...

Cluster mode ?

Standard (1 master, N workers)

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 ?

n1-standard-4 (4 vCPU, 15.0 GB ...

Nodes (minimum 2) ?

3

Primary disk size (minimum 10 GB) ?

32 GB

Local SSDs (0-8) ?

0 x 375 GB

YARN cores ?

12

YARN memory ?

36.0 GB

Preemptible workers, bucket, network, version, initialization, & access options

Create

Cancel

22

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*)

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

Make .jar file for word count

Go to Main Menu → Dataproc → Clusters

Click on your cluster name

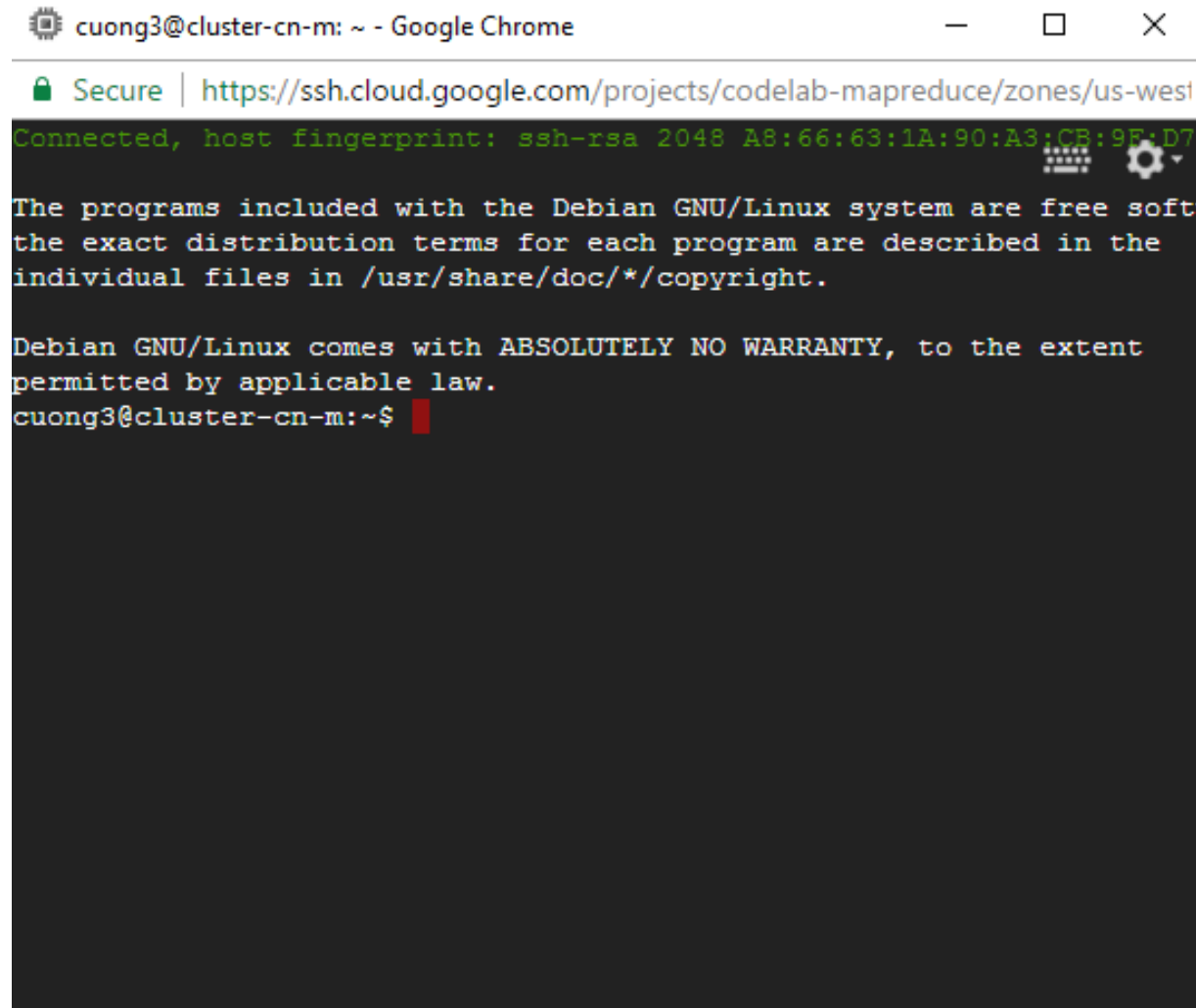
Choose “VM Instances”

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

✓ cluster-cn

Overview	Jobs	VM Instances	Configuration
Name	Role		
✓ <u>cluster-cn-m</u>	Master	SSH	▼
✓ cluster-cn-w-0	Worker		

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 software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

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

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
```

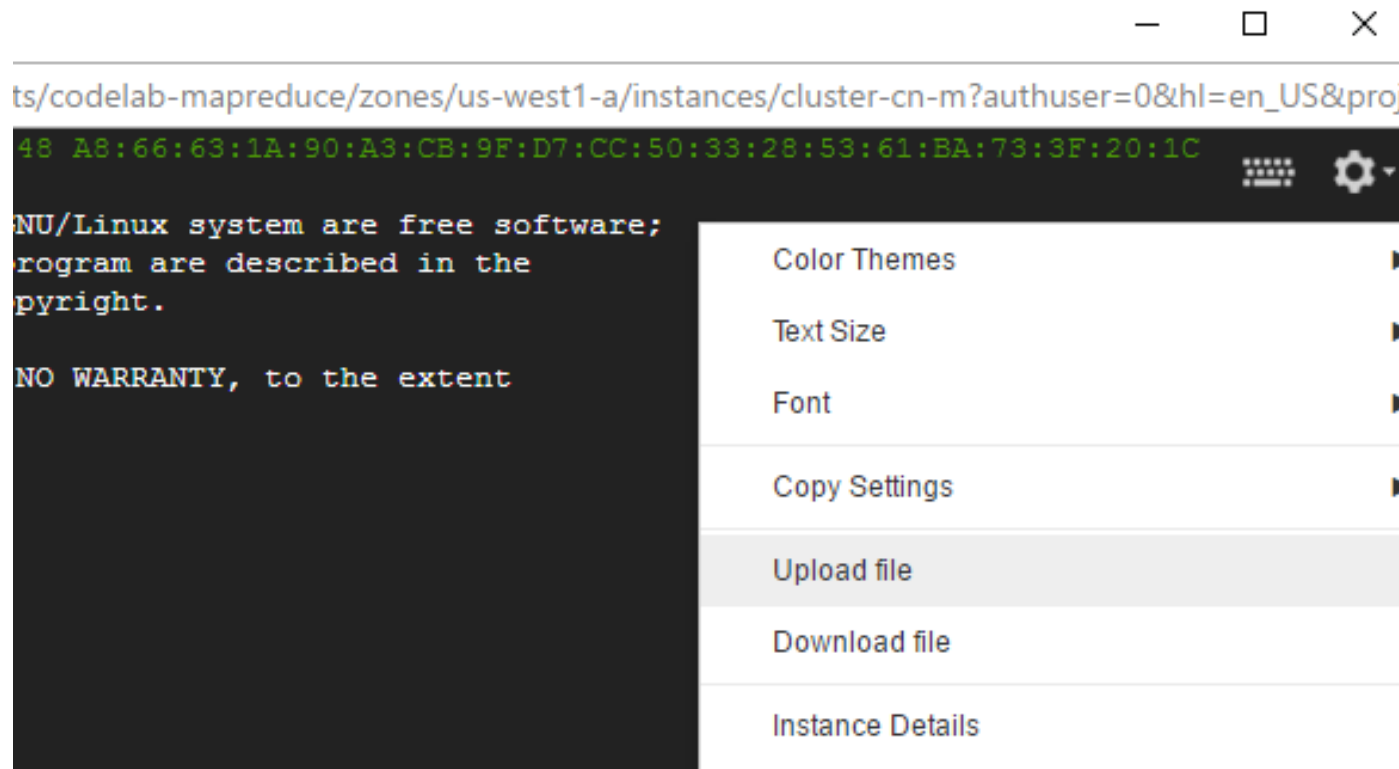
Setup paths

```
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 select “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”

<input type="text" value="Search clusters, press Ent"/>						
<input type="checkbox"/>	Name ^	Zone	Total worker nodes	Cloud Storage staging bucket	Created	Status
<input type="checkbox"/>	<input checked="" type="checkbox"/> 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

<input type="checkbox"/>	Name	Size	Type
<input type="checkbox"/>	 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 24)

Click Submit

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

<input type="checkbox"/>	<input checked="" type="checkbox"/>	a423a58b-a8ef-4f61-b140-b8f281c68159	Hadoop	cluster-cn	Apr 25, 2017, 2:45:31 PM	2 min 49 sec	Succeeded
--------------------------	-------------------------------------	--------------------------------------	--------	------------	--------------------------	--------------	-----------

Check the results in BigQuery

New Query 

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

RUN QUERY 

Save Query

Save View

Format Query

Show Options

Query complete (0.8s elap)

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

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

Clusters

+

CREATE CLUSTER

↺

REFRESH

🗑️

DELETE

🔍

Search clusters, press Ent

?

<div><div><input checked="" type="checkbox"/></div><div>Name ^</div></div>	Zone	Total worker nodes	Cloud Storage staging bucket
<div><div><input checked="" type="checkbox"/></div><div><div>✅</div>cluster-cn</div></div>	us-west1-a	3	dataproc-9b19685e-1912-48fa-afat

