Module 12

Big Data as a Service using Google Cloud Platform

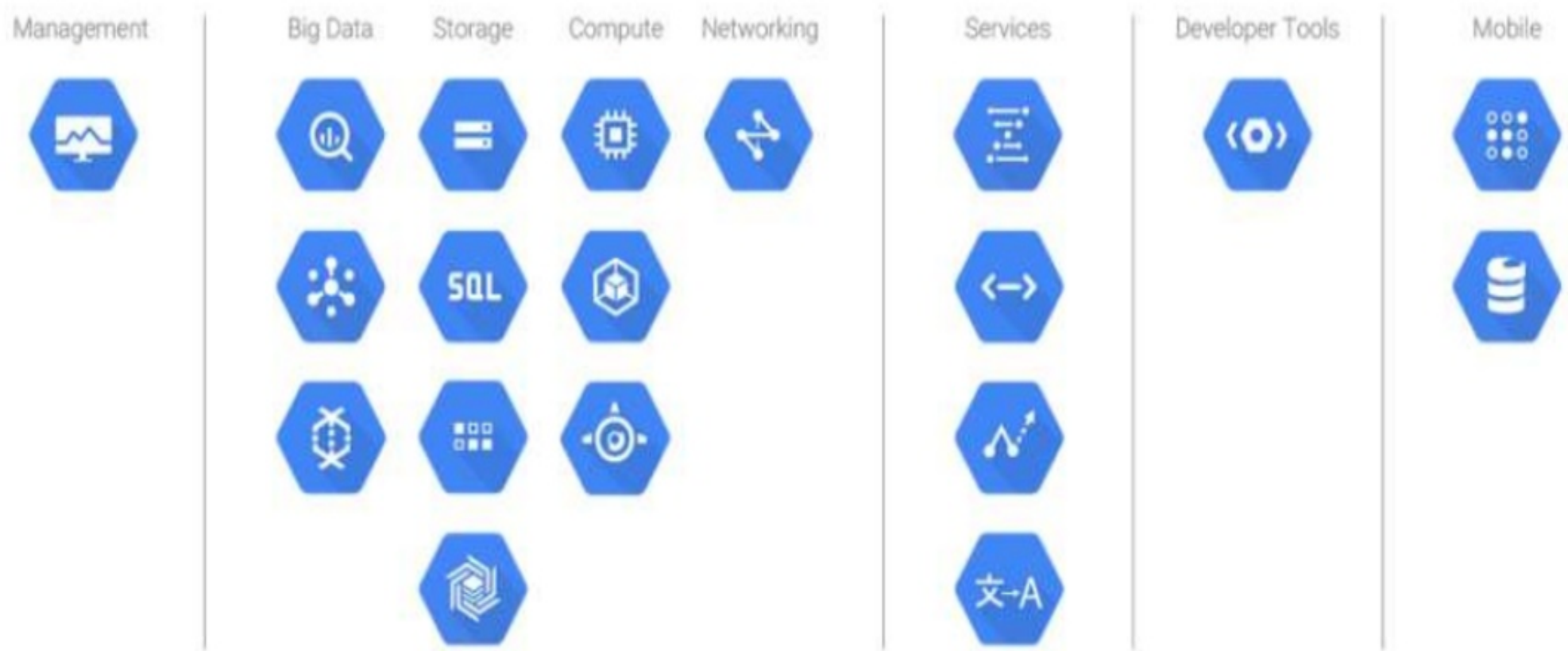
Thanachart Numnonda, Executive Director, IMC Institute

Mr.Aekanun Thongtae, Big Data Consultant, IMC Institute

Thanisa Numnonda, Faculty of Information Technology,

King Mongkut's Institute of Technology Ladkrabang

Google Cloud Platform Services



GCP Compute Services

COMPUTE ENGINE

Run large-scale workloads on virtual machines hosted on Google's infrastructure.

PREEMPTIBLE VMS

Preemptible VMs are a low cost choice for distributed and fault-tolerant workloads.

CUSTOM MACHINE TYPES

Create Compute Engine VMs with optimal amounts of vCPU and memory.

APP ENGINE

A platform for building scalable web apps and mobile backends.

CONTAINER ENGINE

Run Docker containers on Google's infrastructure, powered by Kubernetes.

CONTAINER REGISTRY

Fast, private Docker image storage on Google Cloud Platform.

GCP Storage Services

CLOUD STORAGE

Powerful, simple and cost effective object storage service with global edge-caching.

CLOUD STORAGE NEARLINE

A highly available, affordable solution for backup, archiving and disaster recovery.

CLOUD SQL

Store and manage data using a fully-managed, relational MySQL database.

CLOUD DATASTORE

A managed, NoSQL, schemaless database for storing non-relational data.

CLOUD BIGTABLE

Cloud Bigtable is a fast, fully managed, massively scalable NoSQL database service.

GCP Big Data Services

BIGQUERY

Analyze Big Data in the cloud. Run fast, SQL-like queries against petabytes of data in seconds.

CLOUD DATAFLOW

Dataflow is a real-time data processing service for batch and stream data processing.

CLOUD DATAPROC

Google Cloud Dataproc is a managed Spark and Hadoop service that is fast, easy to use, and low cost.

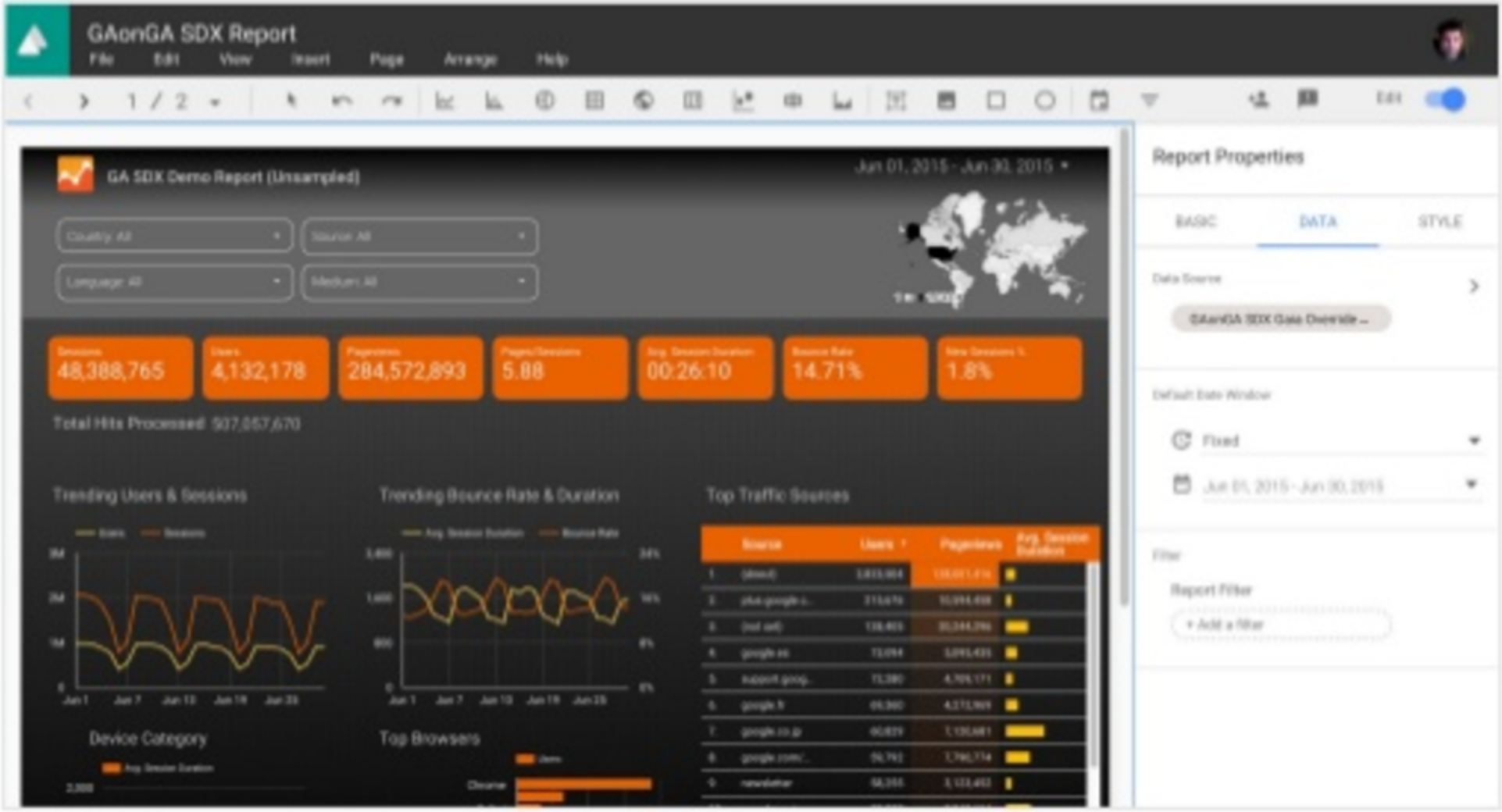
CLOUD DATALAB

An easy to use interactive tool for large-scale data exploration, analysis and visualization.

CLOUD PUB/SUB

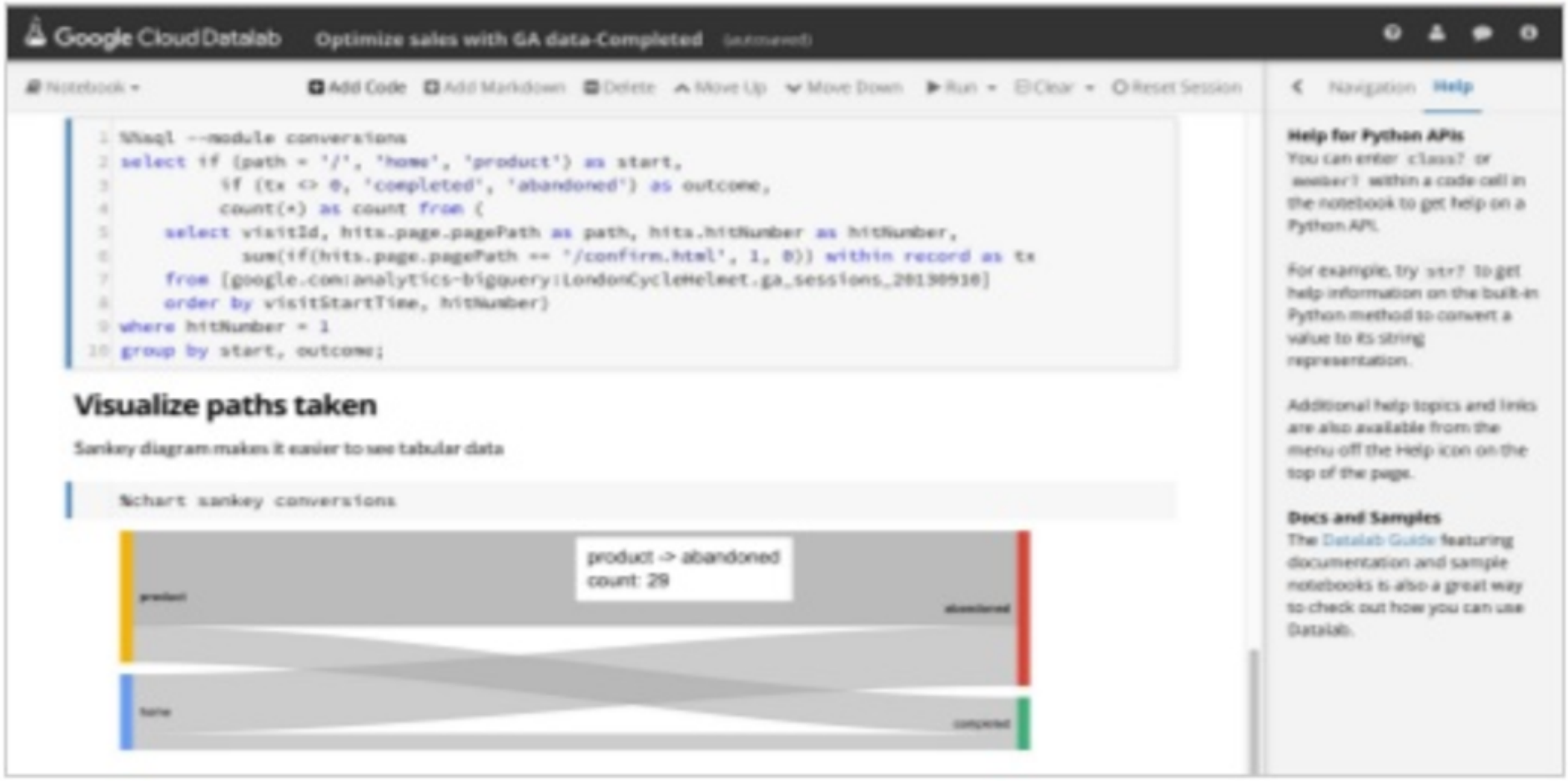
Connect your services with reliable, many-to-many, asynchronous messaging hosted on Google's infrastructure.

Google Data Studio 360



Source:

Cloud Datalab



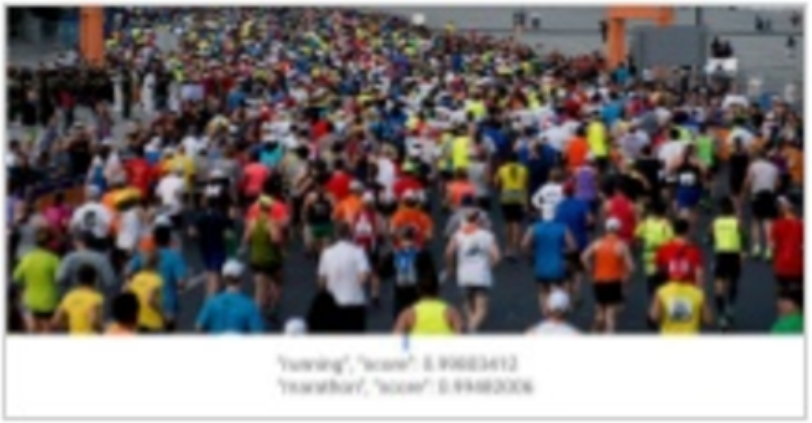
Source:

TensorFlow

- Deep Learning technology powering over 100 Google services
- Generalizable to vision, sound, text, video and other data
- Runs on CPUs or GPUs, desktop, server, or mobile computing platforms
- Available as Apache 2.0 OSS license

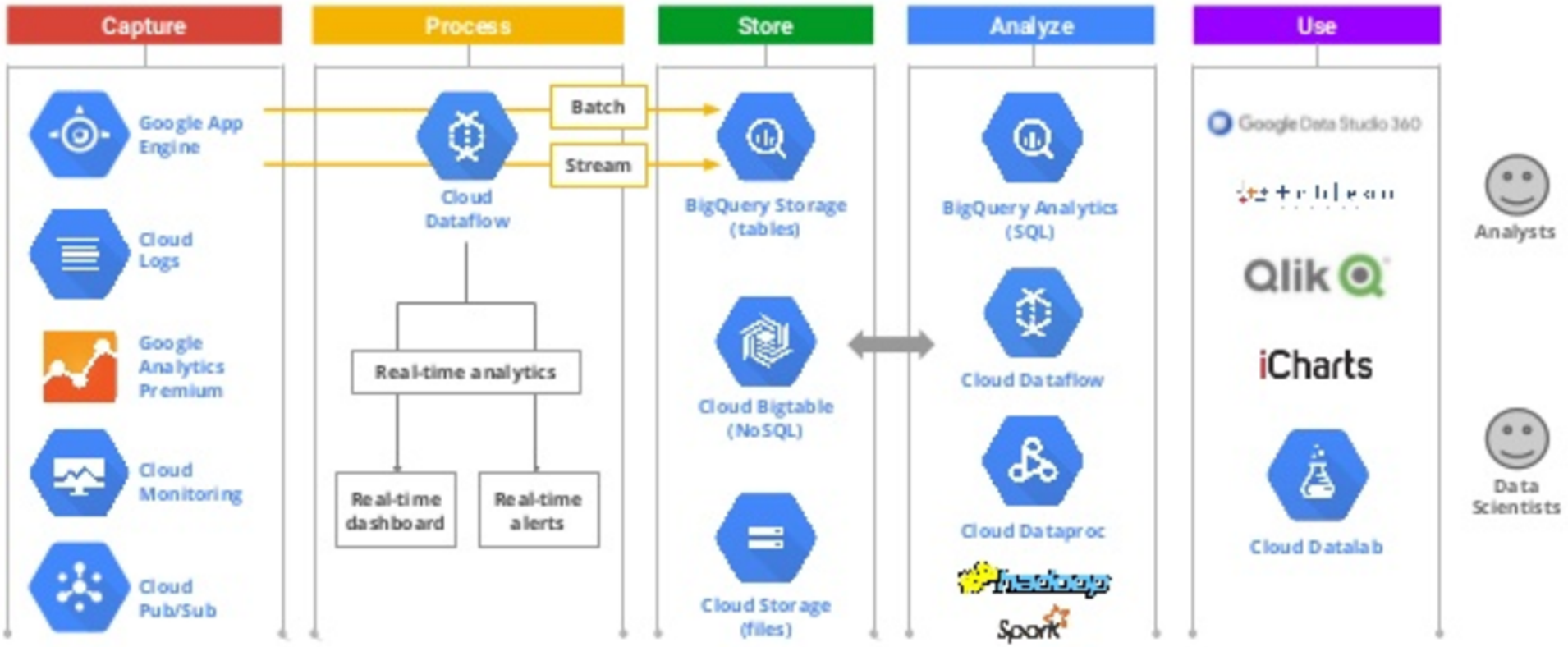
Source:

Cloud Vision API



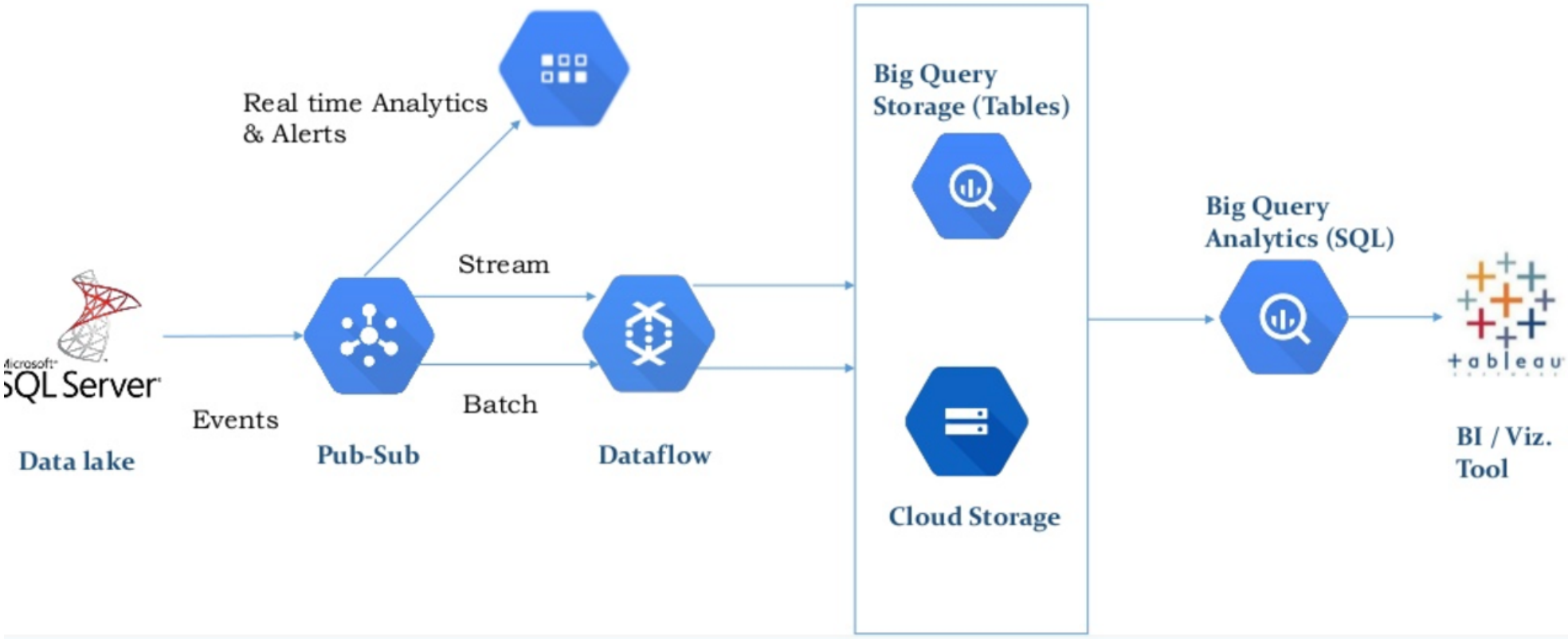
Source:

Big Data Lifecycle



Source: Google Cloud Platform & rockPlace Big Data

Big Data – Google Cloud Architecture



Source: Big Data with Google Cloud Architecture

Hadoop v.s. Public Cloud Services

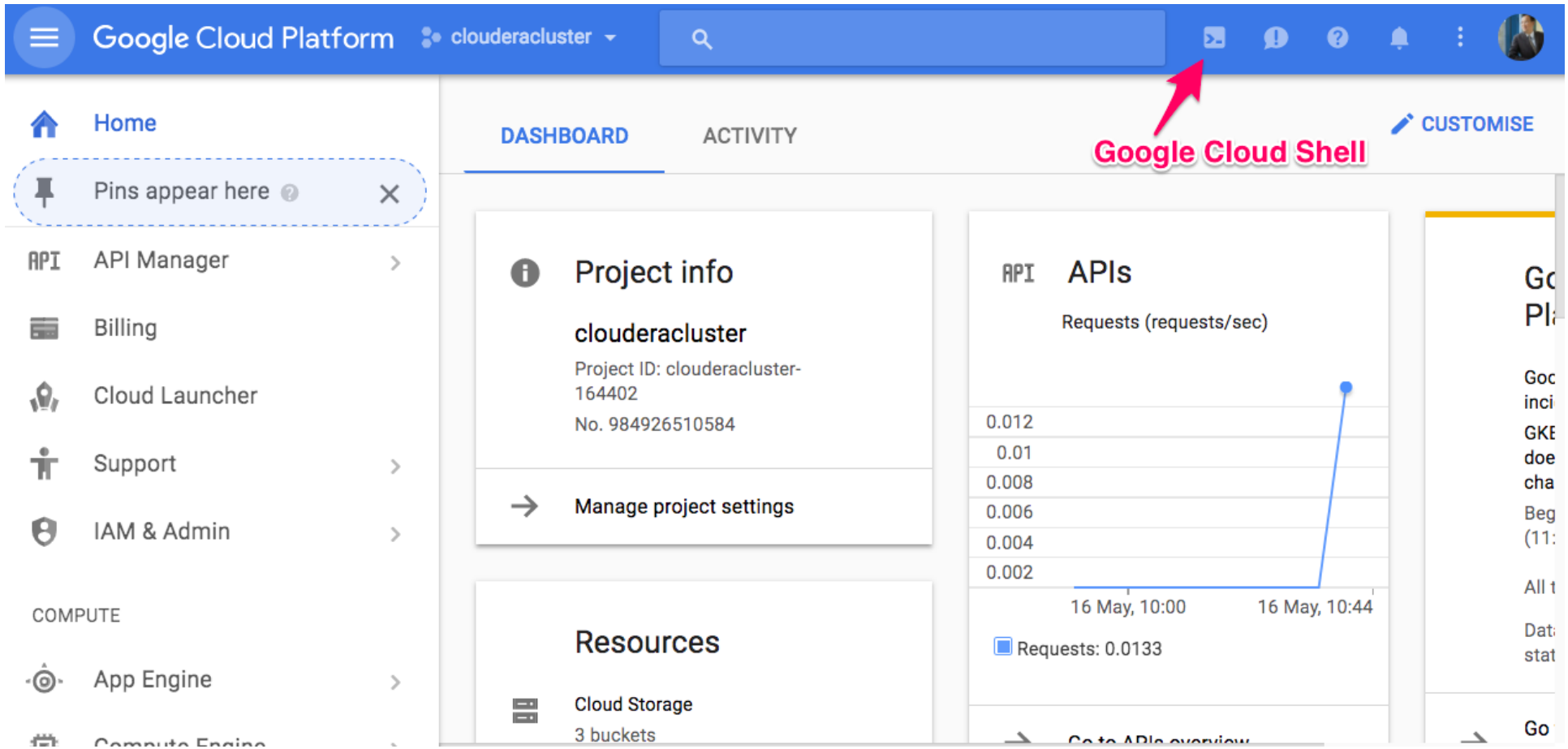
	Traditional	Google Cloud	AWS	Azure
Ingestion	Sqoop			
	Flume	DataFlow	Kenesis	Event Hub
	KafKa	Pub/Sub		
Storage	HDFS	Google Storage	S3	AZure Blob
	HBase/NoSQL	Big Table	DynamoDB	Storage Table
	RDBMS	Cloud SQL	RDS	Azure Database
Data warehouse	Teradata etc.		Redshift	Azure DW
Big Dta Processing	Hadoop/Spark	DataProc	EMR	HD Insight
	Jupiter	DataLab		Visual Studio
	Hive	Big Query	Athena	AZure Functions
Machine Learning	Spark MLlib	ML Engine	Machine Learning	Azure ML
Data Visiualization	Tableau etc.	Data Studio	QuickSight	Power BI

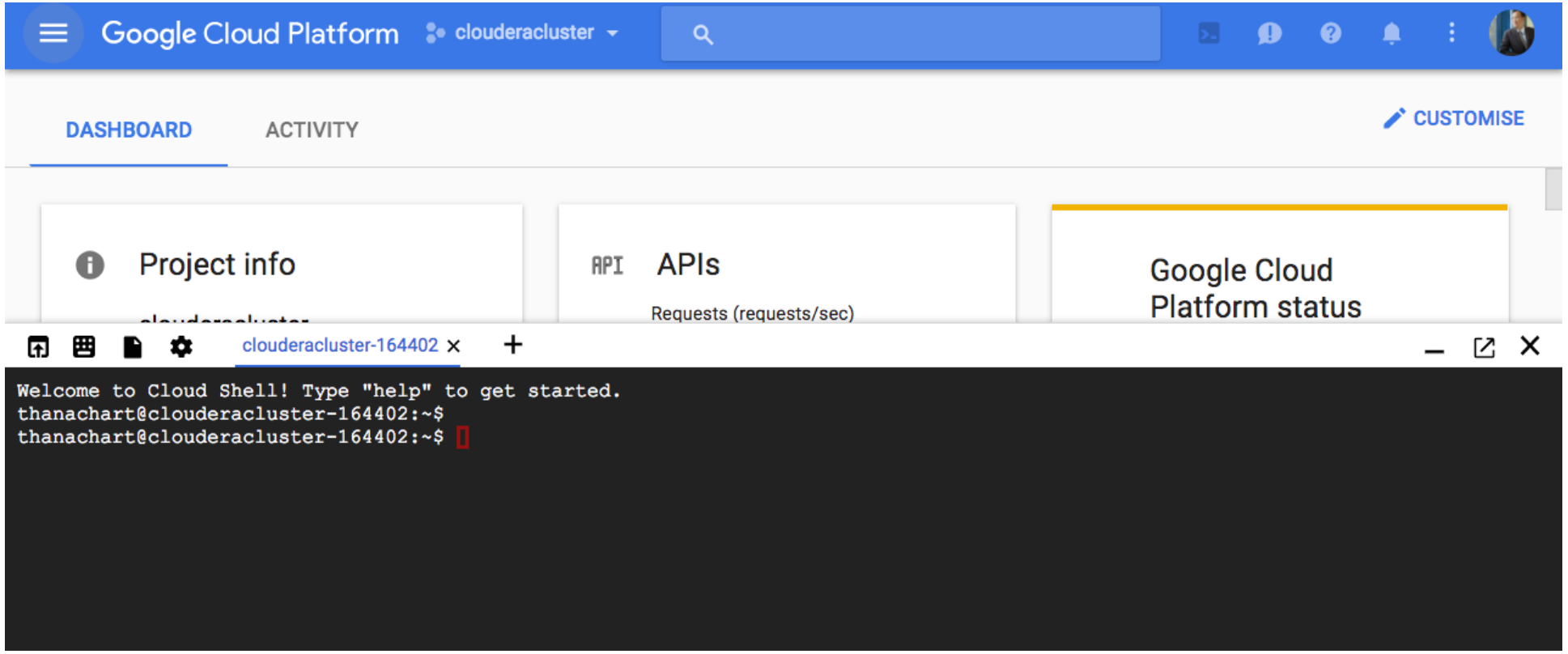
Google Cloud Shell

Google Cloud Shell is a shell environment for managing resources hosted on Google Cloud Platform. It is a micro VM and it is free of charge.

Launching Cloud Shell

- Select or create a Cloud Platform project.
- Click the Activate Google Cloud Shell button at the top of the Google Cloud Platform Console.
- A Cloud Shell session opens inside a new frame at the bottom of the console and displays a command-line prompt.


Google Cloud Platform
clouderacluster
Google Cloud Shell
Home
Pins appear here
API Manager
Billing
Cloud Launcher
Support
IAM & Admin
COMPUTE
App Engine
DASHBOARD
ACTIVITY
CUSTOMISE
Project info
clouderacluster
Project ID: clouderacluster-164402
No. 984926510584
Manage project settings
Resources
Cloud Storage
3 buckets
APIs
Requests (requests/sec)
0.012
0.01
0.008
0.006
0.004
0.002
16 May, 10:00
16 May, 10:44
Requests: 0.0133


Google Cloud Platform
clouderacluster
DASHBOARD
ACTIVITY
CUSTOMISE
Project info
APIs
Requests (requests/sec)
Google Cloud Platform status
clouderacluster-164402
Welcome to Cloud Shell! Type "help" to get started.
thanachart@clouderacluster-164402:~$
thanachart@clouderacluster-164402:~$

Google Cloud Storage

Google Cloud Storage is unified object storage for developers and enterprises, from live data serving to data analytics/ML to data archiving. It provides a unified offering across the availability spectrum: from live data tapped by today's most demanding applications, to cloud archival solutions Nearline and Coldline. Featuring a consistent API, latency, and speed across storage classes

Cloud Storage

- Multi-Regional or Regional storage
- Nearline and Coldline storage solutions
- Offers a unified product offering with consistent access APIs across the entire range of storage classes
- With no minimum fees and a pay-per-use model,

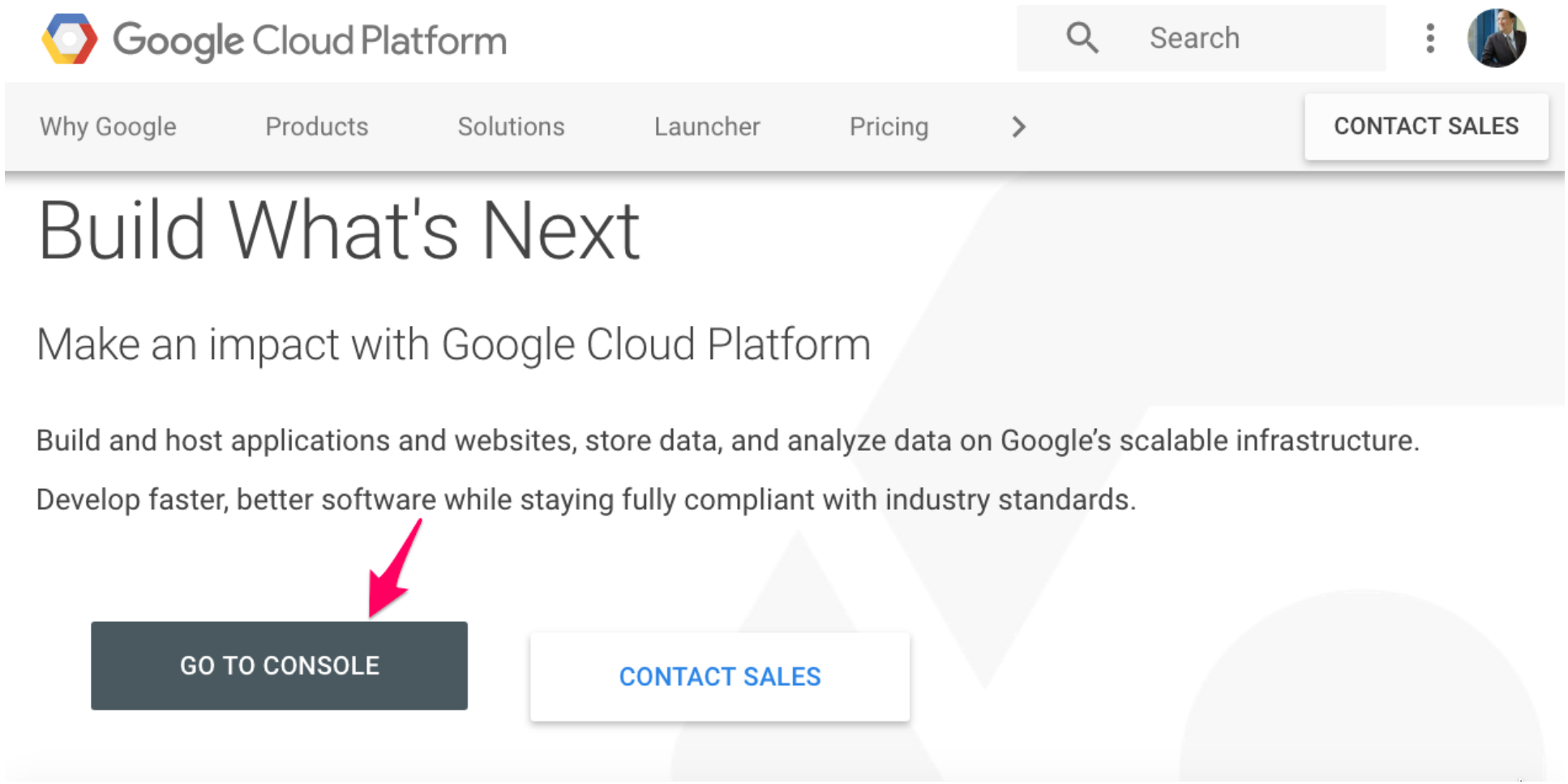
Cloud Storage

General Pricing

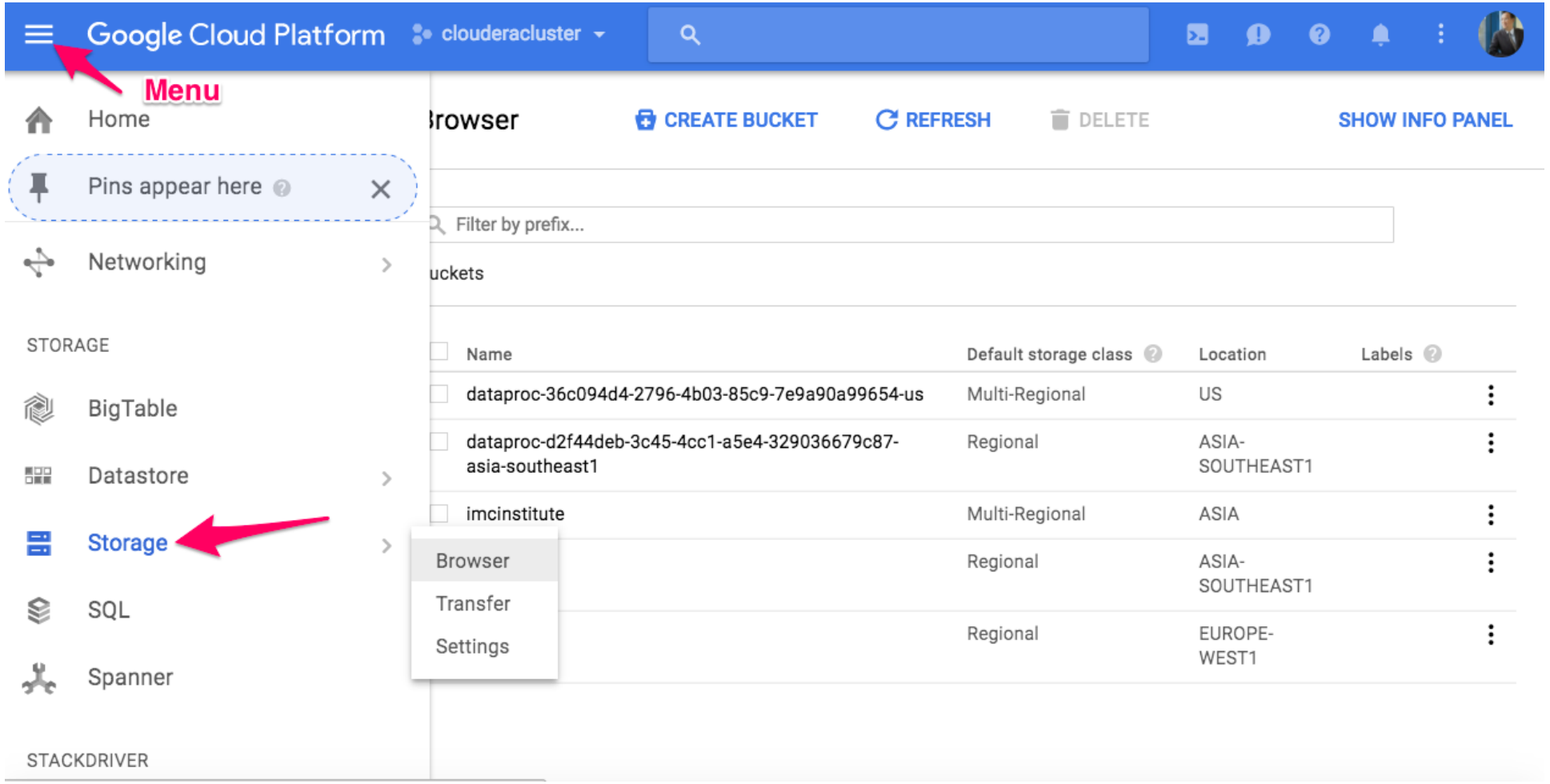
Multi-Regional Storage (per GB per Month)	Regional Storage (per GB per Month)	Nearline Storage (per GB per Month)	Coldline Storage (per GB per Month)
$0.026	$0.02	$0.01	$0.007

Lab I: Importing/Exporting data to Google Cloud Storage

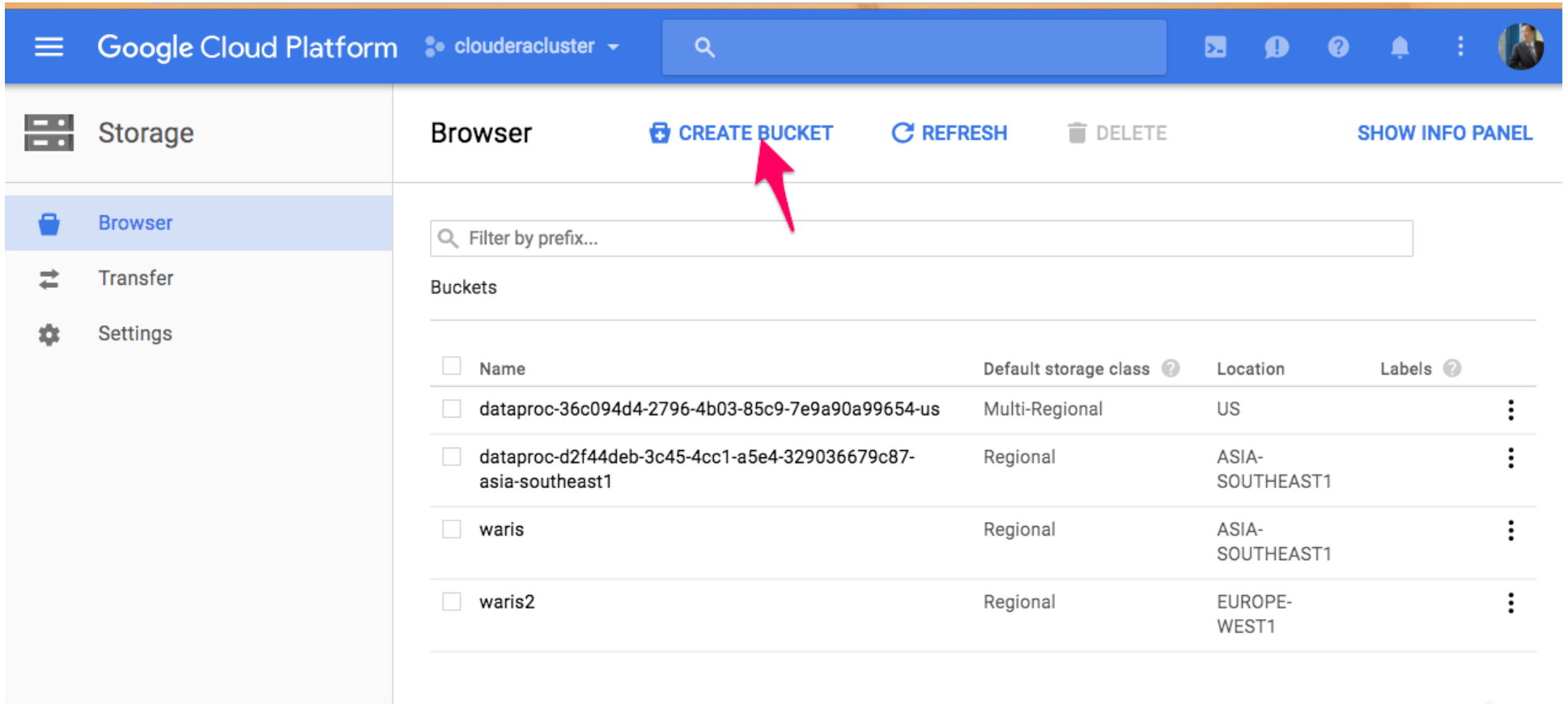
Launch Google Cloud console



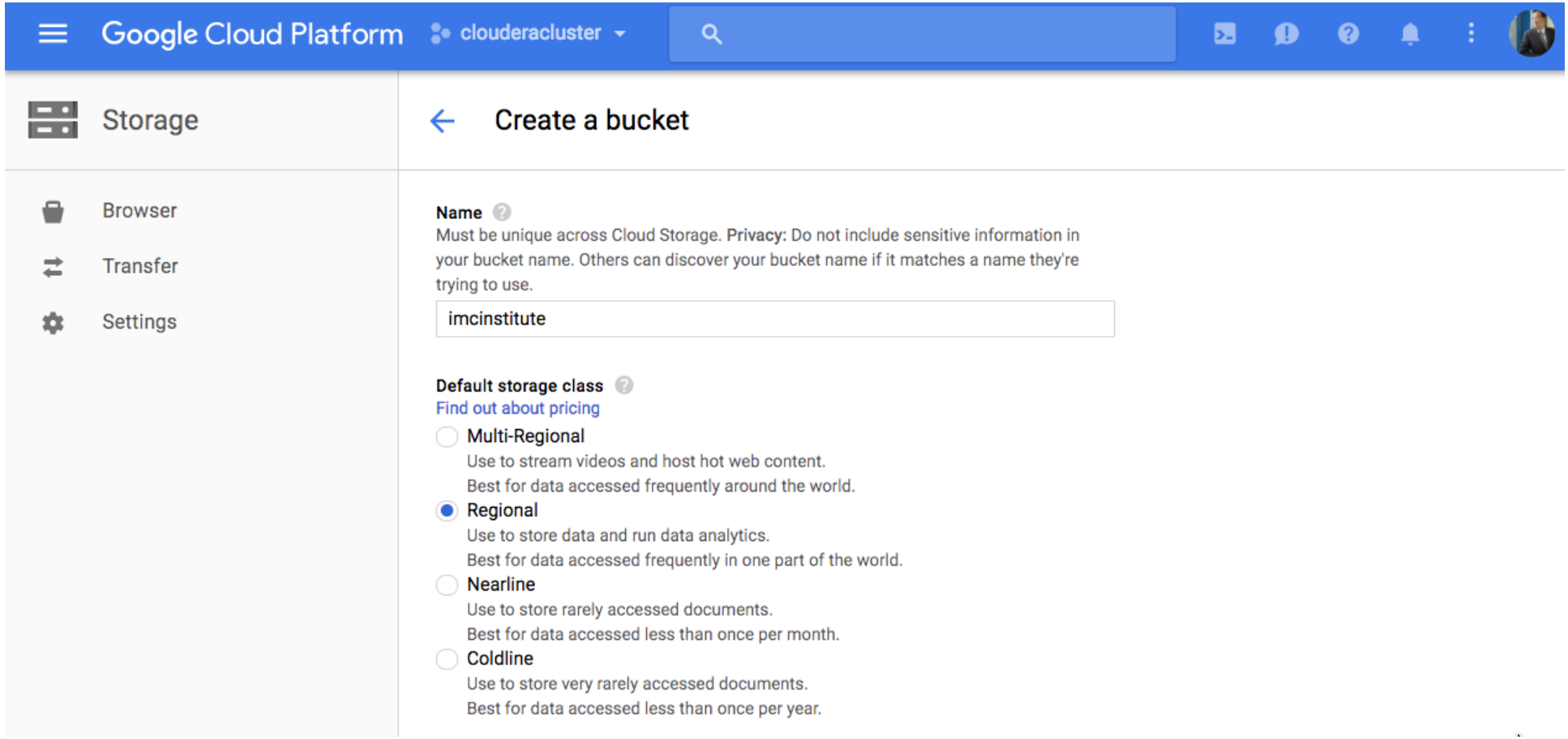
Select Storage



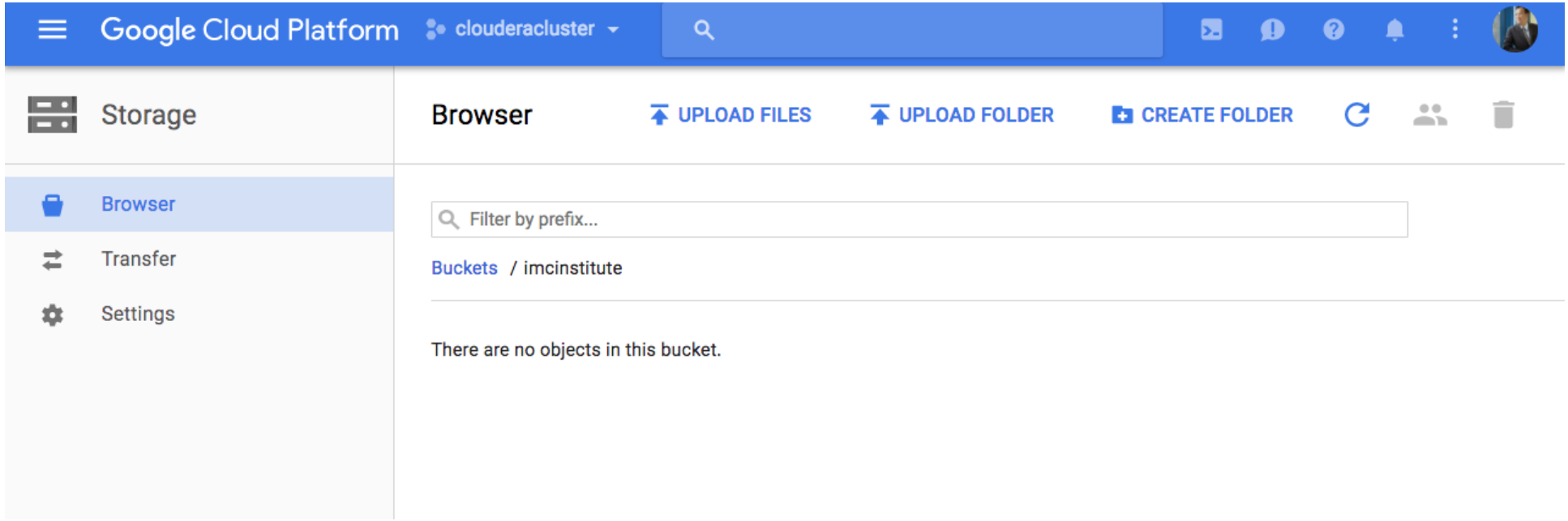
Click on Create Bucket



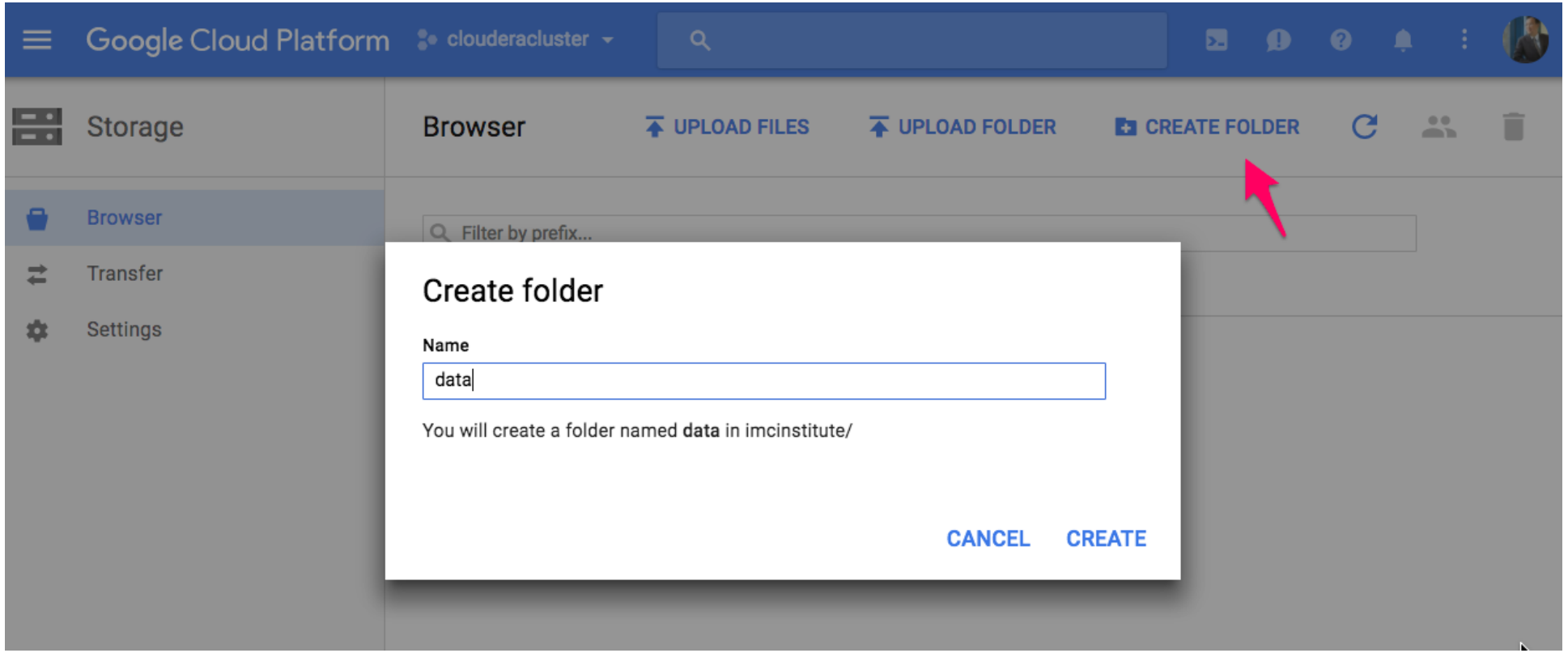
Choose a globally unique bucket name

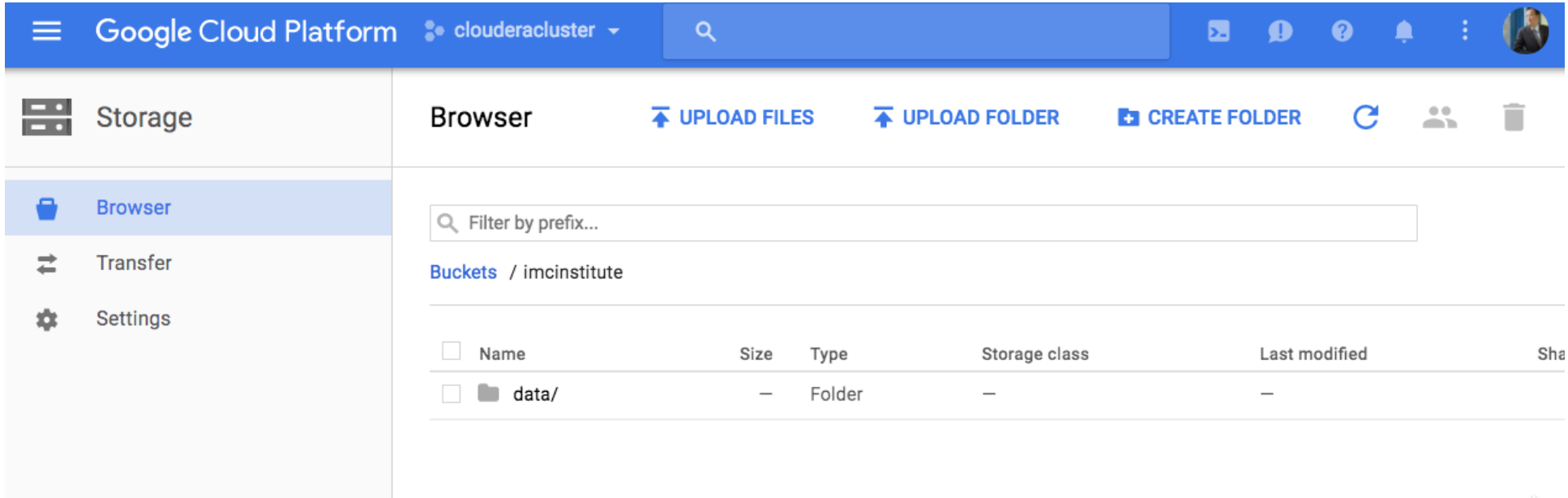


Review objects in the bucket

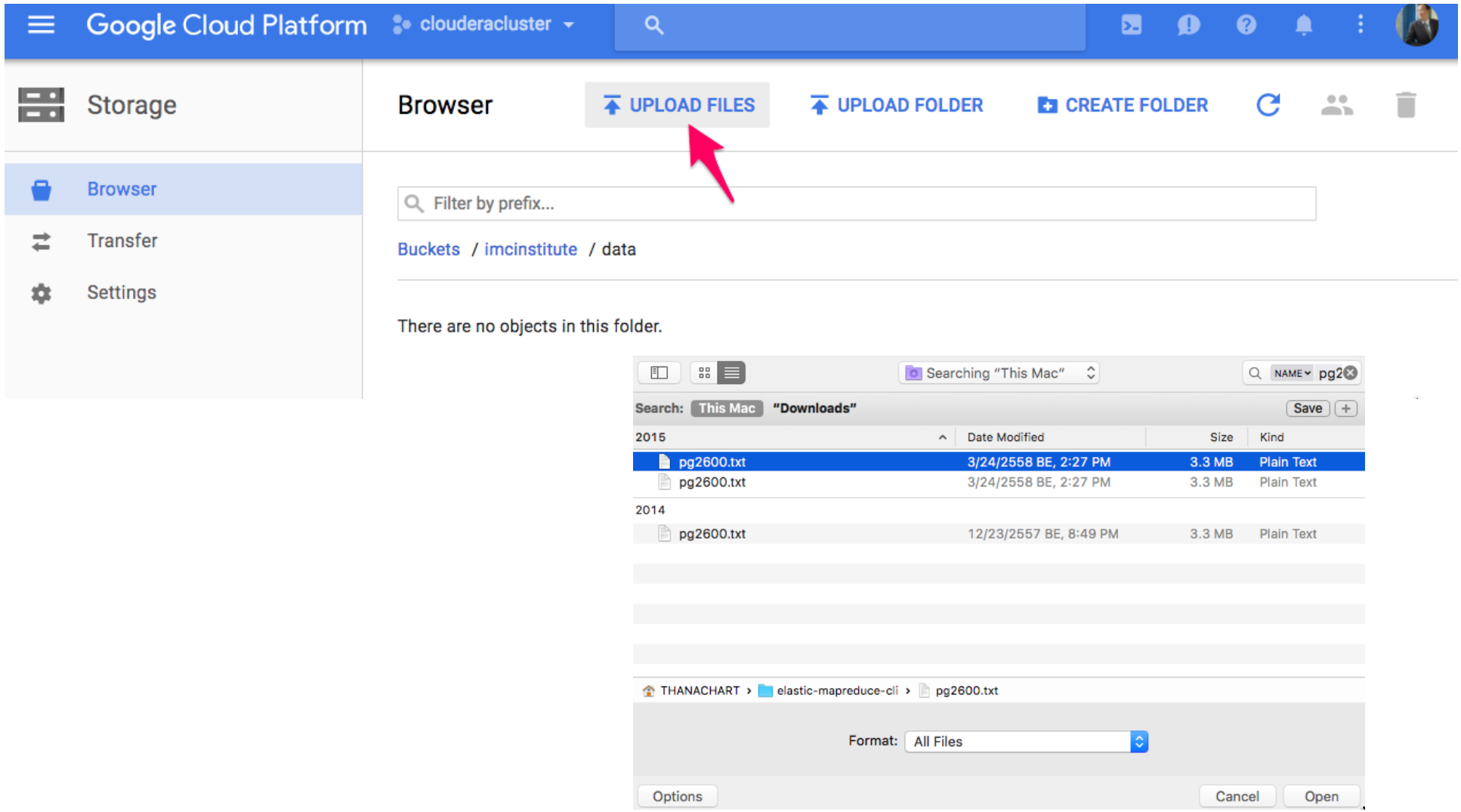


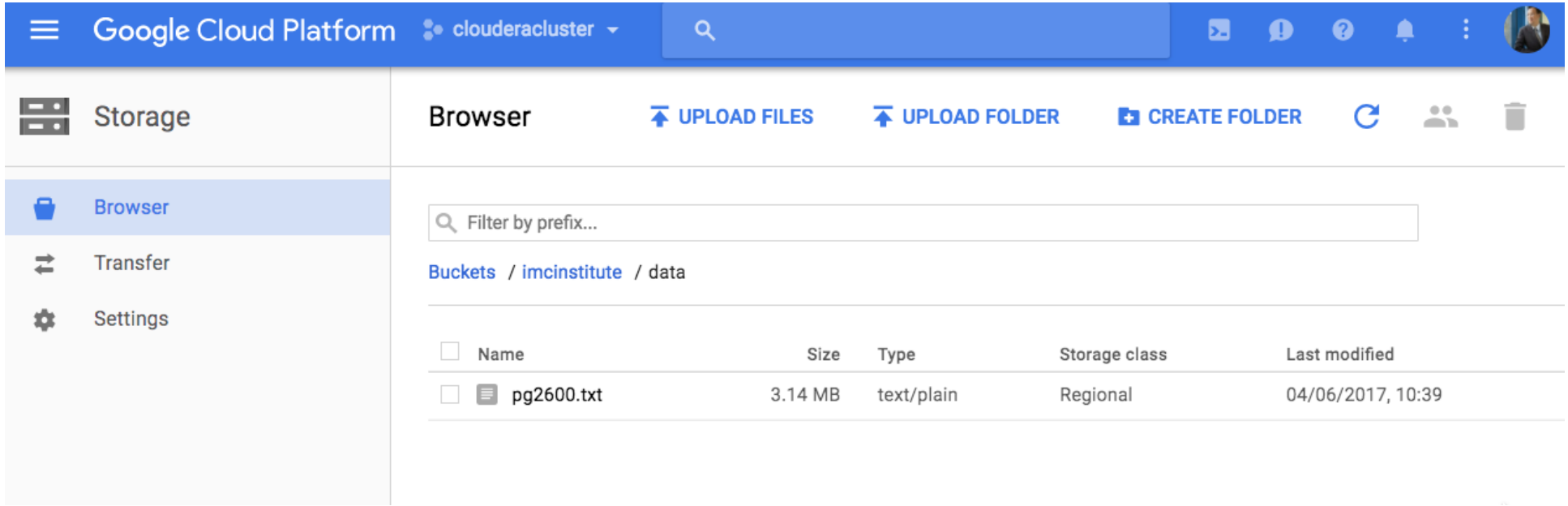
Create a new folder name as: data



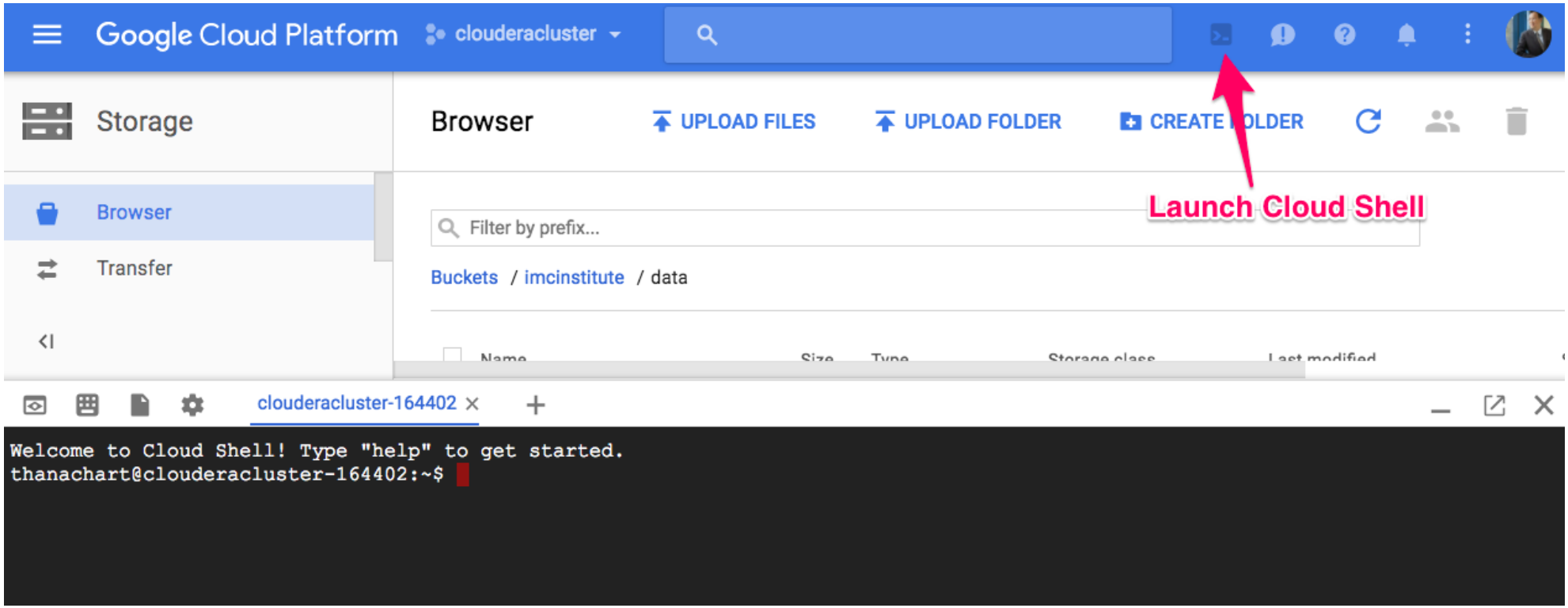

Google Cloud Platform
clouderacluster
Storage
Browser
UPLOAD FILES
UPLOAD FOLDER
CREATE FOLDER
Browser
Transfer
Settings
Filter by prefix...
Buckets / imcinstitute
Name
Size
Type
Storage class
Last modified
data/
–
Folder
–
–

Upload a local file to the cloud storage




Google Cloud Platform
clouderacluster
Storage
Browser
UPLOAD FILES
UPLOAD FOLDER
CREATE FOLDER
Browser
Transfer
Settings
Filter by prefix...
Buckets / imcinstitute / data
Name
Size
Type
Storage class
Last modified
pg2600.txt
3.14 MB
text/plain
Regional
04/06/2017, 10:39

Upload a large file using Cloud shell



Download an example text file using wget command

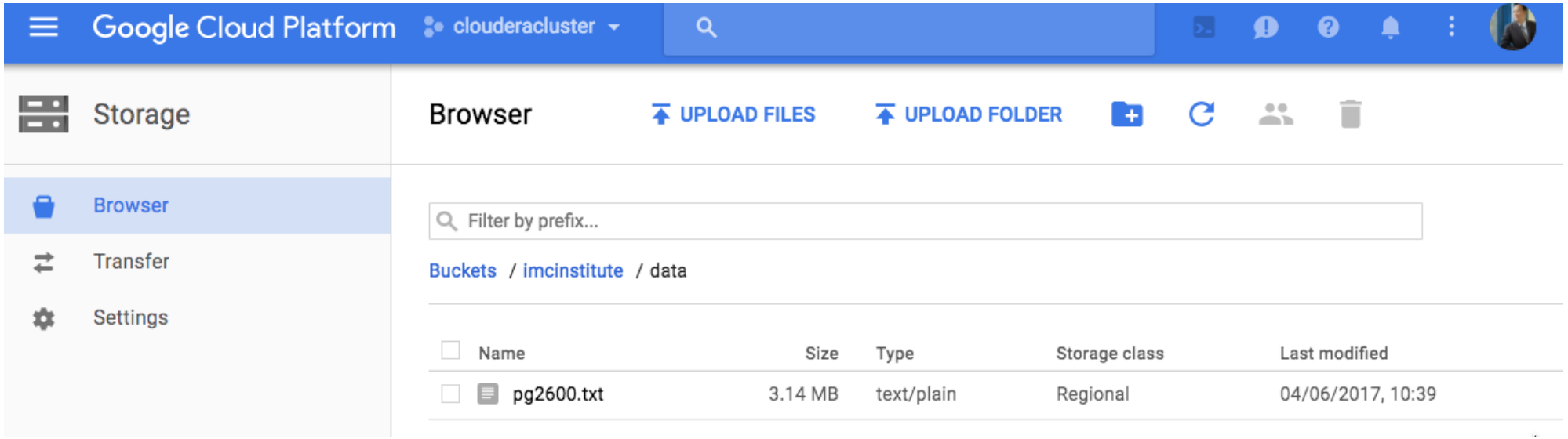
```
$wget https://s3.amazonaws.com/imcbucket/input/pg2600.txt
```

```
Welcome to Cloud Shell! Type "help" to get started.
thanachart@clouderacluster-164402:~$ wget https://s3.amazonaws.com/imcbucket/input/pg2600.txt
--2017-06-04 10:47:06--  https://s3.amazonaws.com/imcbucket/input/pg2600.txt
Resolving s3.amazonaws.com (s3.amazonaws.com)... 52.216.1.3
Connecting to s3.amazonaws.com (s3.amazonaws.com)|52.216.1.3|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3291648 (3.1M) [text/plain]
Saving to: 'pg2600.txt'

pg2600.txt                      100%[=============================================================>]   3.14M   402KB/s   in 19s

2017-06-04 10:47:27 (165 KB/s) - 'pg2600.txt' saved [3291648/3291648]

thanachart@clouderacluster-164402:~$
```

Upload Data to Google cloud storage using gsutil command

```
$gsutil cp pg2600.txt gs://<YOUR-BUCKET>/
```

```
thanachart@clouderacluster-164402:~$ gsutil cp pg2600.txt gs://imcinstitute/
Copying file://pg2600.txt [Content-Type=text/plain]...
- [1 files][  3.1 MiB/  3.1 MiB]
Operation completed over 1 objects/3.1 MiB.
thanachart@clouderacluster-164402:~$
```


Google Cloud Platform
clouderacluster
Storage
Browser
UPLOAD FILES
UPLOAD FOLDER
Browser
Transfer
Settings
Filter by prefix...
Buckets / imcinstitute / data
Name
Size
Type
Storage class
Last modified
pg2600.txt
3.14 MB
text/plain
Regional
04/06/2017, 10:39

Google Cloud Dataproc

Google Cloud Dataproc is a managed Spark and Hadoop service which is fast, easy to use, and low cost

Google Cloud Dataproc

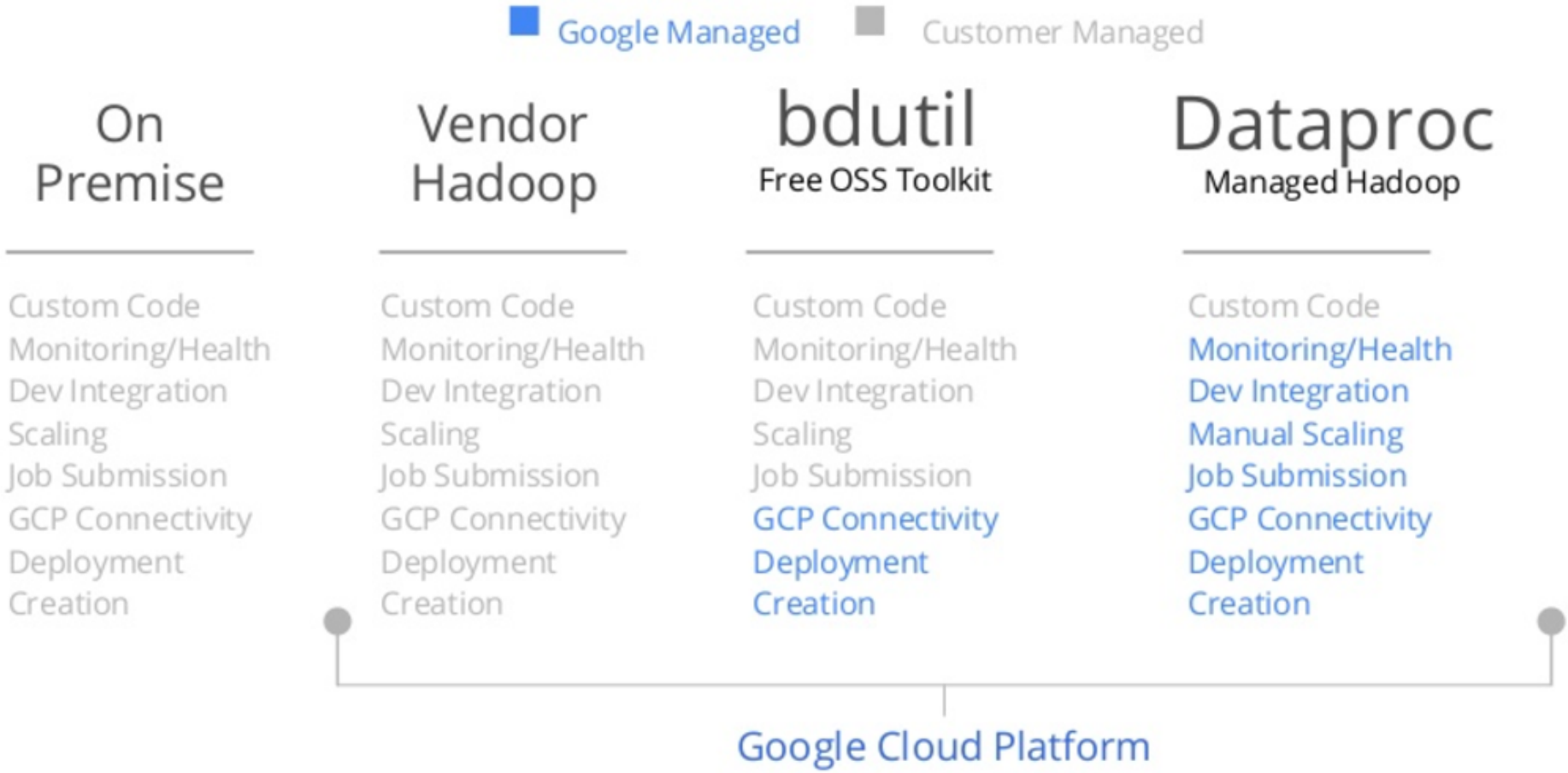
- Managed Hadoop & Spark, an Apache Hadoop, Apache Spark, Apache Pig, and Apache Hive service, to easily process big datasets at low cost.

- Fast & Scalable Data Processing

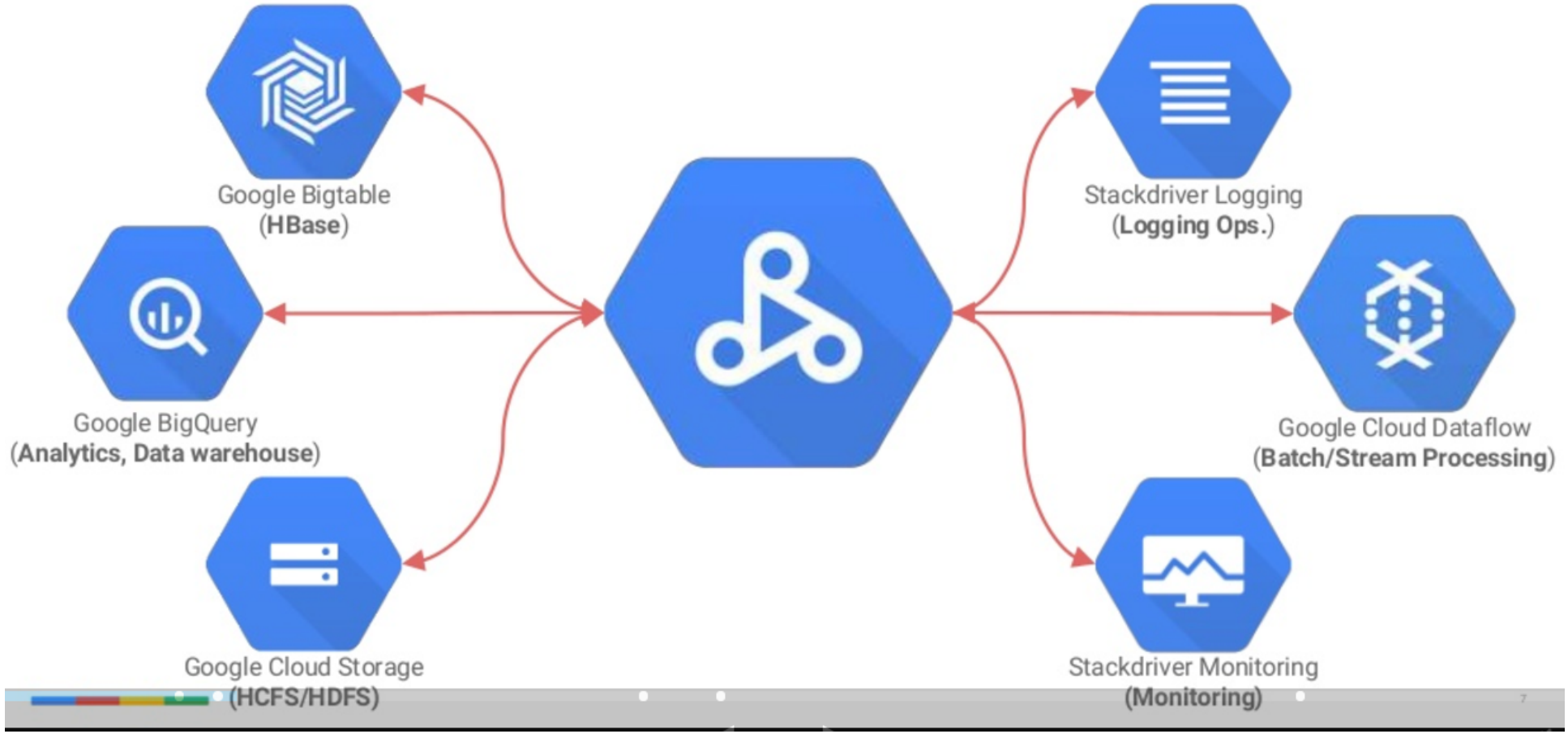
- Affordable Pricing

- Open Source Ecosystem

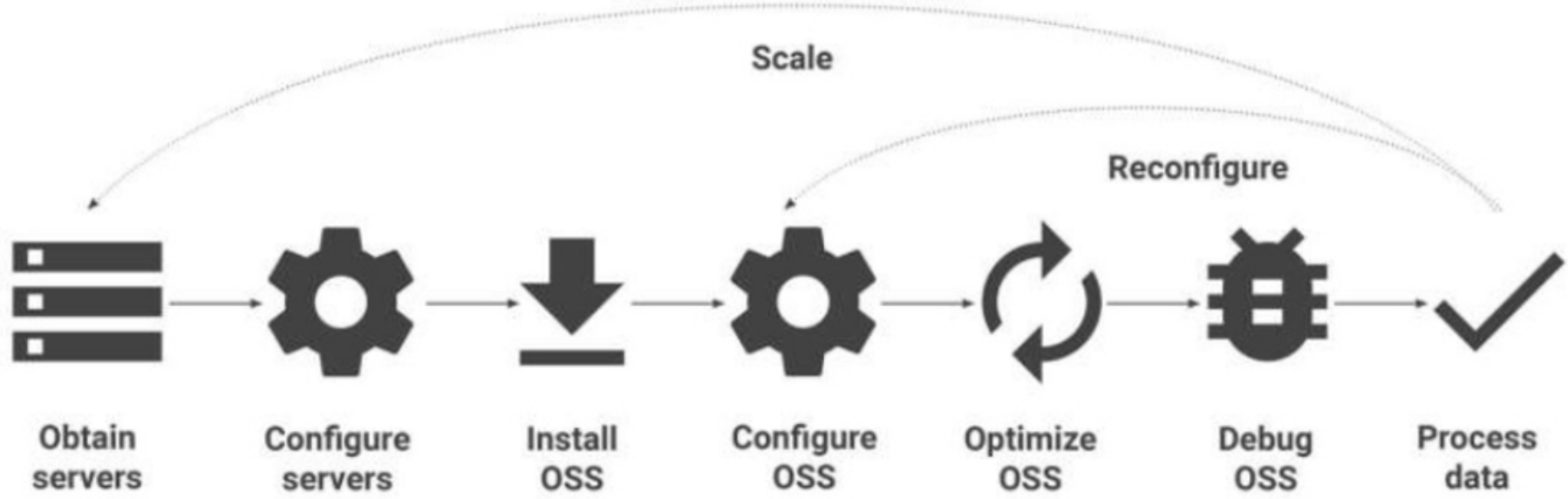
Running Hadoop on Google Cloud



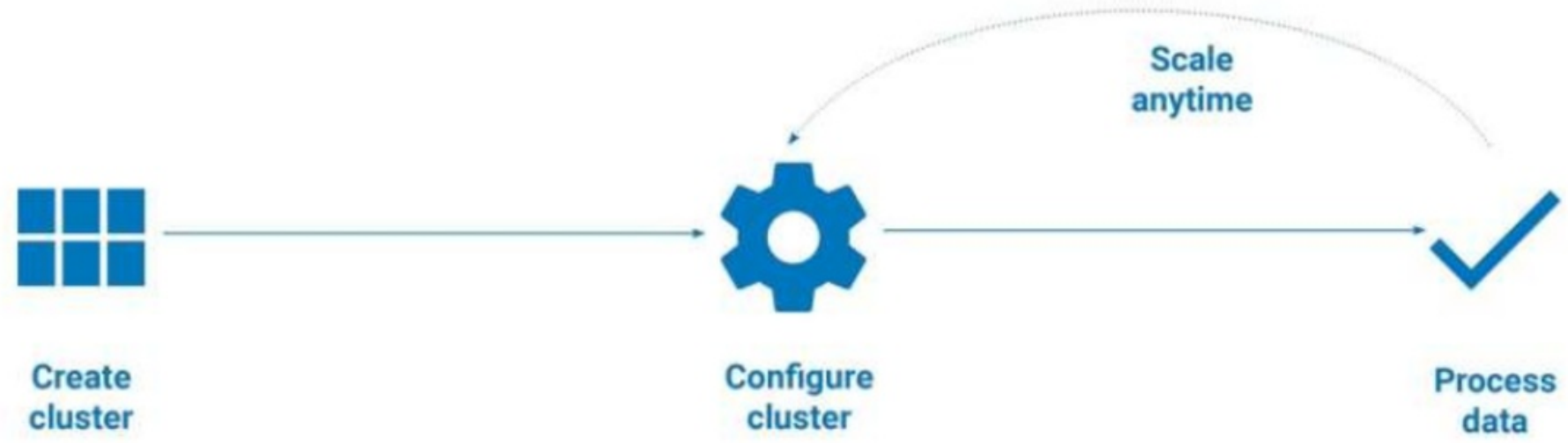
Where Cloud Dataproc fits into GCP



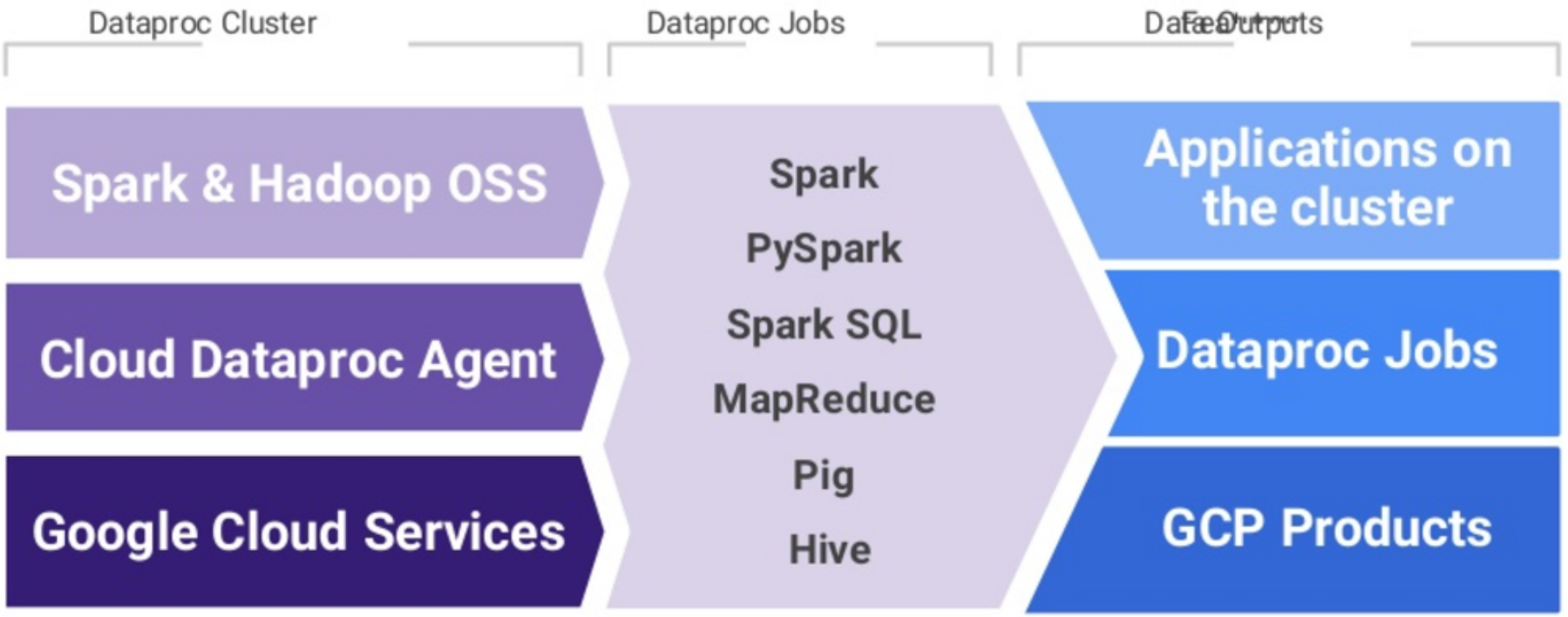
Traditional Spark and Hadoop clusters



Google Cloud Dataproc

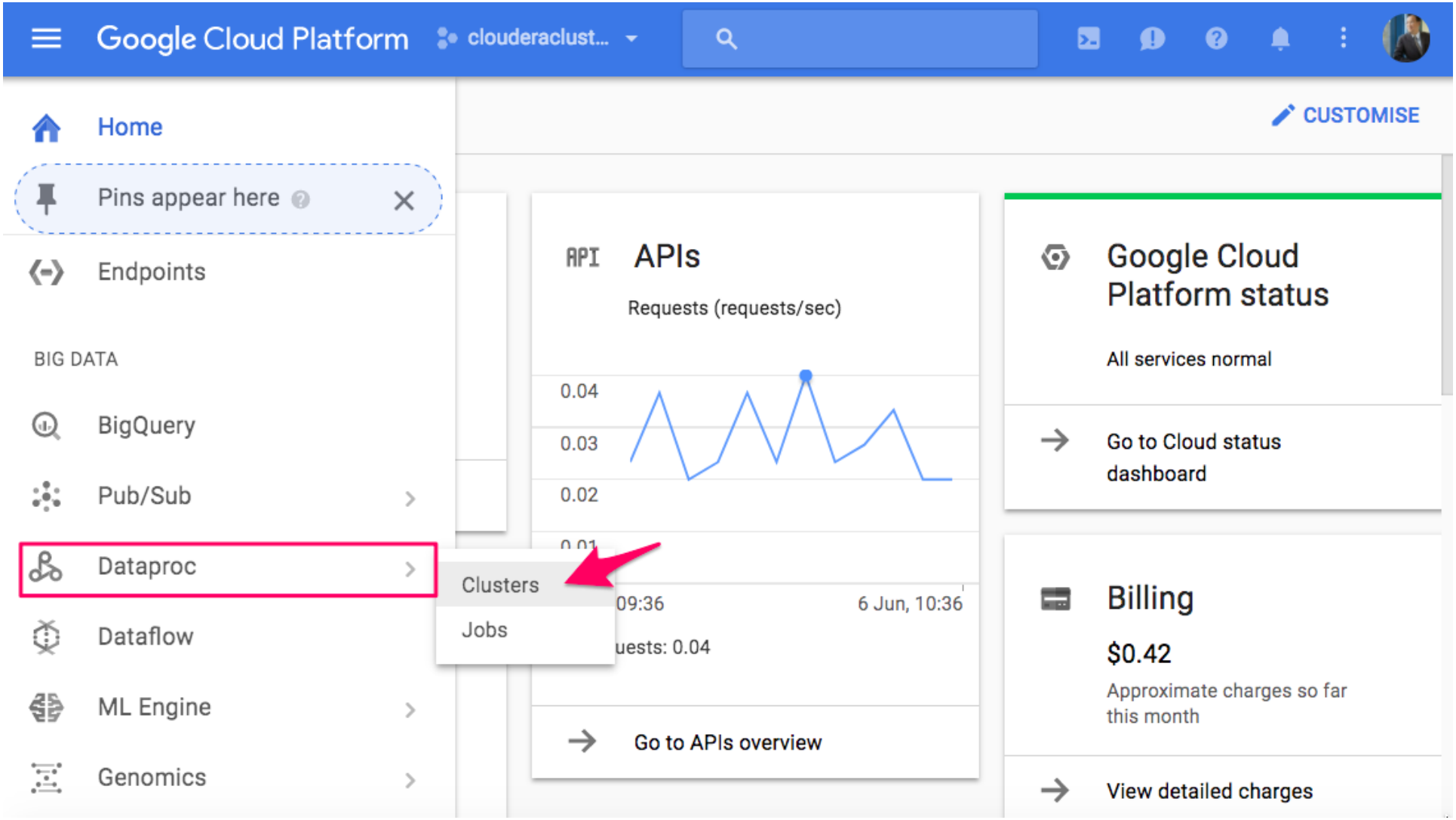


Google Cloud Dataproc - **under the hood**

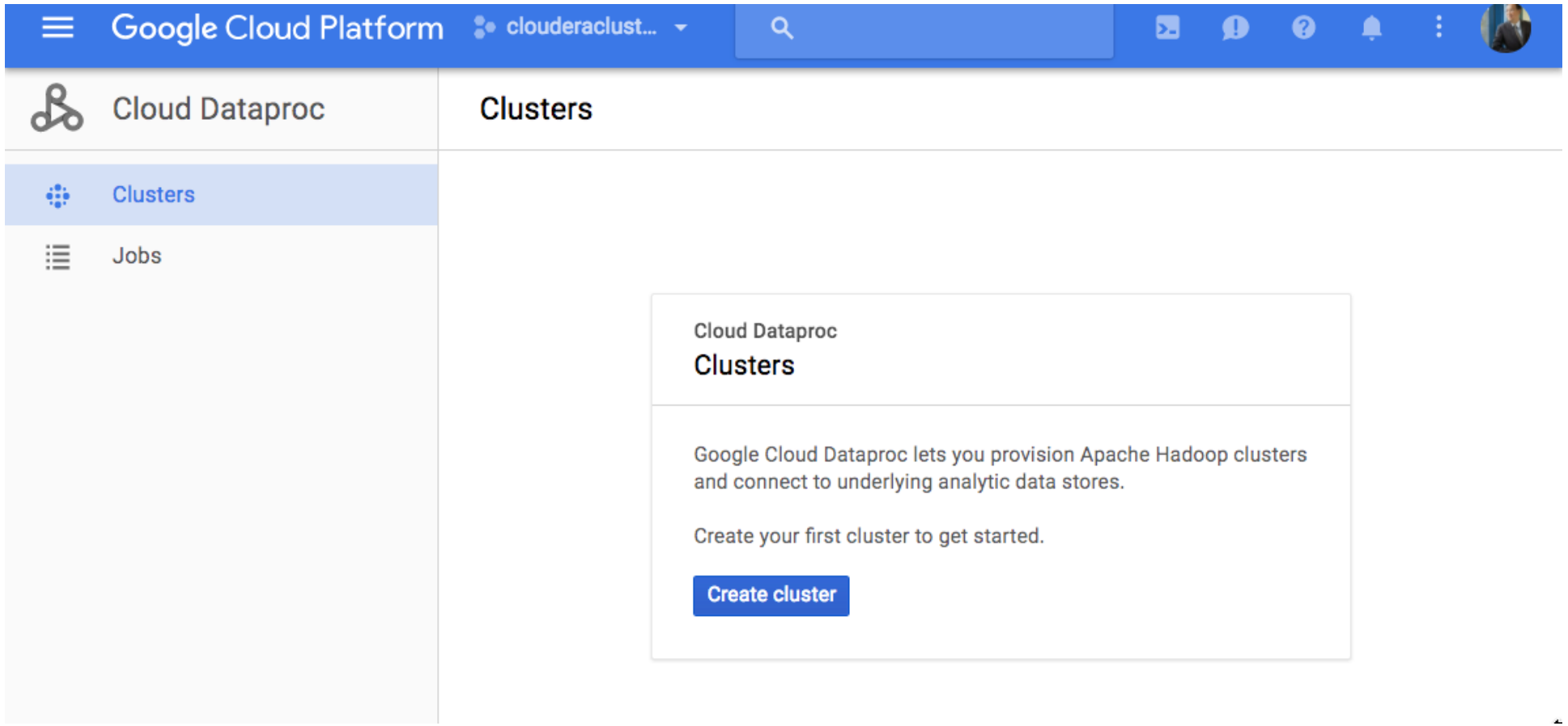


Lab II：Launch Dataproc

select Dataproc >> Clusters

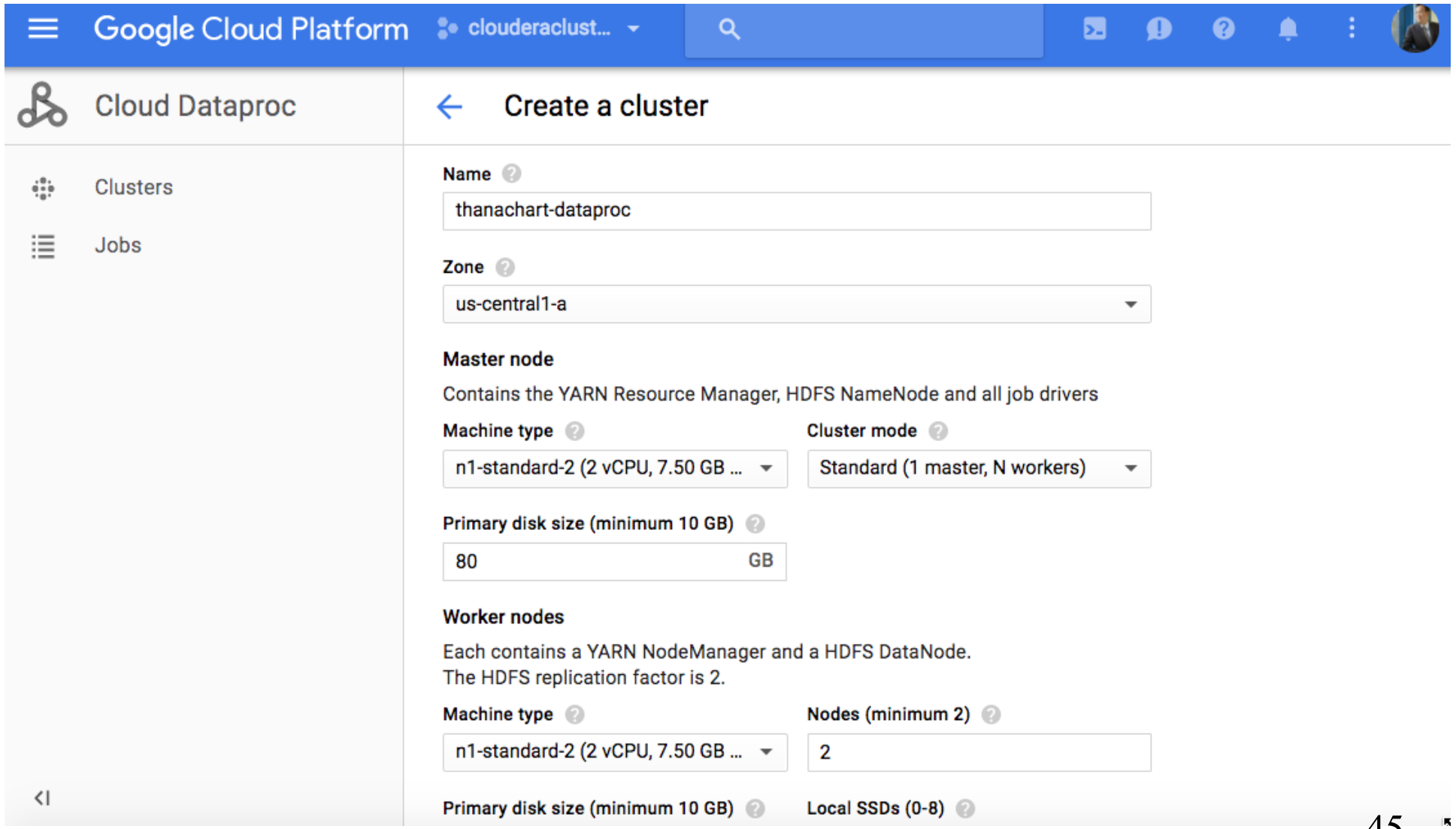


click Create cluster



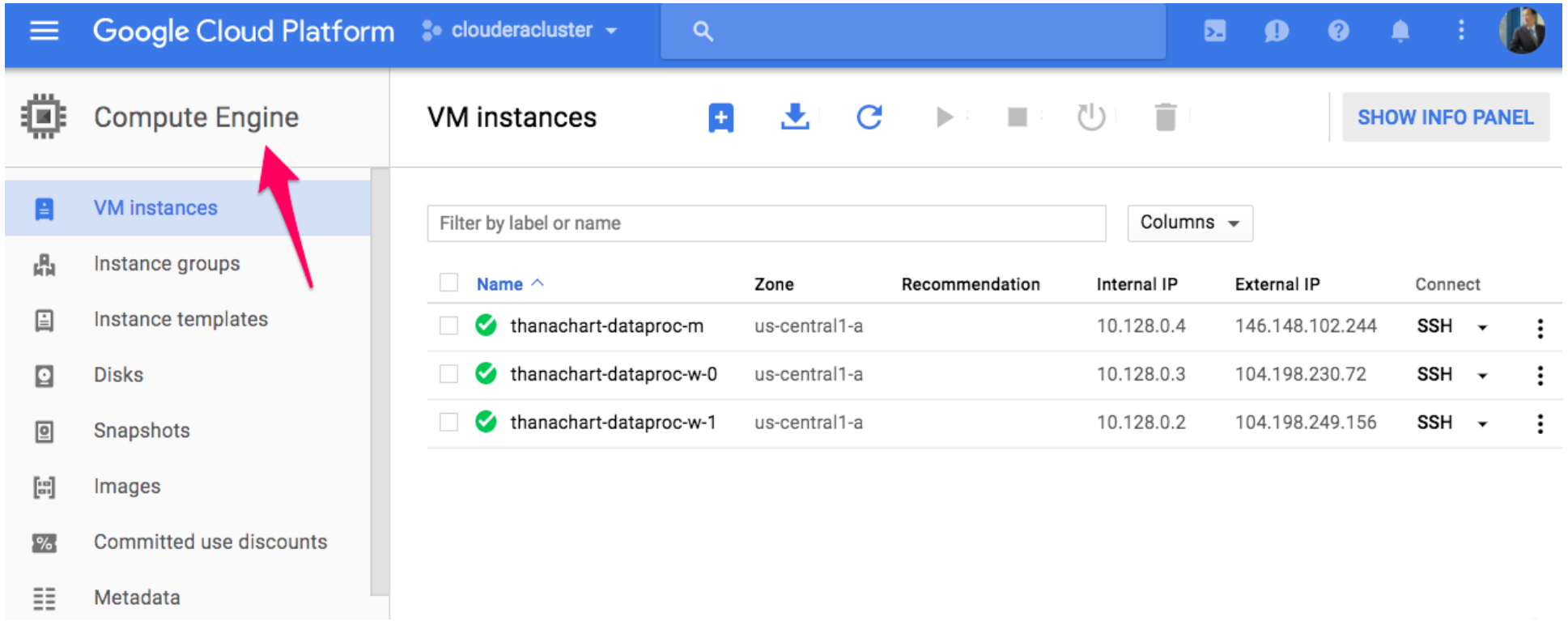
Using the following configuration

- Name the cluster.
- Choose the region
- Change the machine type of both the Master and the Worker nodes to n1-standard-2
- Select disk size to 80 GB

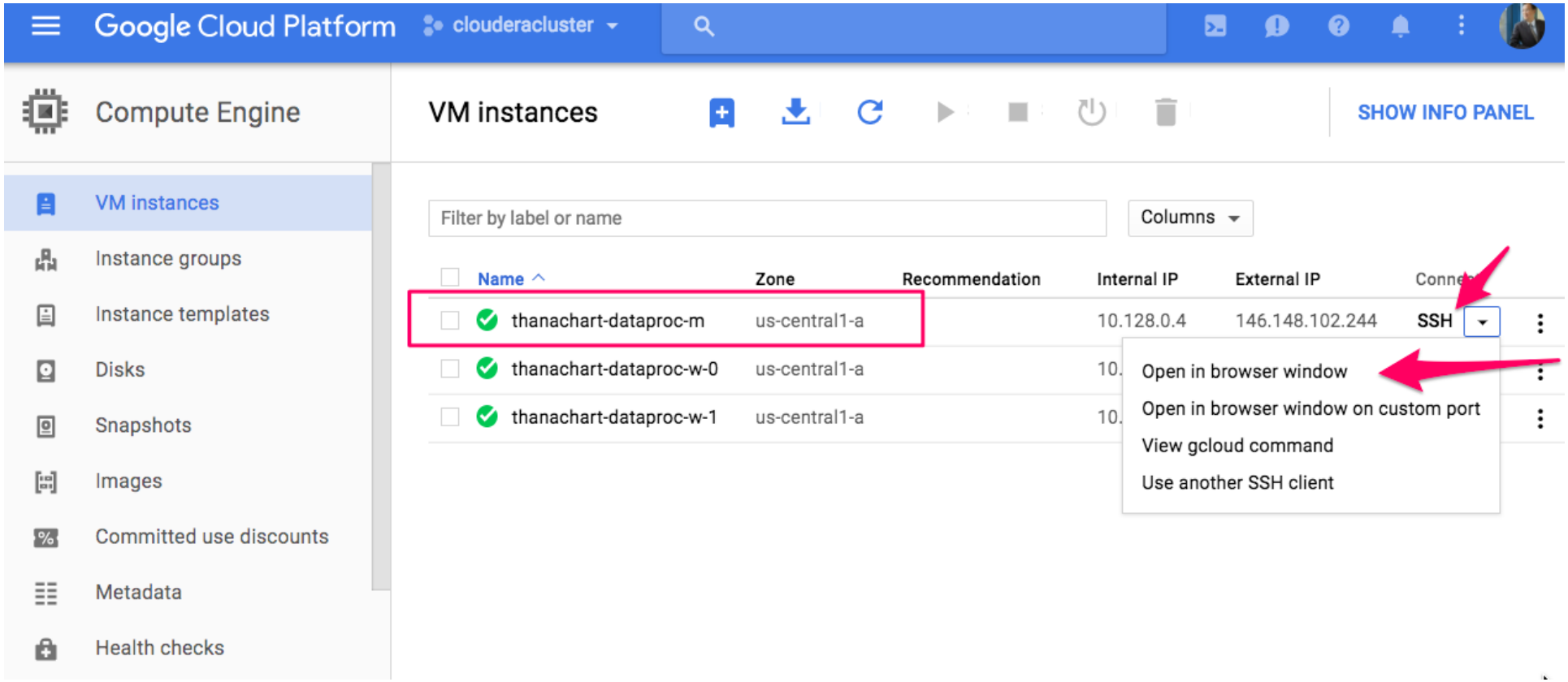

Google Cloud Platform
clouderaclust...
Cloud Dataproc
Clusters
Jobs
Create a cluster
Name
thanachart-dataproc
Zone
us-central1-a
Master node
Contains the YARN Resource Manager, HDFS NameNode and all job drivers
Machine type
n1-standard-2 (2 vCPU, 7.50 GB ...
Cluster mode
Standard (1 master, N workers)
Primary disk size (minimum 10 GB)
80
GB
Worker nodes
Each contains a YARN NodeManager and a HDFS DataNode.
The HDFS replication factor is 2.
Machine type
n1-standard-2 (2 vCPU, 7.50 GB ...
Nodes (minimum 2)
2
Primary disk size (minimum 10 GB)
Local SSDs (0-8)

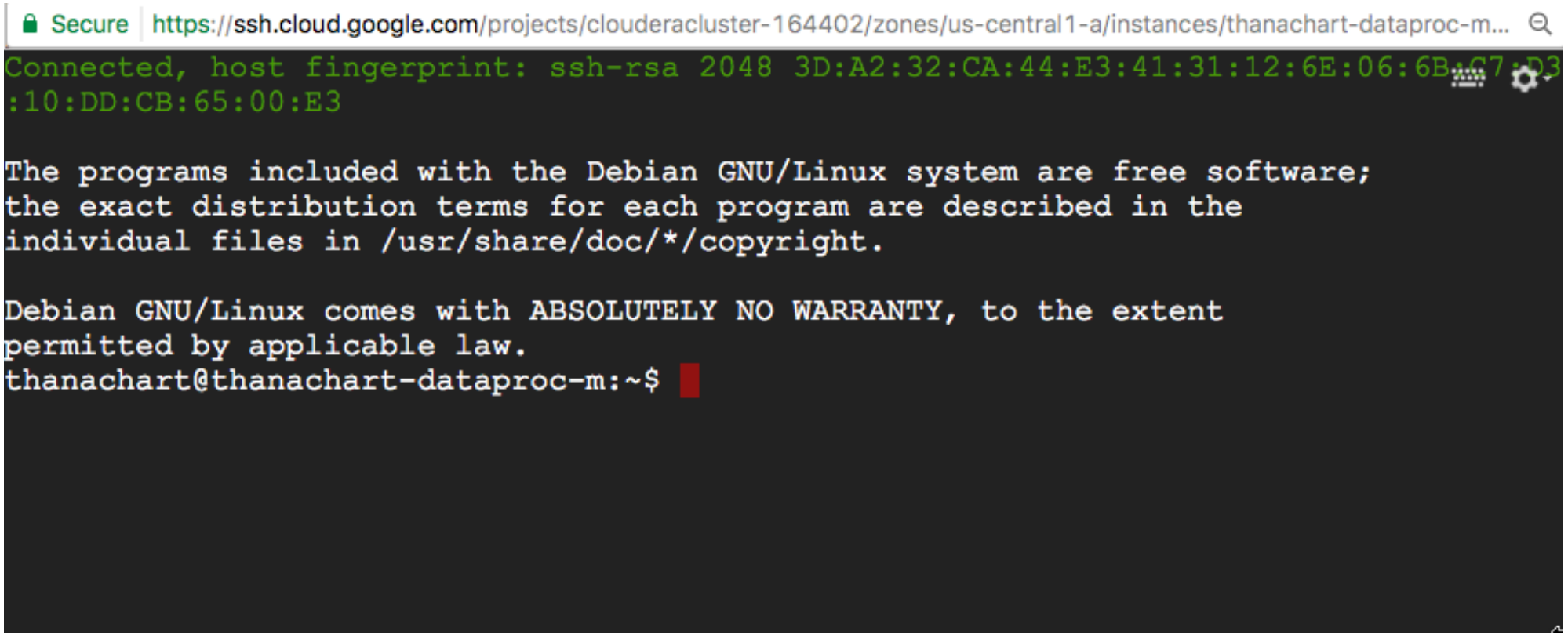
Lab III: Running a Pig script

Select the Dataproc master instance from Compute engine



Start SSH from the master node




Secure
https://ssh.cloud.google.com/projects/clouderacluster-164402/zones/us-central1-a/instances/thanachart-dataproc-m...
Connected, host fingerprint: ssh-rsa 2048 3D:A2:32:CA:44:E3:41:31:12:6E:06:6B:C7:D3
:10:DD:CB:65:00:E3
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.
thanachart@thanachart-dataproc-m:~$

Move Data from Google cloud storage to HDFS

```
$ gsutil cp gs://<YOUR-BUCKET>/pg2600.txt .
$ hadoop fs -mkdir /user/test
$ hadoop fs -put pg2600.txt /user/test
```

```
thanachart@thanachart-dataproc-m:~$ gsutil cp gs://imcinstitute/pg2600.txt .
Copying gs://imcinstitute/pg2600.txt...
/ [1 files][  3.1 MiB/  3.1 MiB]
Operation completed over 1 objects/3.1 MiB.
```

Starting Pig Command Line

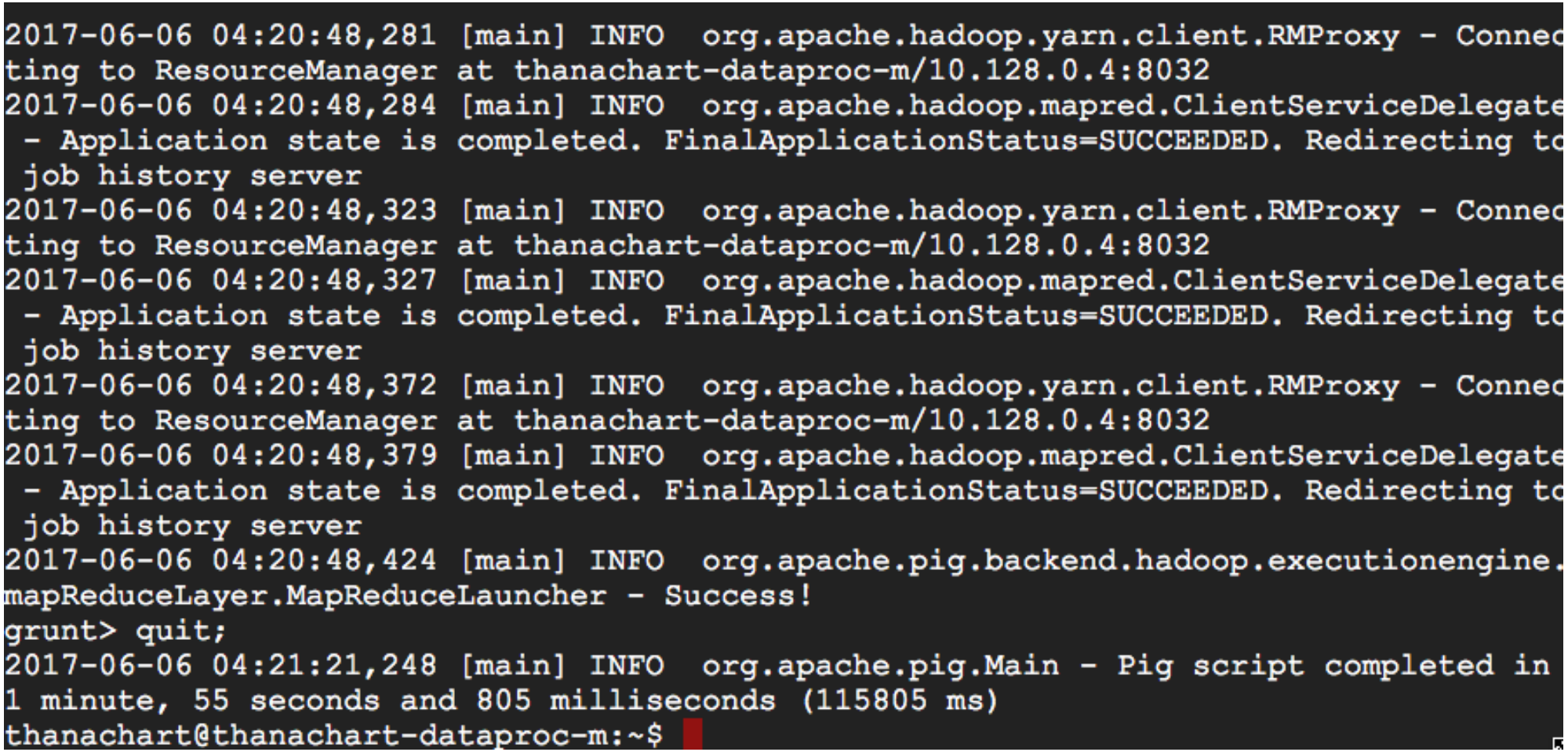
```
thanachart@thanachart-dataproc-m:~$ pig -x mapreduce
```

```
thanachart@thanachart-dataproc-m:~$ pig -x mapreduce

17/06/06 04:07:12 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
17/06/06 04:07:12 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
17/06/06 04:07:12 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType
2017-06-06 04:07:12,981 [main] INFO  org.apache.pig.Main - Apache Pig version 0.16.
0 (r: unknown) compiled May 18 2017, 01:29:04
2017-06-06 04:07:12,981 [main] INFO  org.apache.pig.Main - Logging error messages t
o: /home/thanachart/pig_1496722032980.log
2017-06-06 04:07:13,004 [main] INFO  org.apache.pig.impl.util.Utils - Default bootu
p file /home/thanachart/.pigbootup not found
2017-06-06 04:07:13,480 [main] INFO  org.apache.hadoop.conf.Configuration.deprecati
on - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2017-06-06 04:07:13,481 [main] INFO  org.apache.hadoop.conf.Configuration.deprecati
on - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-06-06 04:07:13,481 [main] INFO  org.apache.pig.backend.hadoop.executionengine.
HExecutionEngine - Connecting to hadoop file system at: hdfs://thanachart-dataproc-
m
2017-06-06 04:07:13,511 [main] INFO  com.google.cloud.hadoop.fs.gcs.GoogleHadoopFil
eSystemBase - GHFS version: 1.6.1-hadoop2
2017-06-06 04:07:14,092 [main] INFO  org.apache.pig.PigServer - Pig Script ID for t
he session: PIG-default-bac626fc-7afe-48cd-9f24-86adc3c08c4a
2017-06-06 04:07:14,092 [main] WARN  org.apache.pig.PigServer - ATS is disabled sin
ce yarn.timeline-service.enabled set to false
grunt>
```

Writing a Pig Script for wordcount

Suggestion: Before run below codes, please make sure you that the file pg2600.txt still exists

```
A = load '/user/test/*';
B = foreach A generate flatten(TOKENIZE((chararray)$0)) as
word;
C = group B by word;
D = foreach C generate COUNT(B), group;
store D into '/user/test/wordcountPig';
```


2017-06-06 04:20:48,281 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connec
ting to ResourceManager at thanachart-dataproc-m/10.128.0.4:8032
2017-06-06 04:20:48,284 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate
- Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to
job history server
2017-06-06 04:20:48,323 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connec
ting to ResourceManager at thanachart-dataproc-m/10.128.0.4:8032
2017-06-06 04:20:48,327 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate
- Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to
job history server
2017-06-06 04:20:48,372 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connec
ting to ResourceManager at thanachart-dataproc-m/10.128.0.4:8032
2017-06-06 04:20:48,379 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate
- Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to
job history server
2017-06-06 04:20:48,424 [main] INFO org.apache.pig.backend.hadoop.executionengine.
mapReduceLayer.MapReduceLauncher - Success!
grunt> quit;
2017-06-06 04:21:21,248 [main] INFO org.apache.pig.Main - Pig script completed in
1 minute, 55 seconds and 805 milliseconds (115805 ms)
thanachart@thanachart-dataproc-m:~$

Viewing result in the HDFS

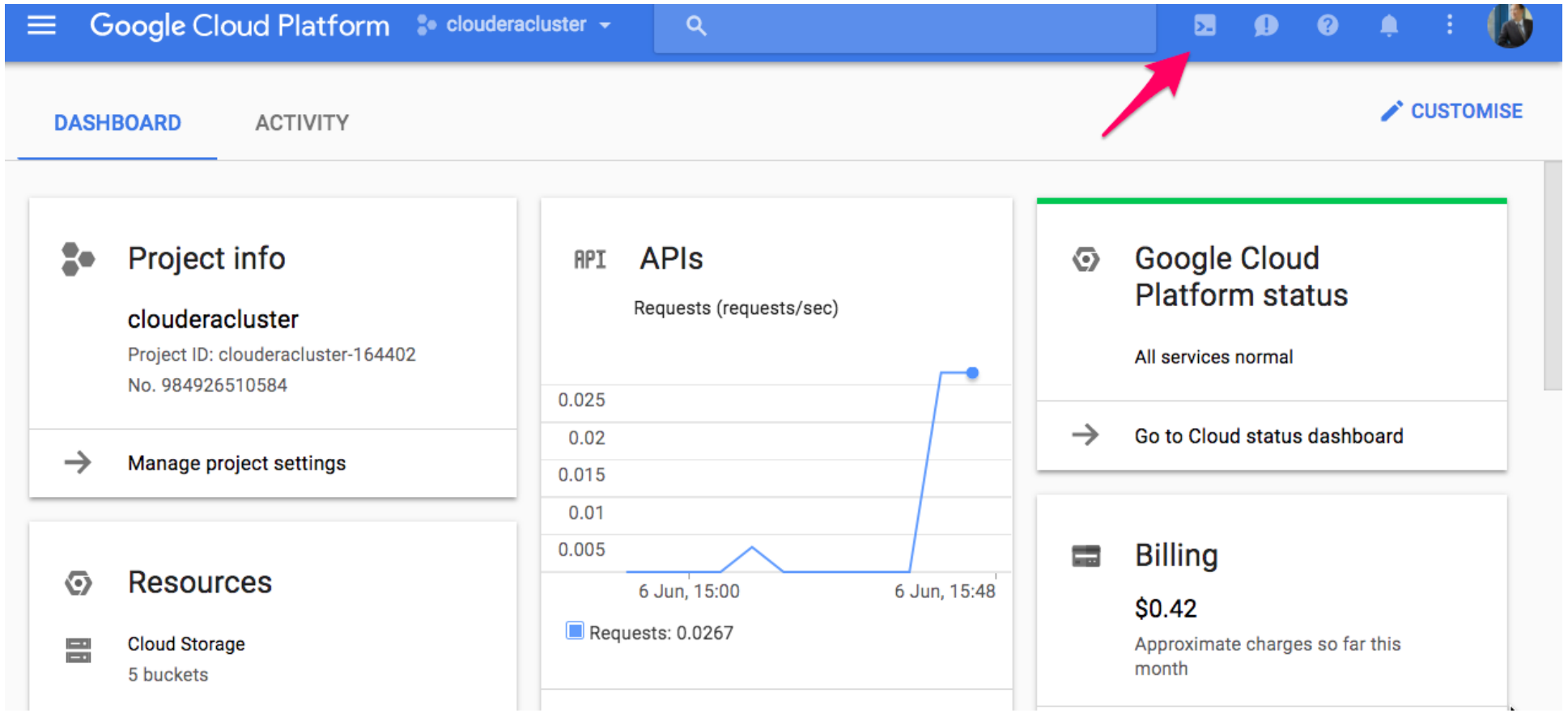
```
$ hadoop fs -ls /user/test/wordcountPig
$ hadoop fs -cat /user/test/wordcountPig/part-r-00000
```

```
thanachart@thanachart-dataproc-m:~$ hadoop fs -ls /user/test/wordcountPig
17/06/06 04:26:54 INFO gcs.GoogleHadoopFileSystemBase: GHFS version: 1.6.1-hadoop2
Found 2 items
-rw-r--r--   2 thanachart hadoop          0 2017-06-06 04:20 /user/test/wordcountPig/_SUCCESS
-rw-r--r--   2 thanachart hadoop     353282 2017-06-06 04:20 /user/test/wordcountPig/part-r-00000
thanachart@thanachart-dataproc-m:~$ hadoop fs -cat /user/test/wordcountPig/part-r-00000
```

Lab IV：Find top 10 best/worst airlines Using Spark

Launch Dataproc using gcloud command:

with Jupyter Notebook I) Launch Cloud Shell



Upload a file to dataproc

```
$ wget https://s3.amazonaws.com/imcbucket/data/flights/2008.csv

$ gsutil cp 2008.csv gs://<YOUR-BUCKET>/
```

```
thanachart@thanachart-dataproc-m:~$ wget  https://s3.amazonaws.com/imcbucket/data/flights/2008.csv
--2017-06-06 04:30:07--  https://s3.amazonaws.com/imcbucket/data/flights/2008.csv
Resolving s3.amazonaws.com (s3.amazonaws.com)... 54.231.81.91
Connecting to s3.amazonaws.com (s3.amazonaws.com)|54.231.81.91|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 689413344 (657M) [binary/octet-stream]
Saving to: '2008.csv'

2008.csv                 100%[=====================================>] 657.48M  70.0MB/s   in 16s

2017-06-06 04:30:23 (40.3 MB/s) - '2008.csv' saved [689413344/689413344]

thanachart@thanachart-dataproc-m:~$ gsutil cp 2008.csv gs://imcinstitute/
```

Launch Dataproc using gcloud command with Jupyter Notebook

II) Type the following command

- gcloud dataproc clusters create datascience \

  --zone us-central1-a \

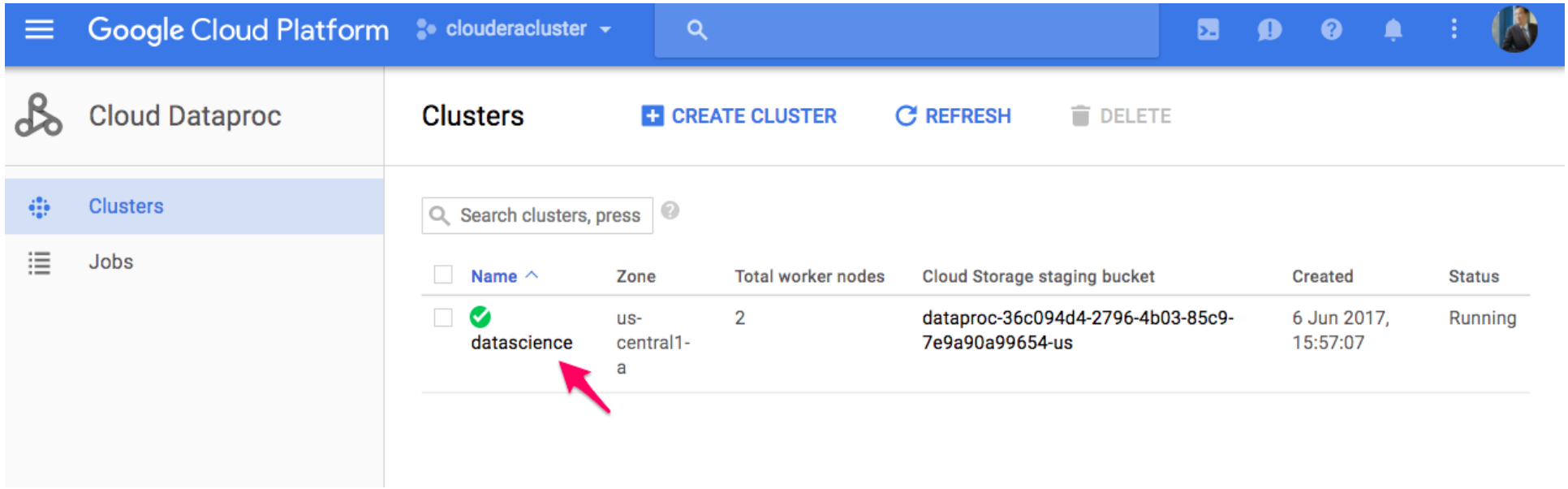
  --master-machine-type=n1-standard-2 \

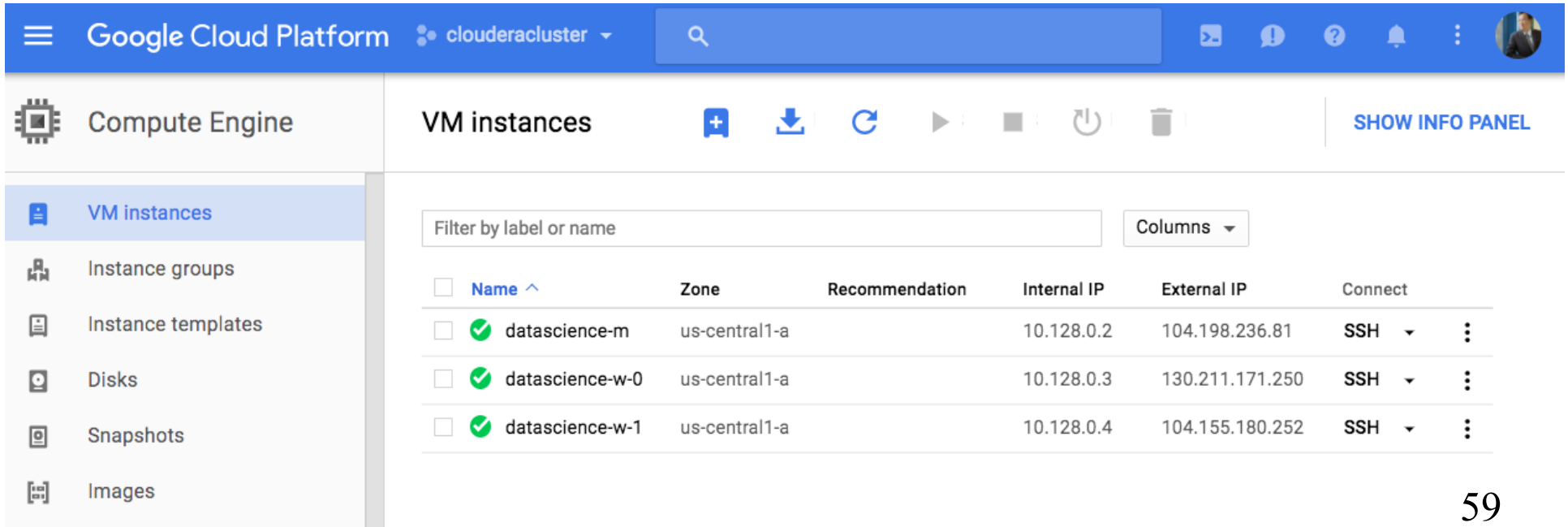
  --worker-machine-type=n1-standard-2 \

  --initialization-actions \

  gs://dataproc-initialization-actions/jupyter/jupyter.sh

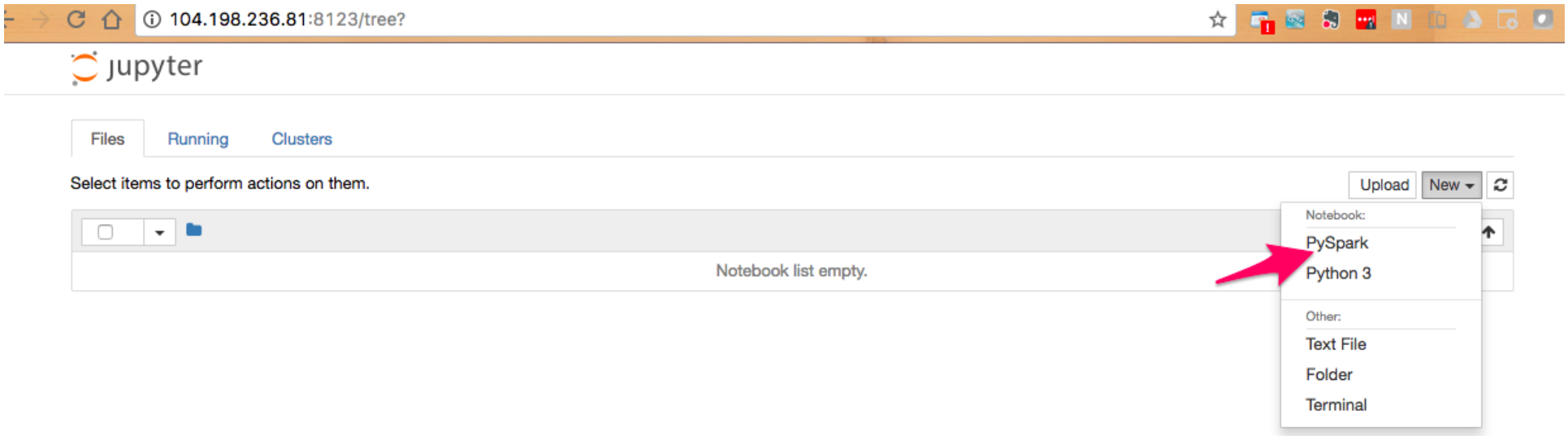
```
thanachart@clouderacluster-164402:~$    --worker-machine-type=n1-standard-2 \
>     --initialization-actions \
>         gs://dataproc-initialization-actions/jupyter/jupyter.sh
-bash: --worker-machine-type=n1-standard-2: command not found
thanachart@clouderacluster-164402:~$ gcloud dataproc clusters create datascience    --zone us-central1-a    --master-machine-type
=n1-standard-2  --worker-machine-type=n1-standard-2 \
>     --initialization-actions \
>         gs://dataproc-initialization-actions/jupyter/jupyter.sh
Waiting on operation [projects/clouderacluster-164402/regions/global/operations/2ad165f0-c8c1-44cb-ab61-2d13f64077ce].
Waiting for cluster creation operation...done.
Created [https://dataproc.googleapis.com/v1/projects/clouderacluster-164402/regions/global/clusters/datascience].
thanachart@clouderacluster-164402:~$
```


Google Cloud Platform
clouderacluster
Cloud Dataproc
Clusters
CREATE CLUSTER
REFRESH
DELETE
Clusters
Jobs
Search clusters, press
Name
Zone
Total worker nodes
Cloud Storage staging bucket
Created
Status
datascience
us-central1-a
2
dataproc-36c094d4-2796-4b03-85c9-7e9a90a99654-us
6 Jun 2017, 15:57:07
Running


Google Cloud Platform
clouderacluster
Compute Engine
VM instances
SHOW INFO PANEL
VM instances
Instance groups
Instance templates
Disks
Snapshots
Images
Filter by label or name
Columns
Name
Zone
Recommendation
Internal IP
External IP
Connect
datascience-m
us-central1-a
10.128.0.2
104.198.236.81
SSH
datascience-w-0
us-central1-a
10.128.0.3
130.211.171.250
SSH
datascience-w-1
us-central1-a
10.128.0.4
104.155.180.252
SSH

Lunch the Jupyter notebook

<<public ip>> :8123



Spark Program : Navigating Flight Data

```
>>> airline =  sc.textFile("gs ://<YOUR-BUCKET>/2008.csv")
>>> airline.take(2)
```

```
In [1]: airline =  sc.textFile("gs://imcinstitute/2008.csv")

In [2]: airline.take(2)

Out[2]: ['Year,Month,DayofMonth,DayOfWeek,DepTime,CRSDepTime,ArrTime,CRSArrTime,UniqueCarrier,FlightNum,TailNum,ActualElapsed
Time,CRSElapsedTime,AirTime,ArrDelay,DepDelay,Origin,Dest,Distance,TaxiIn,TaxiOut,Cancelled,CancellationCode,Diverte
d,CarrierDelay,WeatherDelay,NASDelay,SecurityDelay,LateAircraftDelay',
 '2008,1,3,4,2003,1955,2211,2225,WN,335,N712SW,128,150,116,-14,8,IAD,TPA,810,4,8,0,,0,NA,NA,NA,NA,NA']
```

Spark Program : Preparing Data

```
>>> header_line = airline.first()
>>> header_list = header_line.split(',')
>>> airline_no_header = airline.filter(lambda row: row !=
header_line)
>>> airline_no_header.first()
>>> def make_row(row):
...     row_list = row.split(',')
...     d = dict(zip(header_list,row_list))
...     return d
...
>>> airline_rows = airline_no_header.map(make_row)
>>> airline_rows.take(5)
```

In [3]:
```
header_line = airline.first()
header_list = header_line.split(',')
airline_no_header = airline.filter(lambda row: row != header_line)
airline_no_header.first()
def make_row(row):
    row_list = row.split(',')
    d = dict(zip(header_list,row_list))
    return d
airline_rows = airline_no_header.map(make_row)
airline_rows.take(5)
```

Out[3]:
```
[{'ActualElapsedTime': '128',
  'AirTime': '116',
  'ArrDelay': '-14',
  'ArrTime': '2211',
  'CRSArrTime': '2225',
  'CRSDepTime': '1955',
  'CRSElapsedTime': '150',
  'CancellationCode': '',
  'Cancelled': '0',
  'CarrierDelay': 'NA',
  'DayOfWeek': '4',
  'DayofMonth': '3',
  'DepDelay': '8',
  'DepTime': '2003',
  'Dest': 'TPA',
  'Distance': '810',
  'Diverted': '0',
  'FlightNum': '335',
  'LateAircraftDelay': 'NA',
```

Spark Program : Define convert_float function

```
>>> def convert_float(value):
...     try:
...         x = float(value)
...         return x
...     except ValueError:
...         return 0
...
>>>
```

```
In [4]: def convert_float(value):
            try:
                x = float(value)
                return x
            except ValueError:
                return 0
```

Spark Program : Finding best/worst airline

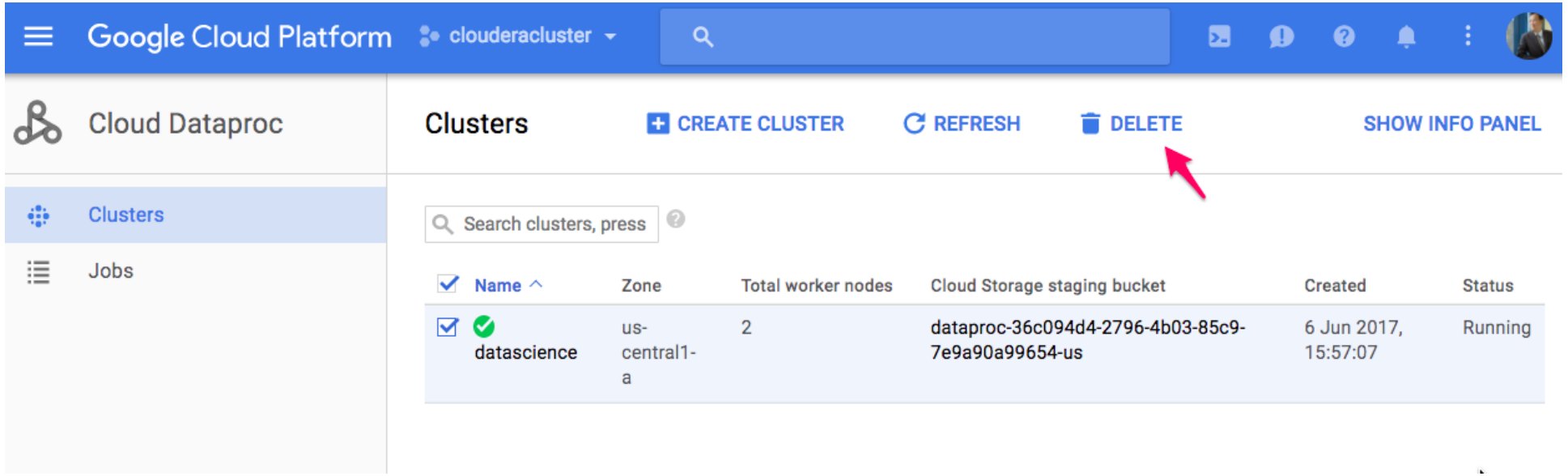
```
>>> carrier_rdd = airline_rows.map(lambda
row:(row['UniqueCarrier'],convert_float(row['ArrDelay'])))
>>> carrier_rdd.take(2)
```

```
In [5]: carrier_rdd = airline_rows.map(lambda row:(row['UniqueCarrier'],convert_float(row['ArrDelay'])))
        carrier_rdd.take(2)

Out[5]: [('WN', -14.0), ('WN', 2.0)]
```

Spark Program : Finding best/worst airlines

```
>>> mean_delays_dest =
carrier_rdd.groupByKey().mapValues(lambda
sum(delays)/len(delays)
>>> mean_delays_dest.sortBy(lambda t:t[1],
ascending=True).take(10)
>>> mean_delays_dest.sortBy(lambda t:t[1],
ascending=False).take(10)
```

```
In [11]: mean_delays_dest = carrier_rdd.groupByKey().mapValues(lambda delays: sum(delays.data)/len(delays.data))

In [12]: mean_delays_dest.sortBy(lambda t:t[1], ascending=True).take(10)

Out[12]: [(u'AQ', -2.8708974358974357),
          (u'HA', 1.2518519716624075),
          (u'US', 2.800998260539828),
          (u'9E', 3.987490846961191),
          (u'AS', 4.721360405553864),
          (u'WN', 5.115703380225903),
          (u'F9', 6.084135669681085),
          (u'OO', 6.43893863978179),
          (u'NW', 7.293465879672776),
          (u'DL', 7.716164635751918)]

In [13]: mean_delays_dest.sortBy(lambda t:t[1], ascending=False).take(10)

Out[13]: [(u'AA', 12.202853434950445),
          (u'OH', 11.404110178283158),
          (u'YV', 11.322566979170753),
          (u'UA', 11.001550560048052),
          (u'B6', 10.859381613638567),
          (u'CO', 10.809820575966226),
          (u'XE', 10.320298523403915),
          (u'EV', 10.00033146217589),
          (u'MQ', 9.496970610952266),
          (u'FL', 8.988157472371256)]
```

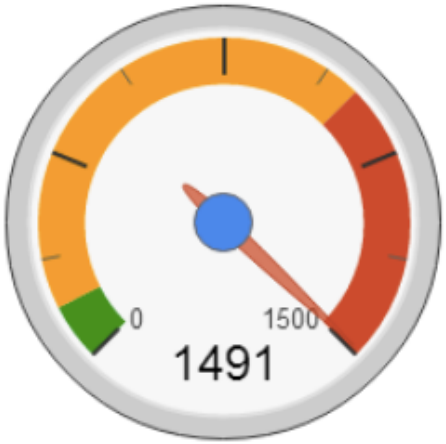

Google Cloud Platform
clouderacluster
Cloud Dataproc
Clusters
Jobs
Clusters
CREATE CLUSTER
REFRESH
DELETE
SHOW INFO PANEL
Search clusters, press
Name
Zone
Total worker nodes
Cloud Storage staging bucket
Created
Status
datascience
us-central1-a
2
dataproc-36c094d4-2796-4b03-85c9-7e9a90a99654-us
6 Jun 2017, 15:57:07
Running

Google BigQuery

Google BigQuery is a fully-managed and cloud-based interactive query service for massive datasets. It's the externalization of Dremel, one of Google's core technologies

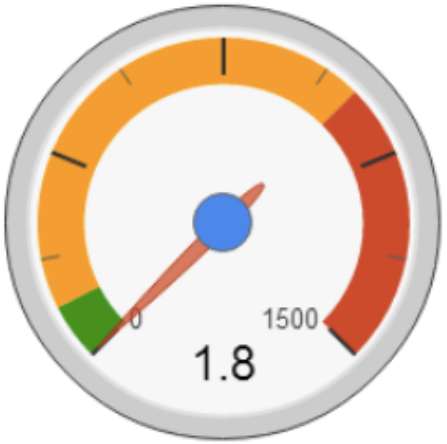
Why Google BigQuery ?

= 1.4 TB



Hadoop
(with Hive)

Amazon
Redshift



Google
BigQuery

On an average its within 8-10 seconds !!

GDG DevFest
2014 Season

Inside Google BigQuery

- BigQuery is based on **Dremel**, a technology pioneered by Google & extensively used within.
- It used **Columnar storage** & **multi-level execution trees** to achieve interactive performance for queries against multi-terabyte datasets.
- BigQuery's performance advantage comes from its parallel processing architecture.
- The query is processed by thousands of servers in a multi-level execution tree structure, with the final results aggregated at the root. BigQuery stores the data in a columnar format so that only data from the columns being queried are real.
- All this & more is now available as a publicly available service for any business or developer to use. This release made it possible for those outside of Google to utilize the power of Dremel for their Big Data processing requirements.

Source: Exploring BigData with Google BigQuery; Dharmesh Vaya

Features & Components

Features:

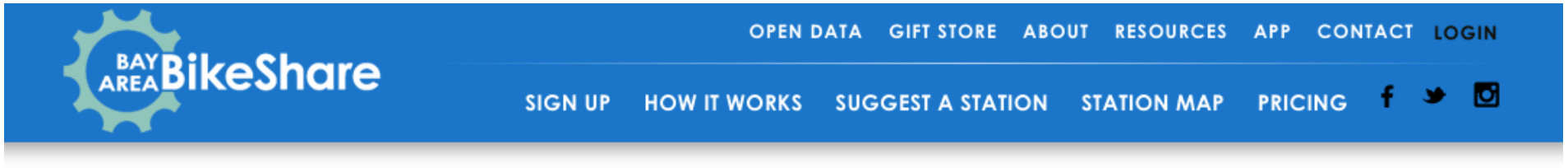
- Web GUI for BigQuery
- Affordable
- Run in Background
- Easy Data Importation
- Flexible (Addition of Columns, Native Support For Timestamp Type Of Data)
- REST API Support
- More than just Standard SQL

Components:

- Project
- Tables
- DataSets
- Jobs

Lab V：Query Bike data

Using Google BigQuery



OPEN DATA

Here you'll find Bay Area Bike Share's trip data for public use. So whether you're a designer, developer, or just plan curious, feel free to download it and bring it to life!

YEAR 1 DATA

(August 2013 - August 2014)

YEAR 2 DATA

(September 2014 - August 2015)

THE DATA

Each trip is anonymized and includes:

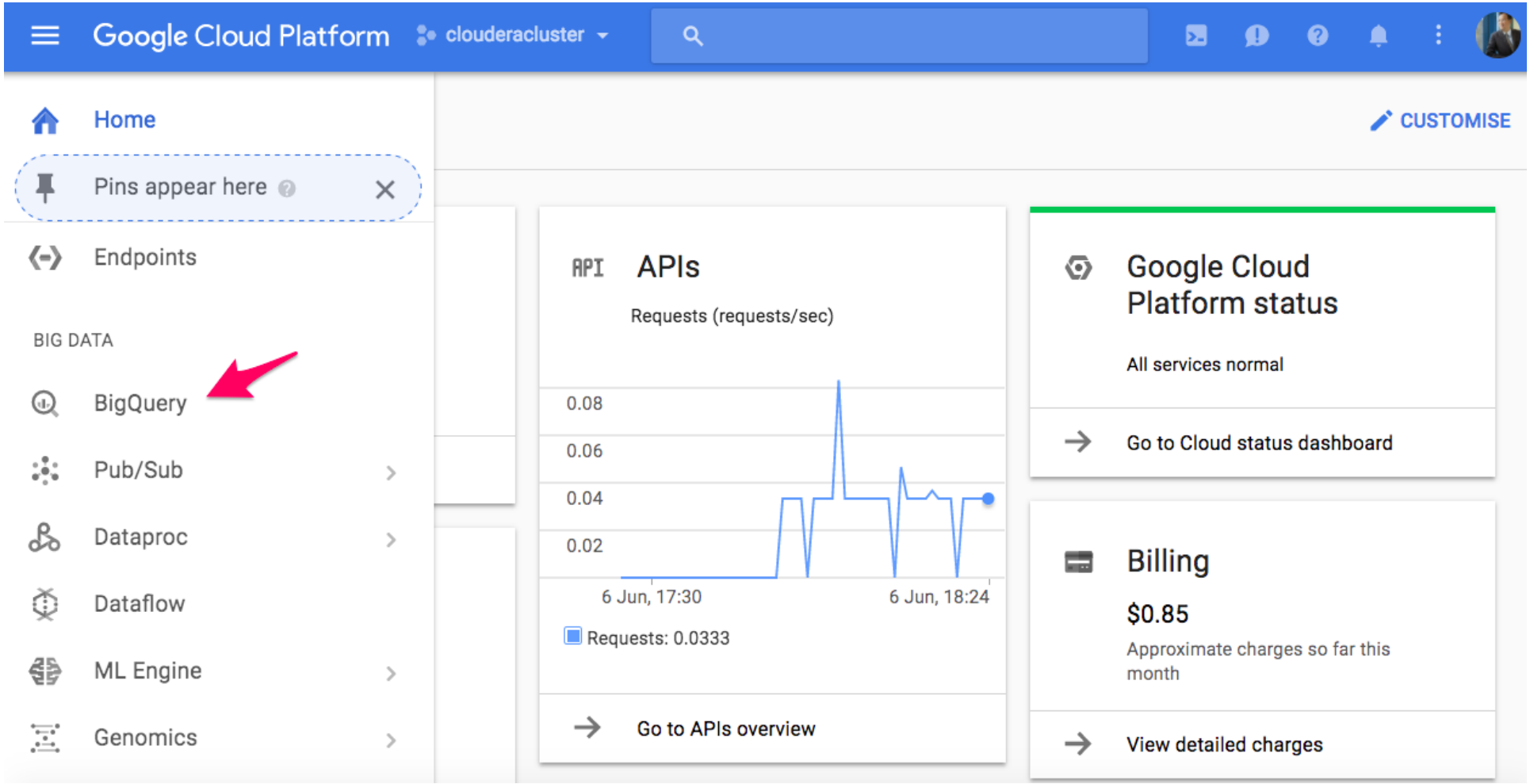
- Bike number
- Trip start day and time
- Trip end day and time

Preparing a bike data from Cloud shell

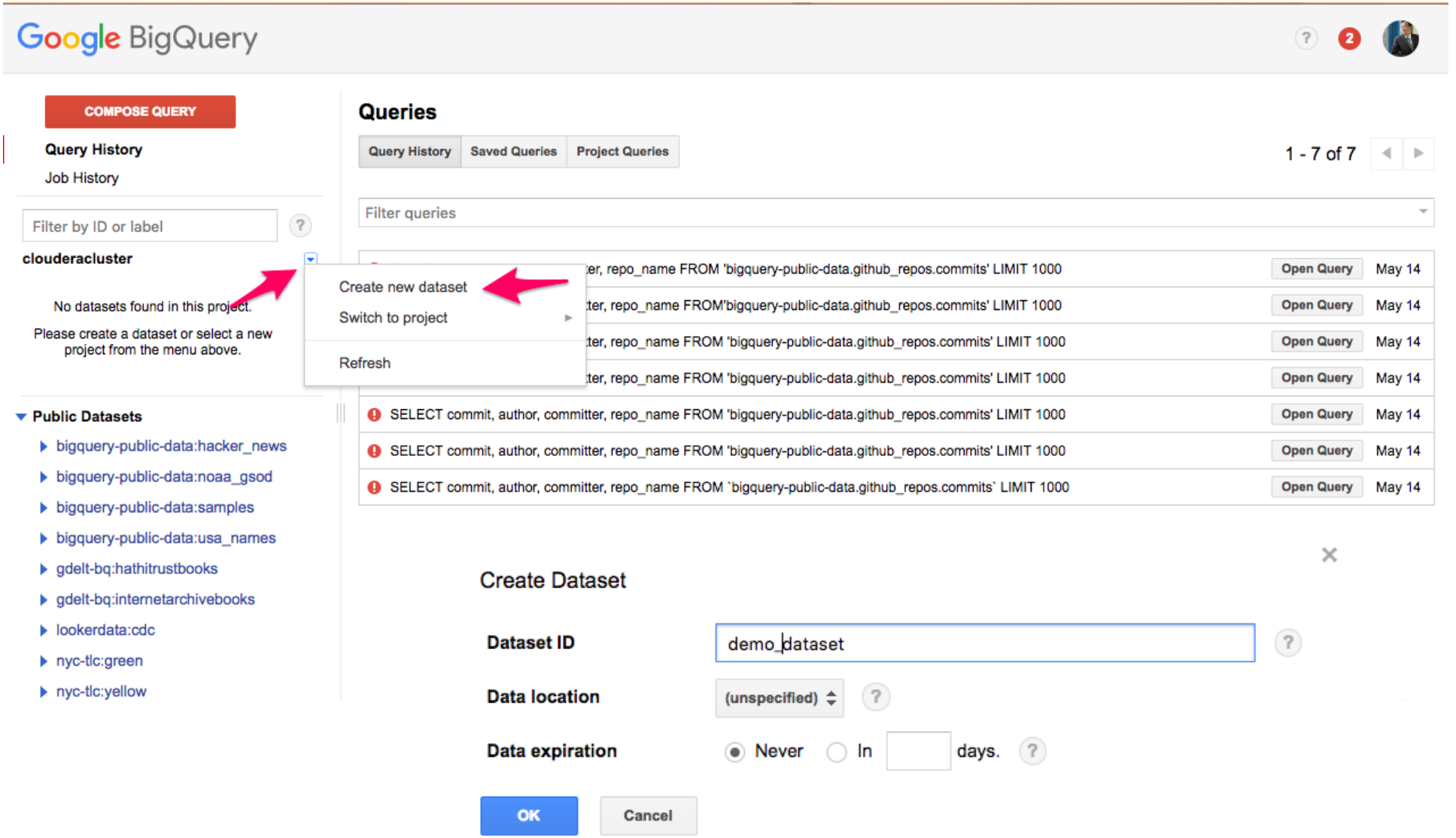
```
$ wget https://s3.amazonaws.com/babs-open-data/babs_open_data_year_1.zip

$ unzip babs_open_data_year_1.zip
$ cd 201402_babs_open_data/
$ gsutil cp 201402_trip_data.csv  gs://<YOUR-BUCKET>/
```

```
thanachart@cloudera cluster-164402:~$ ls
201402_babs_open_data  babs_open_data_year_1.zip  pg2600.txt              training-data-analyst
201408_babs_open_data  data-science-on-gcp        README-cloudshell.txt
thanachart@cloudera cluster-164402:~$  cd 201402_babs_open_data/
thanachart@cloudera cluster-164402:~/201402_babs_open_data$ gsutil cp 201402_trip_data.csv  gs://imcinstitute/
Copying file://201402_trip_data.csv [Content-Type=text/csv]...
\ [1 files][ 16.4 MiB/ 16.4 MiB]
Operation completed over 1 objects/16.4 MiB.
thanachart@cloudera cluster-164402:~/201402_babs_open_data$
```

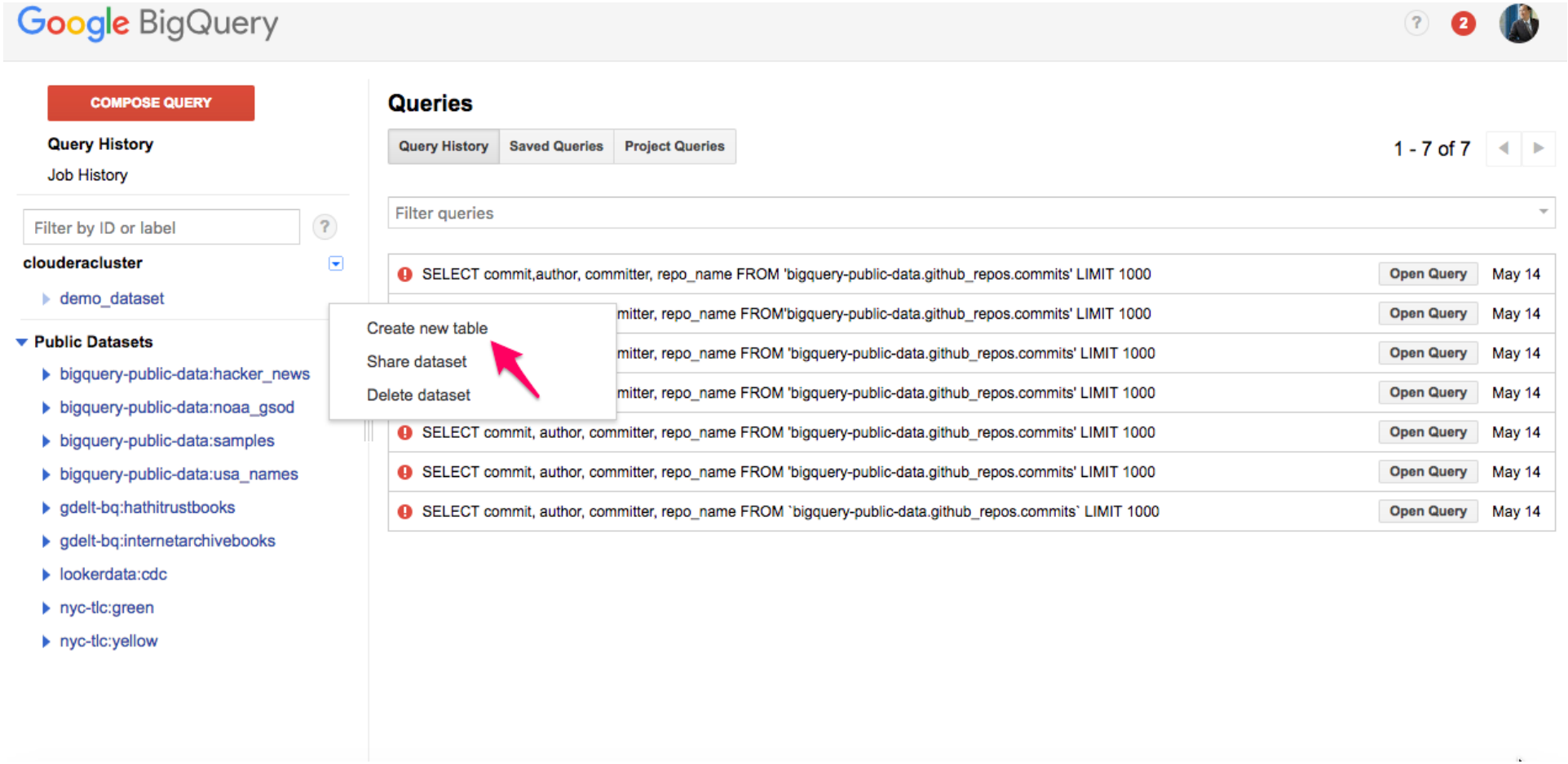
Launch Google BigQuery



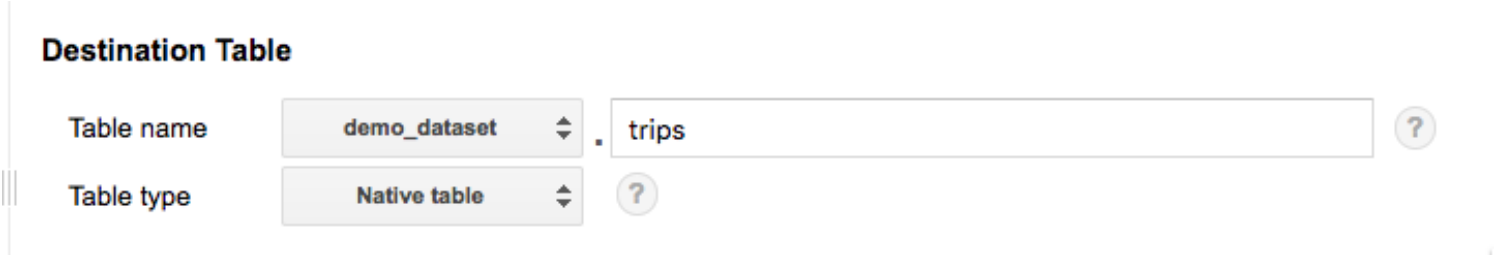
Create new dataset



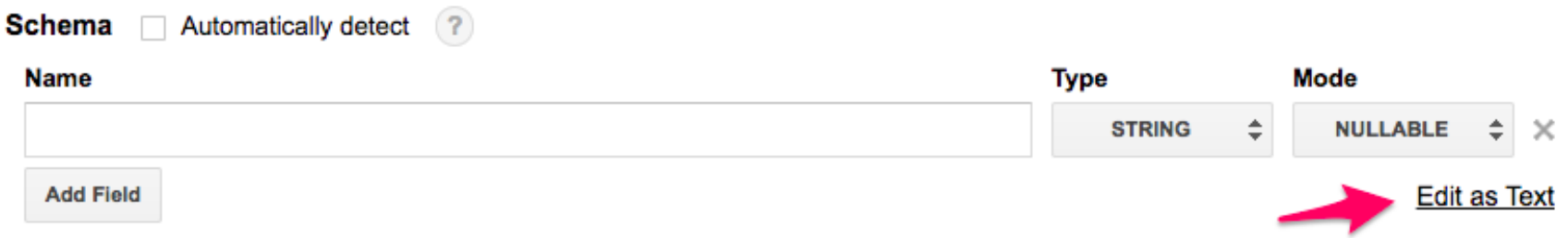
Create new table froma the newly created dataset

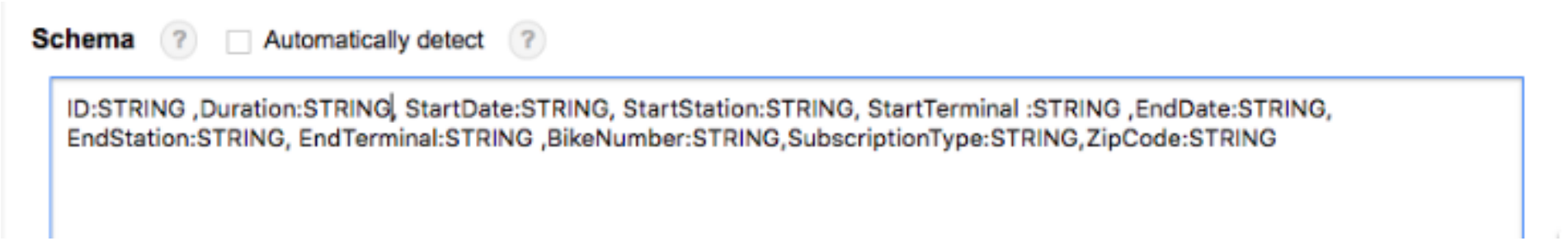


Select Source Data & Destination Table



Select Schema as follow





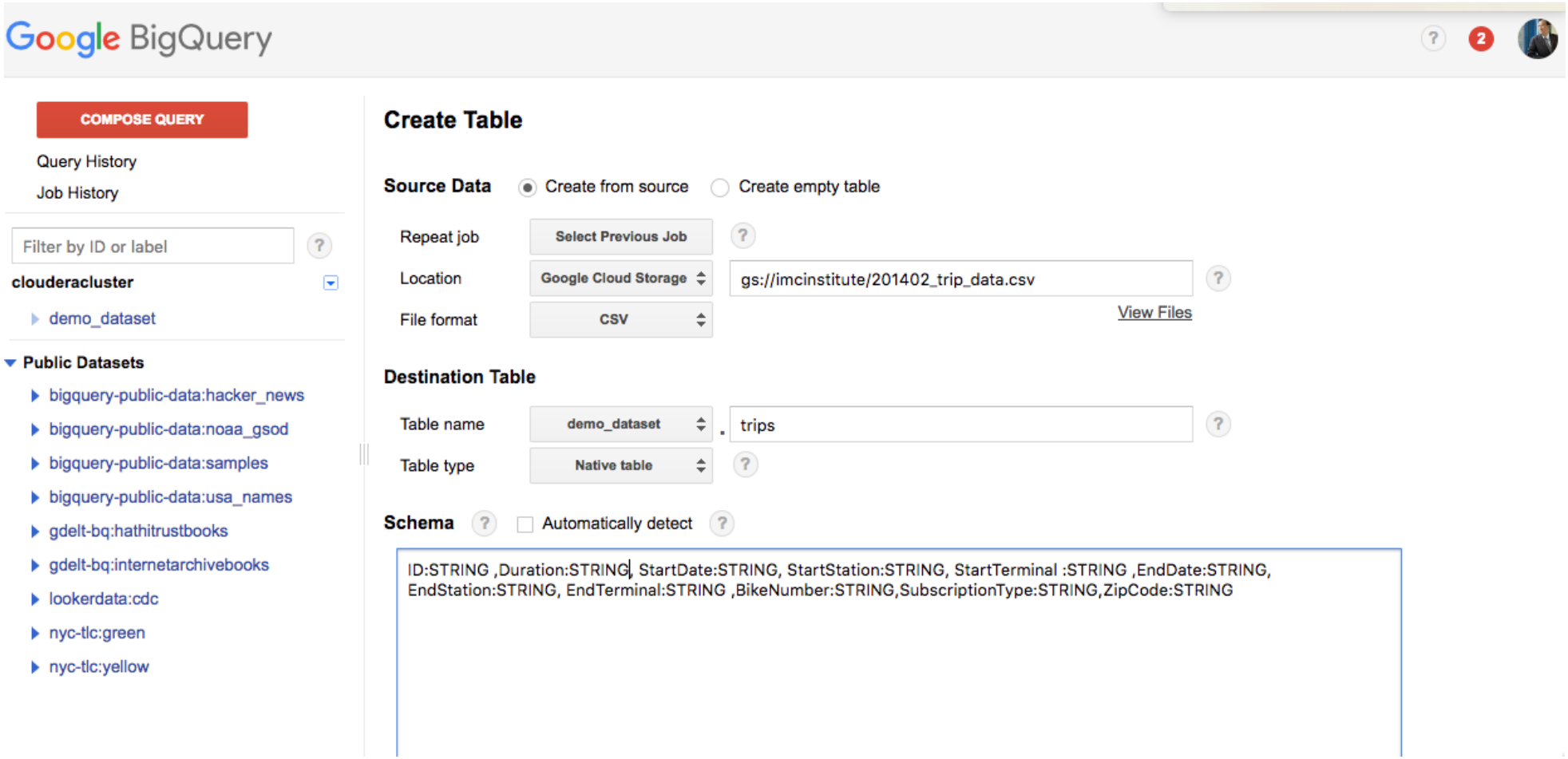
id:STRING,

Duration:STRING,StartDate:STRING,StartStation:STRING,StartTerminal:STRING,EndDate:

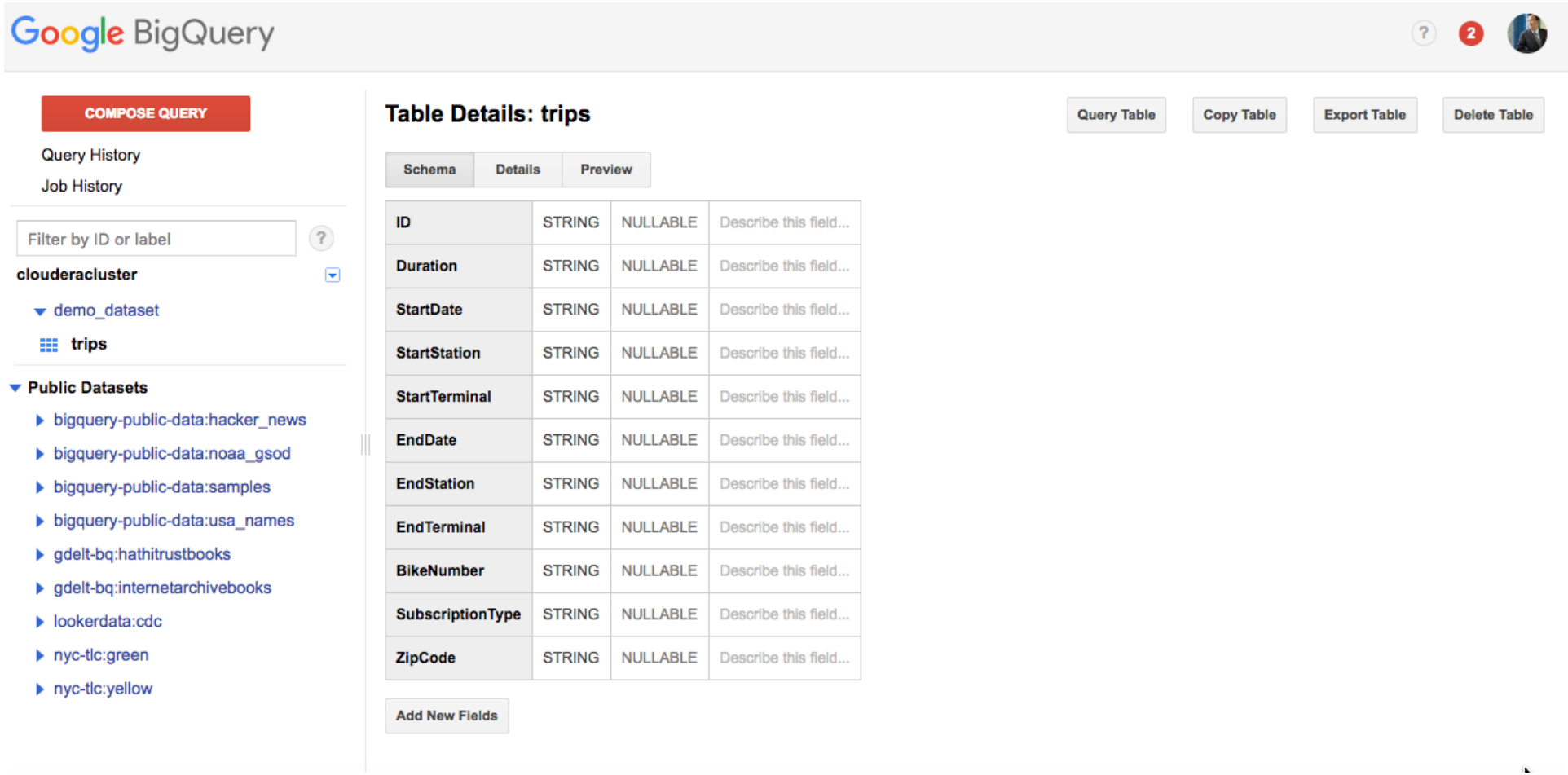
STRING,EndStation:STRING,EndTerminal:STRING,BikeNumber:STRING,SubsriptionType:STRING,

ZipCode:STRING

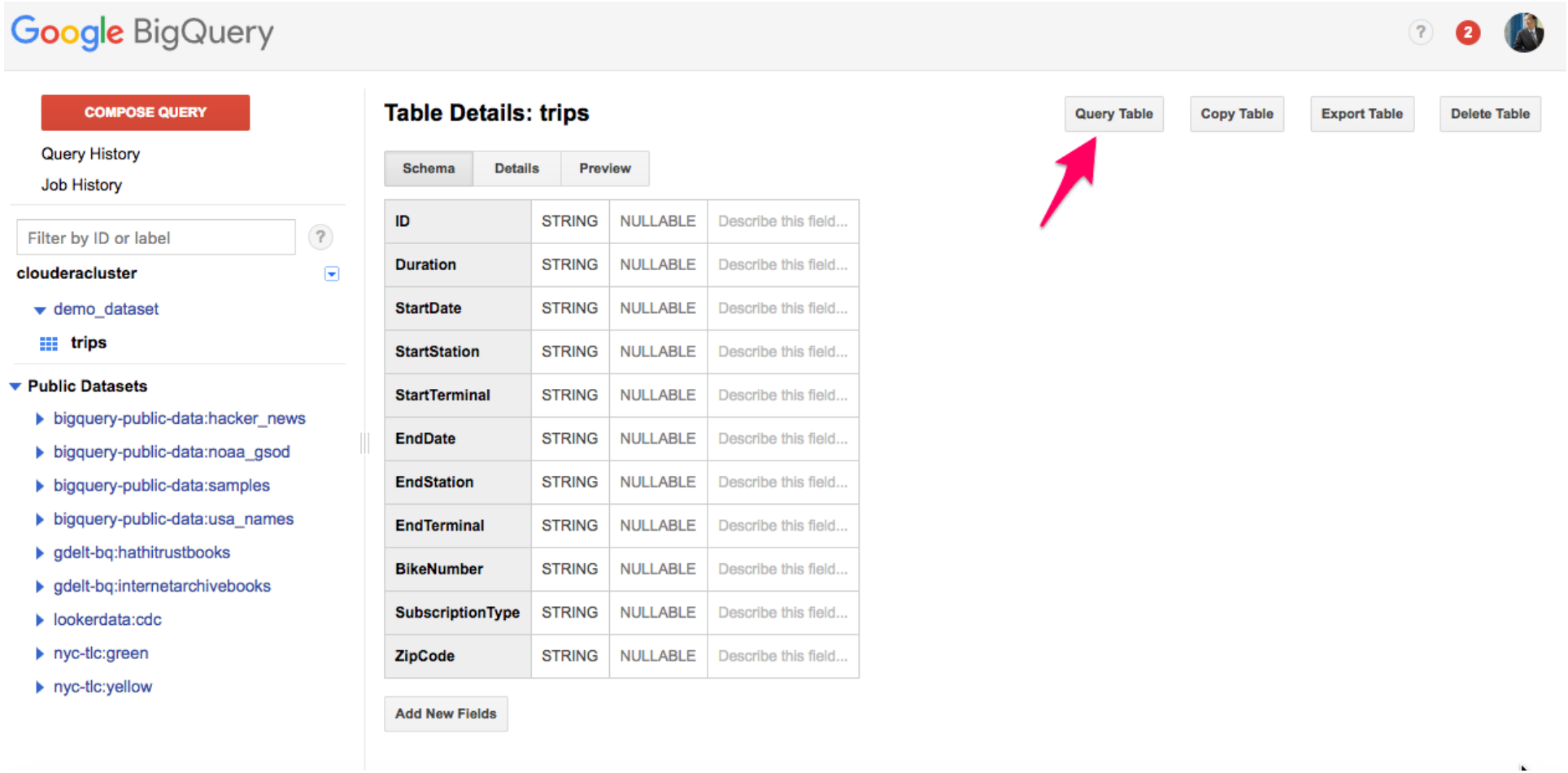
Click Create Table



View Trips Table

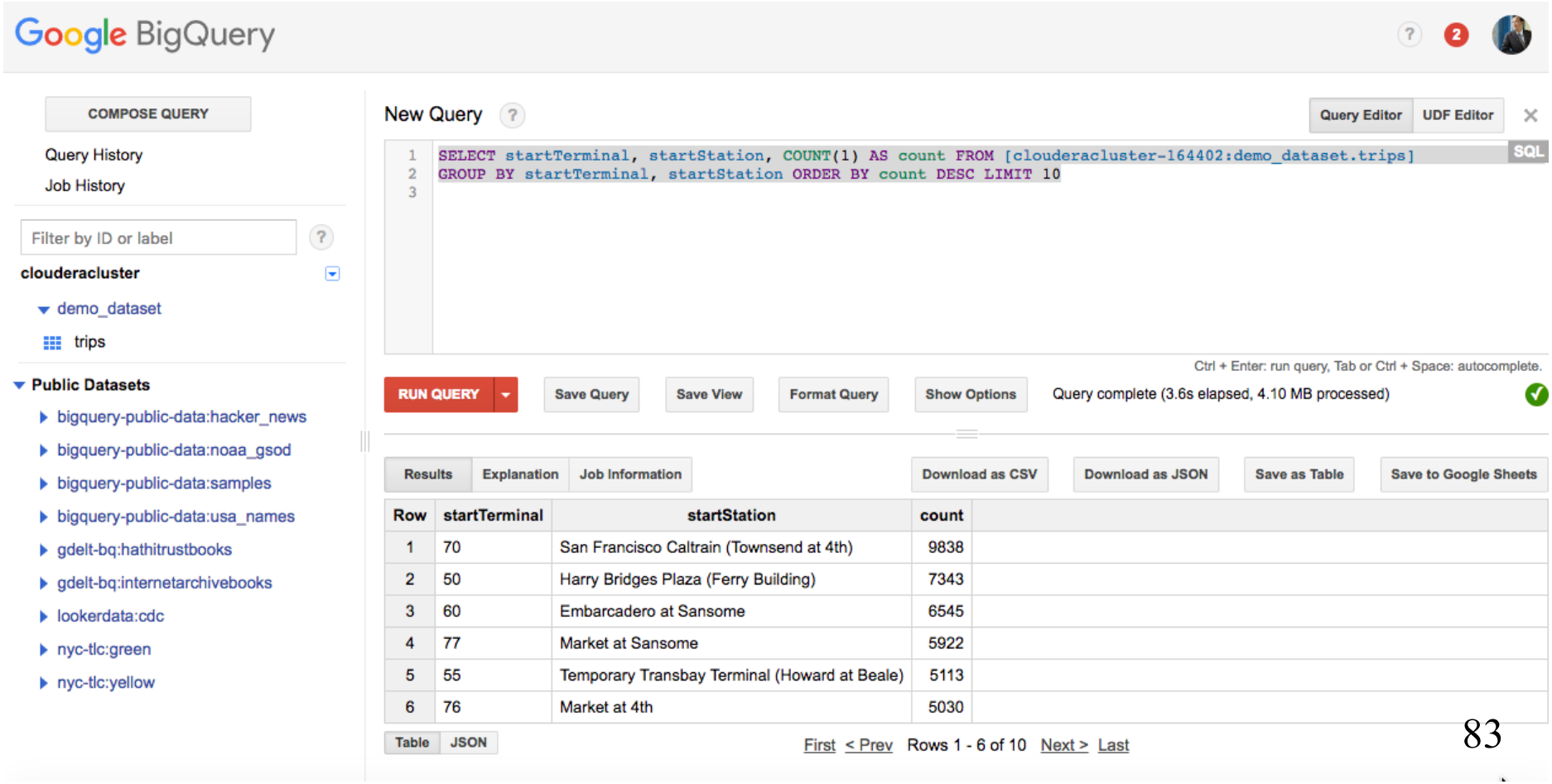


Query Trips Table



Find the top 10 most popular start stations based on the trip data

```
SELECT startTerminal, startStation, COUNT(1) AS count FROM
[ <<Google Cloud Shell>> :demo_dataset.trips]
GROUP BY startTerminal, startStation ORDER BY count DESC LIMIT 10
```



Google Cloud DataLab

Cloud Datalab is a powerful interactive tool created to explore, analyze, transform and visualize data and build machine learning models on Google Cloud Platform. It runs on Google Compute Engine and connects to multiple cloud services easily so you can focus on your data science tasks.

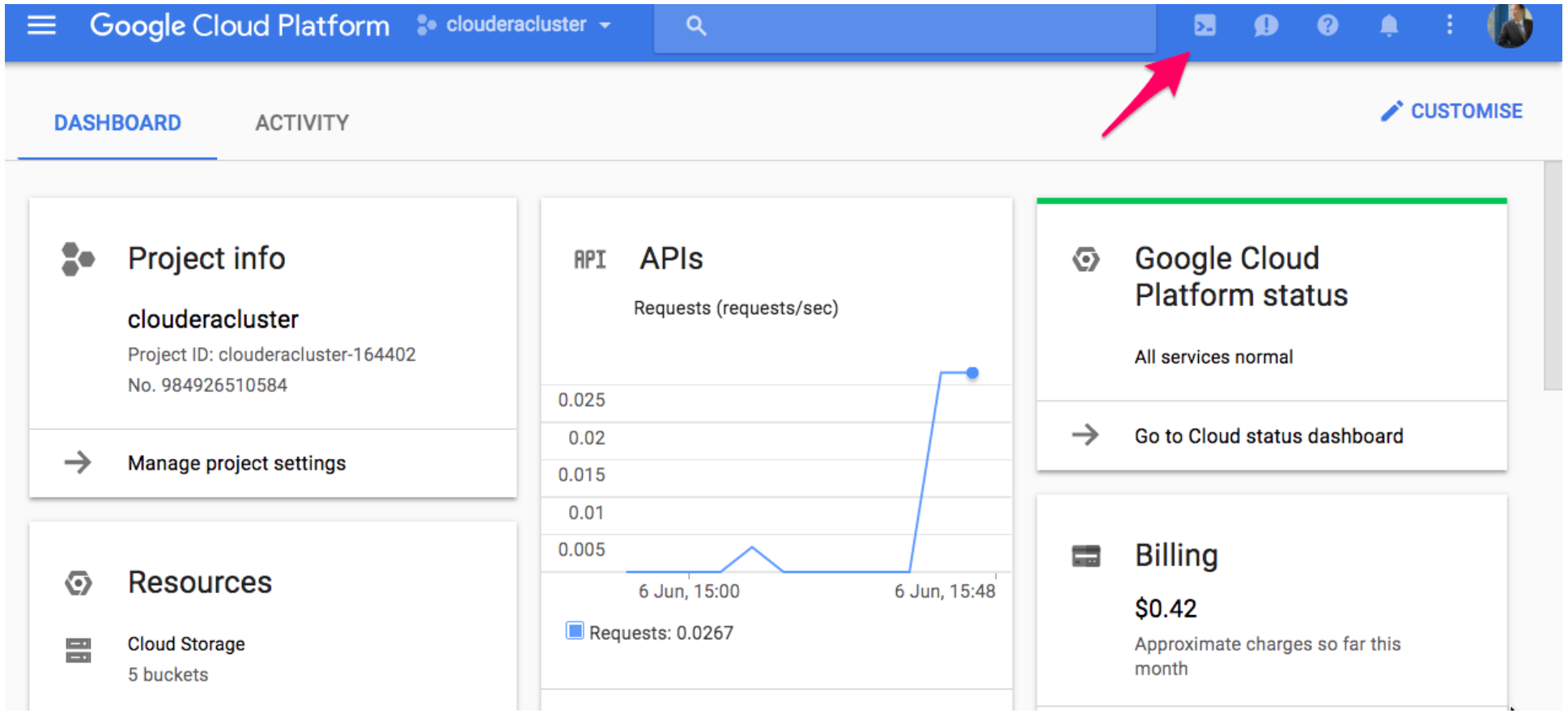
Integrated & Open Source

Cloud Datalab is **built on Jupyter (formerly IPython)**, which boasts a thriving ecosystem of modules and a robust knowledge base. Cloud Datalab **enables analysis of your data on Google BigQuery, Cloud Machine Learning Engine, Google Compute Engine, and Google Cloud Storage** using Python, SQL, and JavaScript (for BigQuery user-defined functions).

- Cloud Datalab simplifies data processing with Cloud BigQuery, Cloud Machine Learning Engine, Cloud Storage.
- Supports Python, SQL, and JavaScript
- Combines code, documentation, results, and visualizations together in an intuitive notebook format.
- Supports TensorFlow-based deep ML models in addition to scikit-learn.
- Based on Jupyter (formerly IPython) so you can use a large number of existing packages for statistics, machine learning etc..

Lab VI：Launch Cloud DataLab

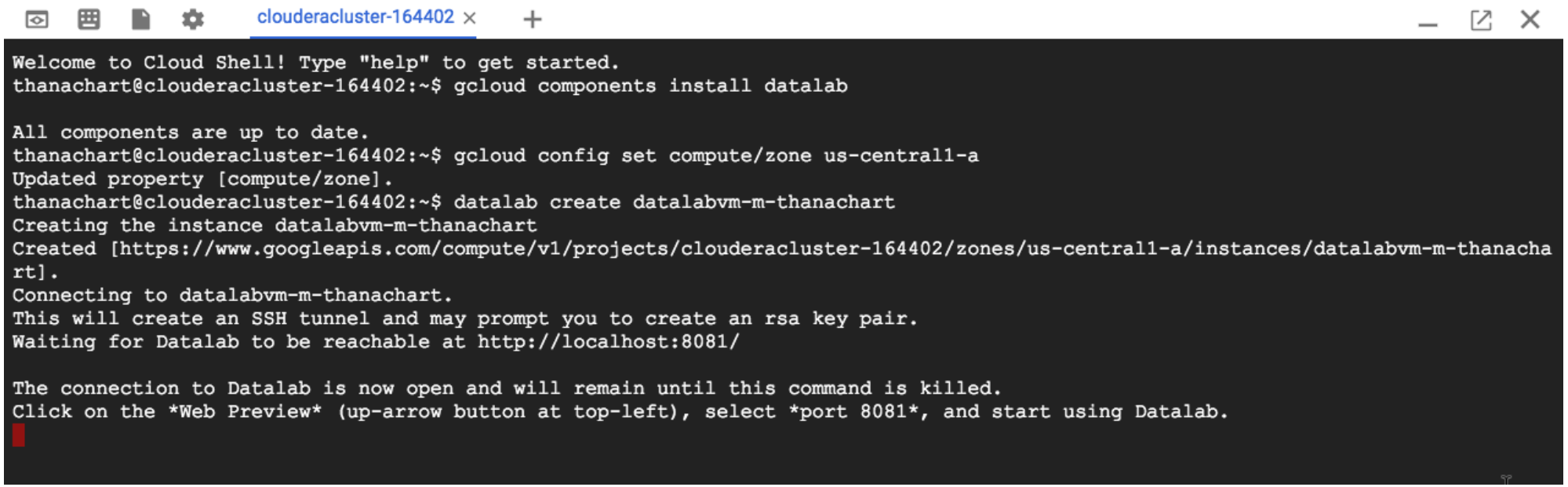
Launch Cloud Shell



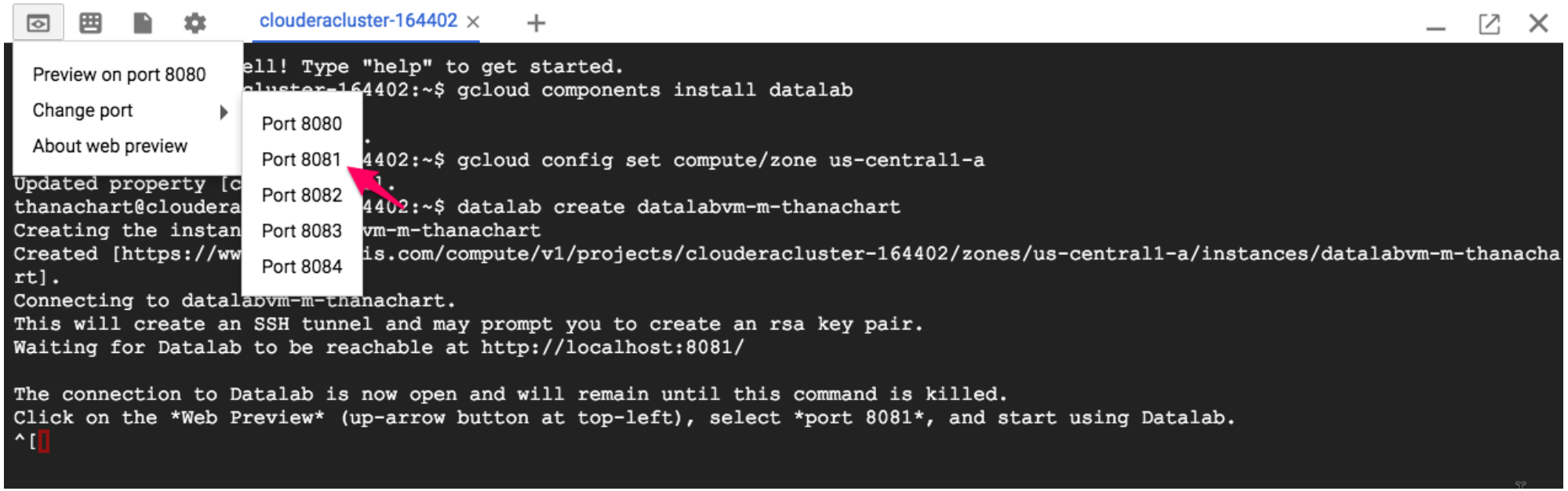
Launch DataLab

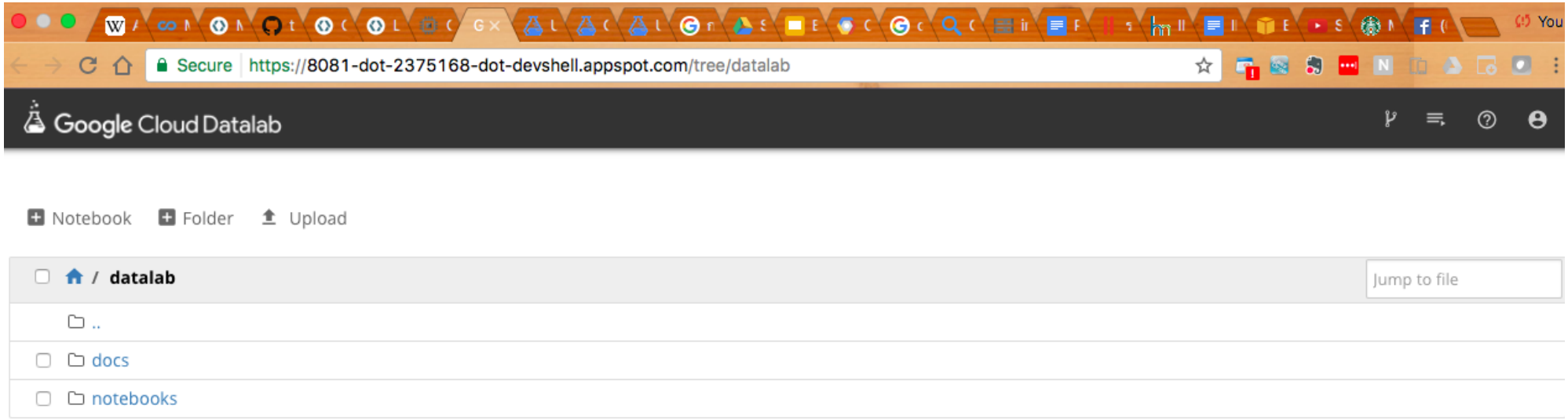
Docker on Compute Engine
Use GCP for processing
Notebooks on GCE disk
CloudShell ssh tunnel

```
$ gcloud components install datalab
$ gcloud config set compute/zone <ZONE>
$ datalab create datalabvm-<USER>

#This will take several minutes. Wait for the message
"You can now connect to Datalab at
http://localhost:8081/"
```

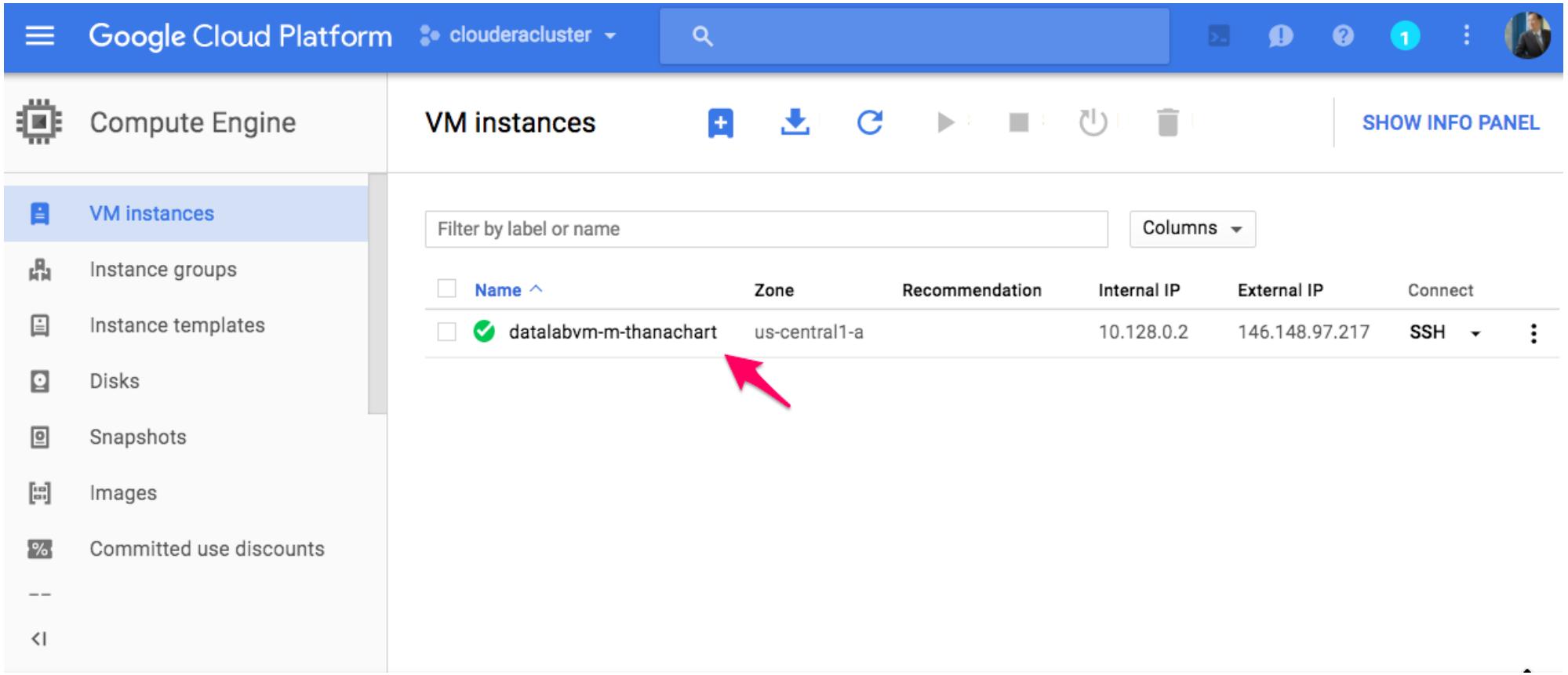

clouderacluster-164402
Welcome to Cloud Shell! Type "help" to get started.
thanachart@clouderacluster-164402:~$ gcloud components install datalab
All components are up to date.
thanachart@clouderacluster-164402:~$ gcloud config set compute/zone us-central1-a
Updated property [compute/zone].
thanachart@clouderacluster-164402:~$ datalab create datalabvm-m-thanachart
Creating the instance datalabvm-m-thanachart
Created [https://www.googleapis.com/compute/v1/projects/clouderacluster-164402/zones/us-central1-a/instances/datalabvm-m-thanacha
rt].
Connecting to datalabvm-m-thanachart.
This will create an SSH tunnel and may prompt you to create an rsa key pair.
Waiting for Datalab to be reachable at http://localhost:8081/
The connection to Datalab is now open and will remain until this command is killed.
Click on the *Web Preview* (up-arrow button at top-left), select *port 8081*, and start using Datalab.

Launch Data Lab port 8081




Secure
https://8081-dot-2375168-dot-devshell.appspot.com/tree/datalab
Google Cloud Datalab
Notebook
Folder
Upload
/ datalab
Jump to file
..
docs
notebooks

Viewing running VM instance



Reconnecting to Datalab

If the datalab command terminates (for example, if the laptop goes to sleep), reconnect to the VM using:

```
datalab connect datalabvm-<USER>
```

Cleanup (important)

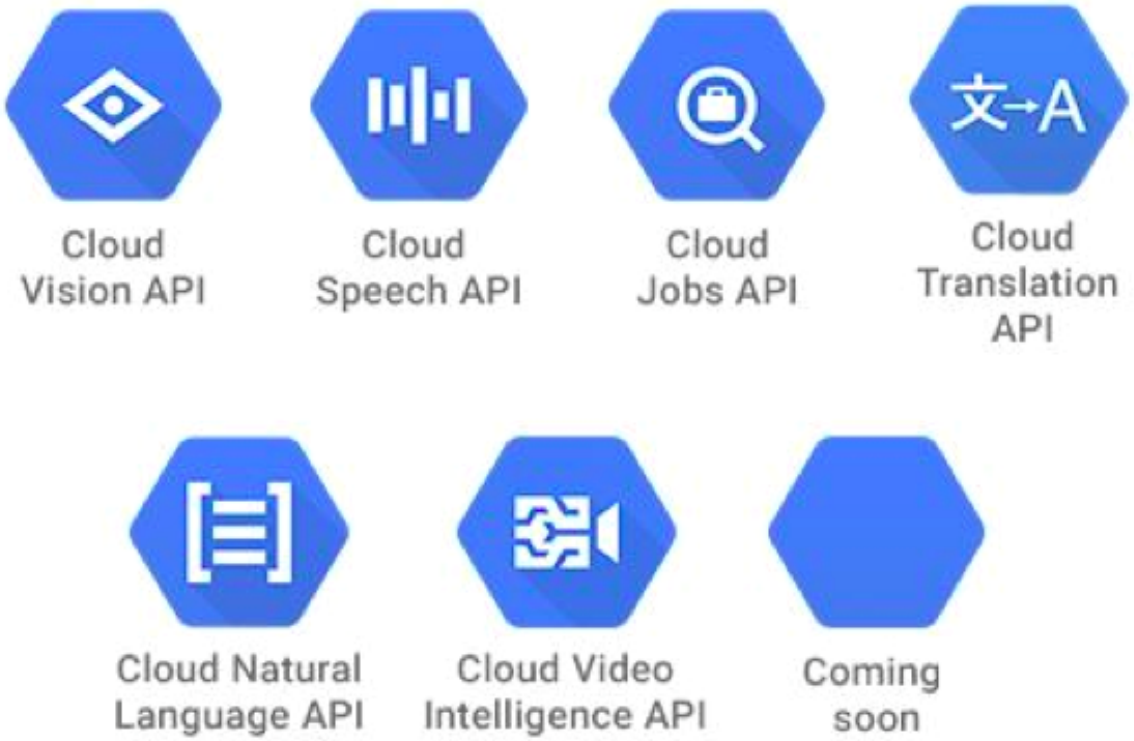
When you are done using the Datalab VM, delete the instance using:

```
datalab delete datalabvm-<USER>
```

Google Machine Learning APIs

Google Cloud Machine Learning provides modern machine learning services, with pre-trained models and a service to generate your own tailored models. Major Google applications use Cloud Machine Learning, including Photos (image search), the Google app (voice search), Translate, and Inbox (Smart Reply).

Ready to use Machine Learning APIS



Cloud Vision APIs

- Label / Face Detection
- Image Attributes
- Landmark Detection
- Logo Detection
- OCR Detection
- Safe Search Detection

Cloud Speech API

- Audio → Text
- 80 languages & variants
- Audio uploaded on request and integrate

  GCS
- Result Streaming supported
- Noisy Environments & Context

Cloud Natural Language API

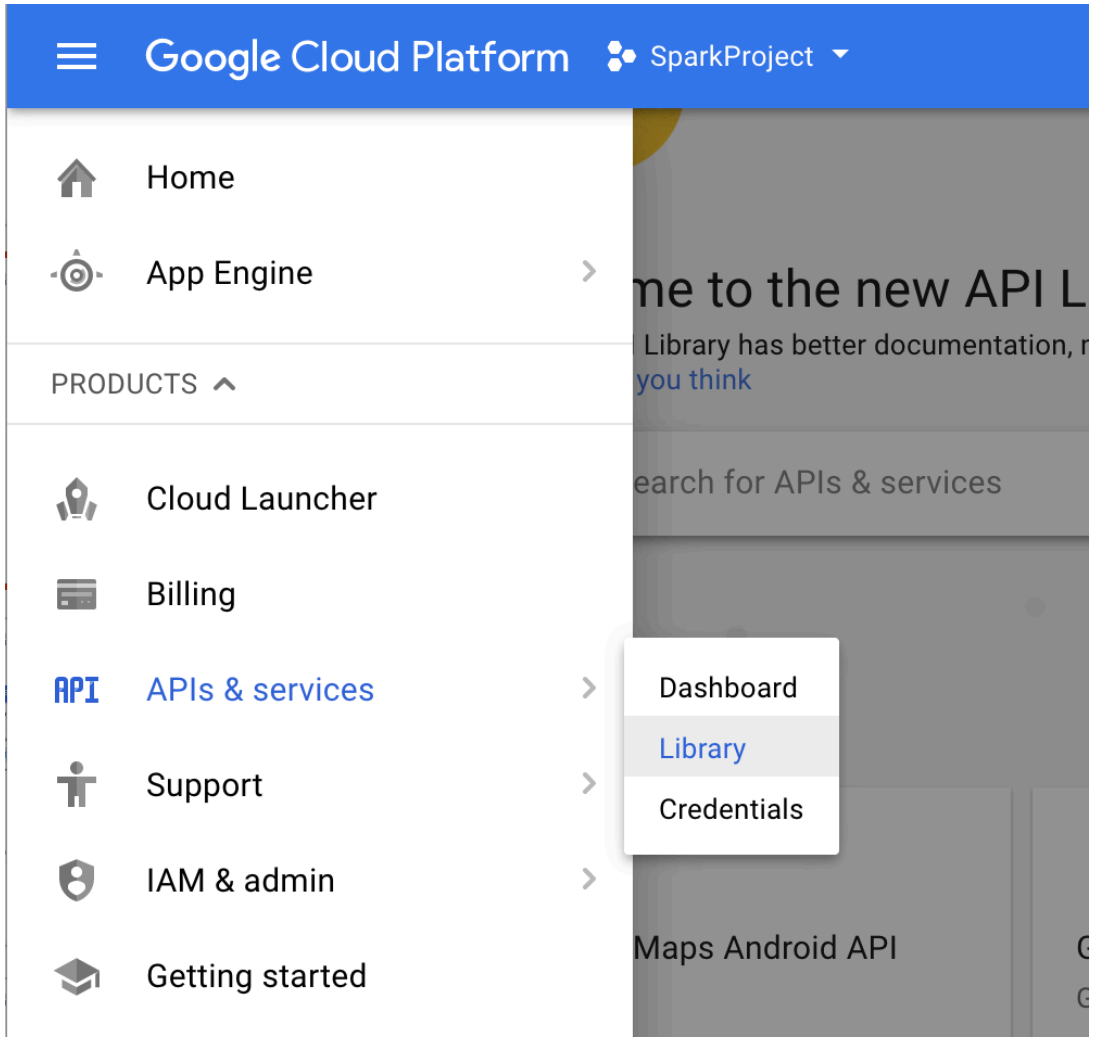
- Syntax Analysis
- Entity Recognition
- Sentiment Analysis
- Use in combination with Cloud Speech and Vision API (OCR)

Cloud Translation API

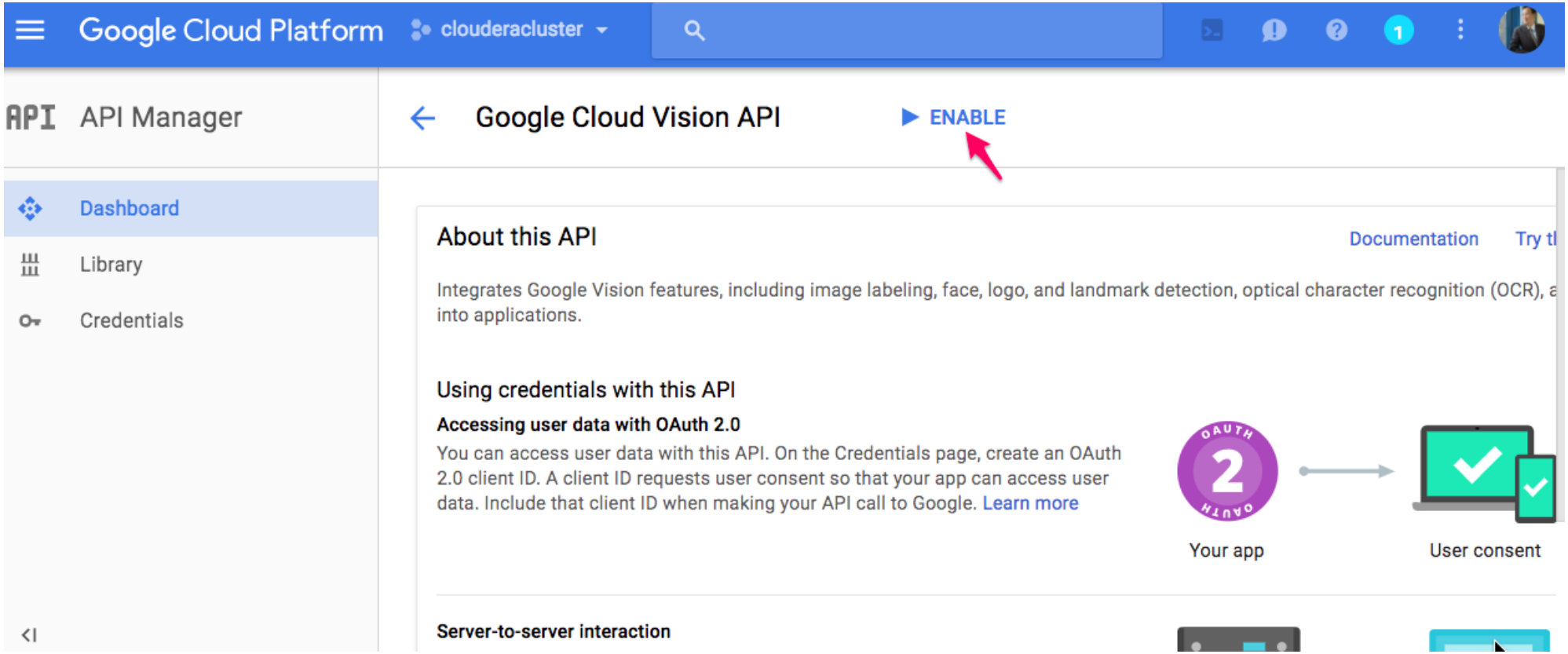
- Translate Many Languages
- Language Detection
- Supports more than 100 languages
- Learning continuously via Logs Analysis and Human Translation

Lab VII：Using ML APIs

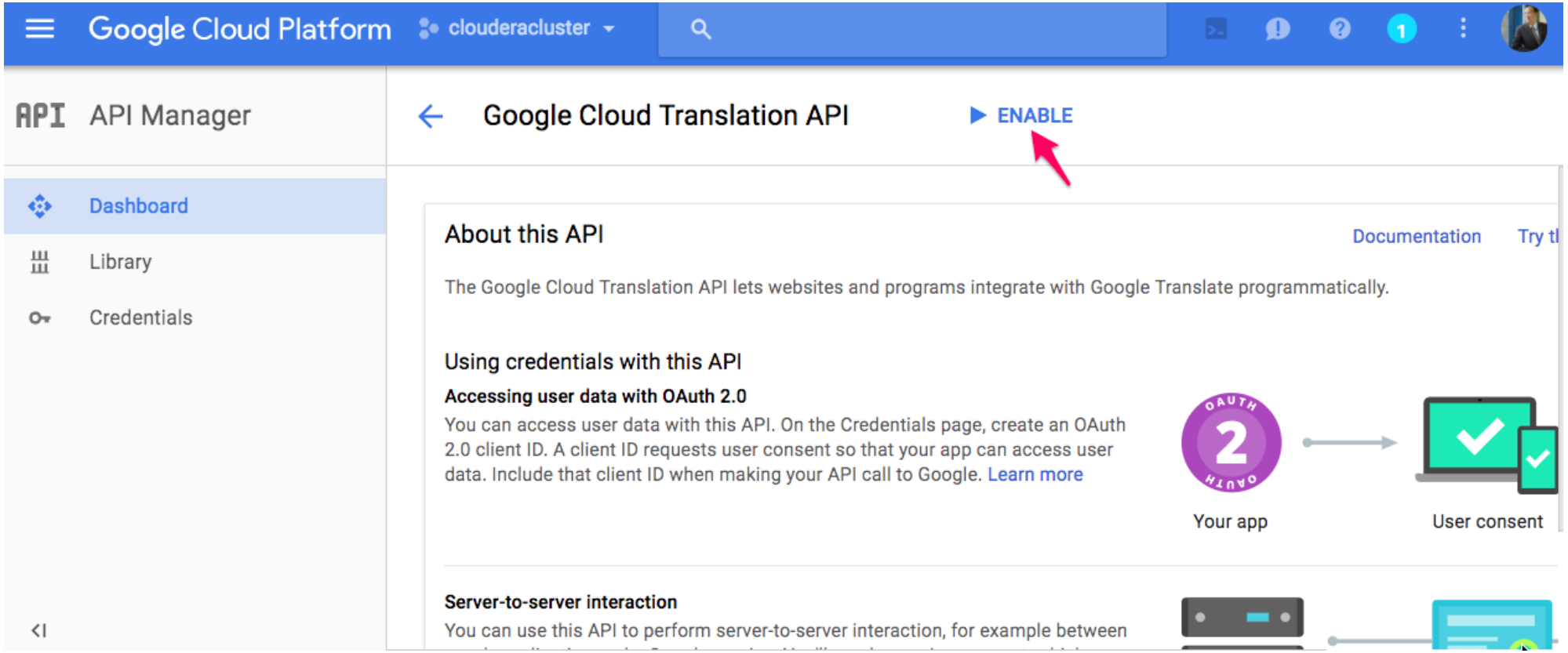
Select APIs & services -> Library



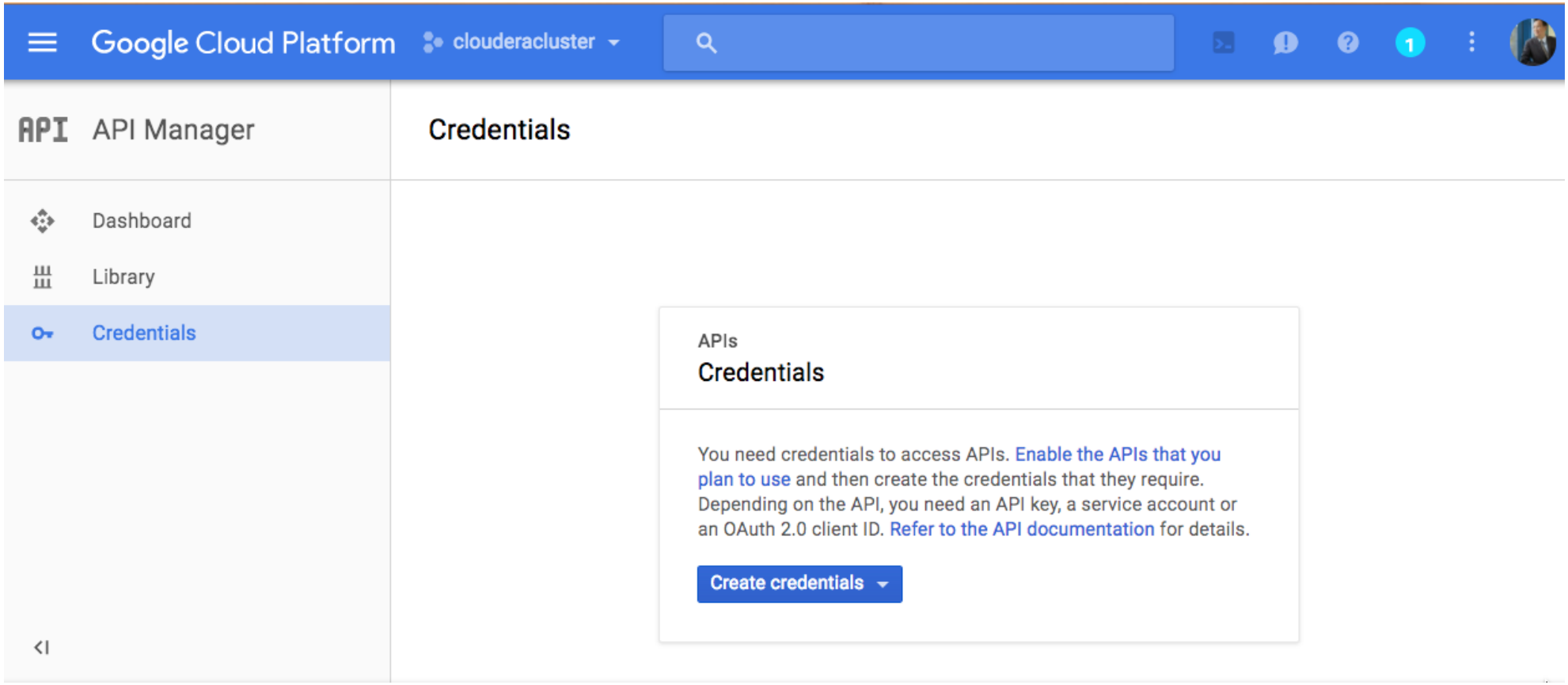
Enable Vision API



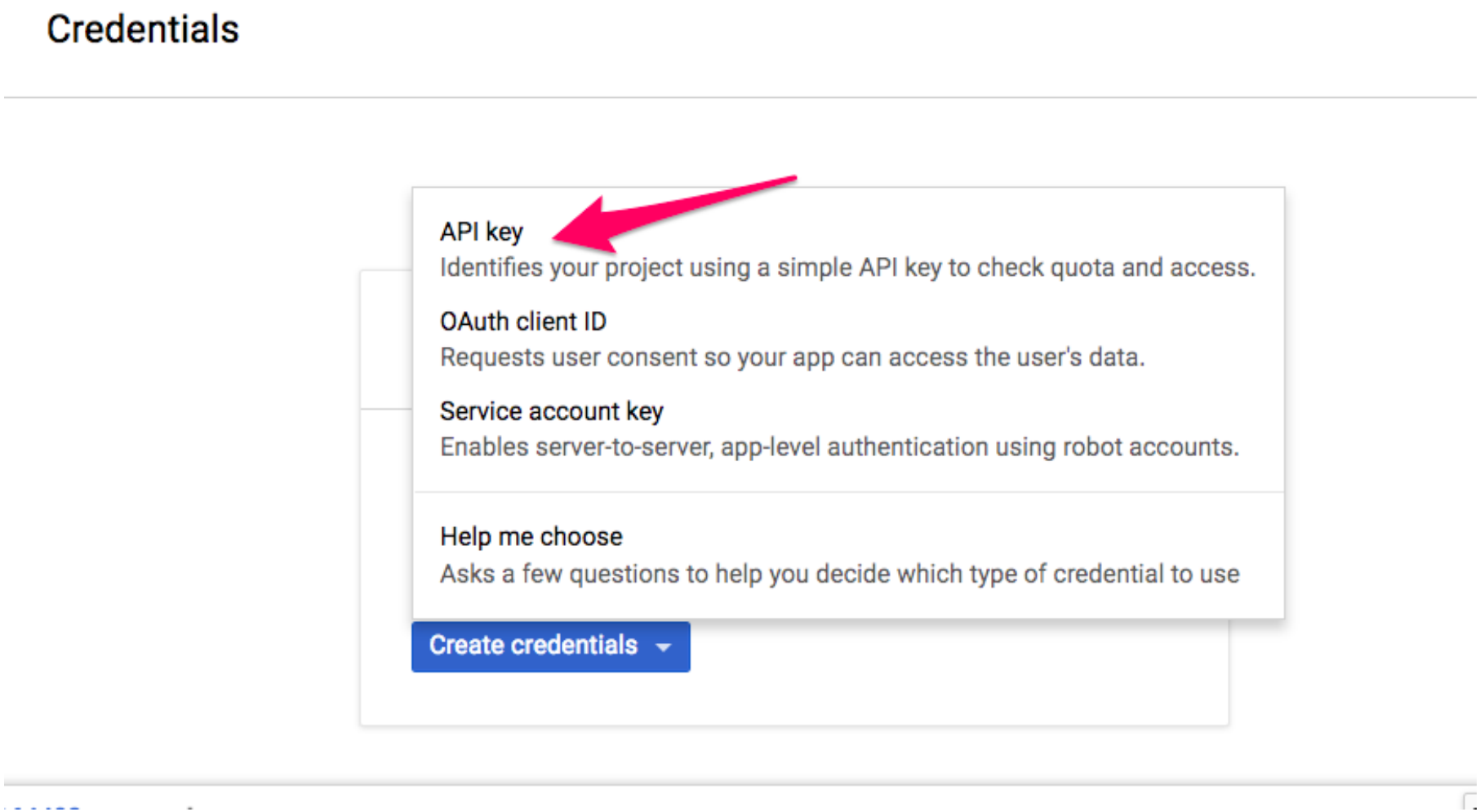
Enable Translation API



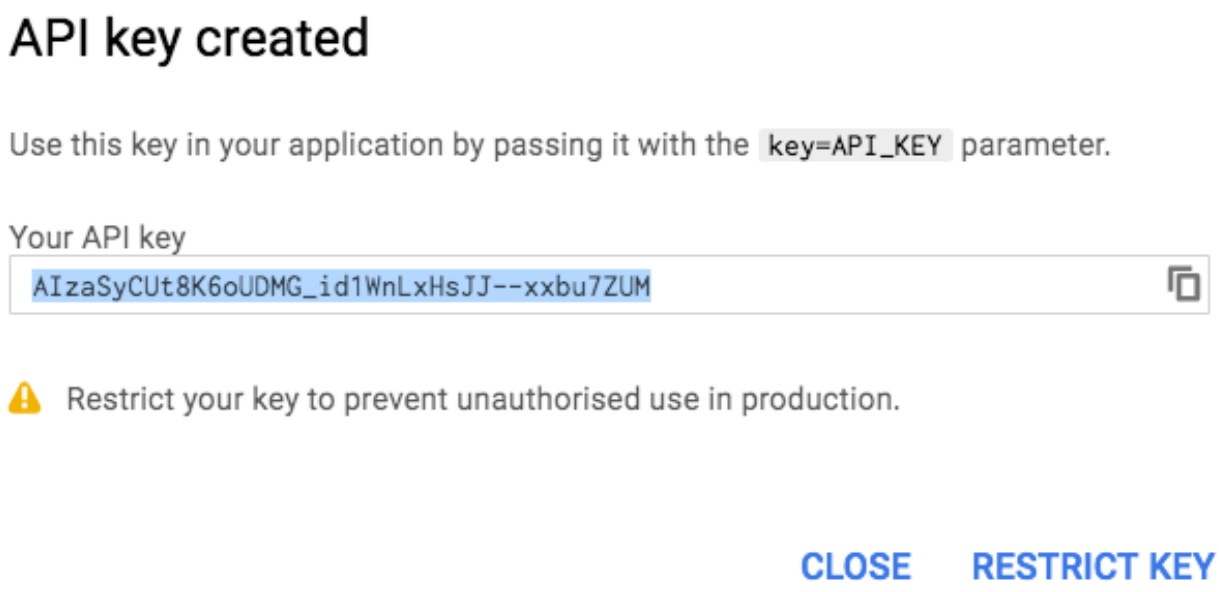
Get API key: Select Credential



Click Create Credential, then API Key



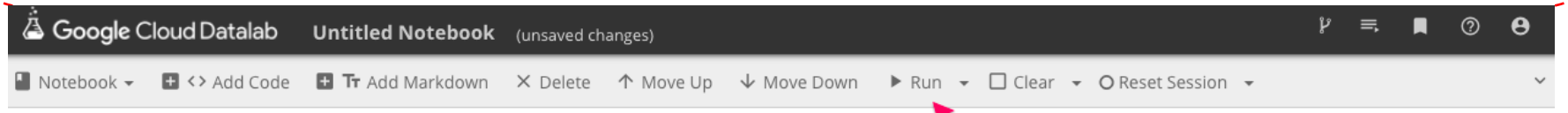
Copy API Key


API key created
Use this key in your application by passing it with the key=API_KEY parameter.
Your API key
AIzaSyCUt8K6oUDMG_id1WnLxHsJJ--xxbu7ZUM
Restrict your key to prevent unauthorised use in production.
CLOSE
RESTRICT KEY

Launch a new notebook

install the Python package

```
APIKEY="AIzaSyCUt8K6oUDMG_id1WnLxHsJJ--xxbu7ZUM"
# Replace with your API key
!pip install --upgrade google-api-python-client
```




```
APIKEY="AIzaSyCUt8K6oUDMG_id1WnLxHsJJ--xxbu7ZUM" # Replace with your API key
!pip install --upgrade google-api-python-client
```

```
Collecting google-api-python-client
  Downloading google_api_python_client-1.6.2-py2.py3-none-any.whl (52kB)
    100% |################################| 61kB 2.2MB/s
Requirement already up-to-date: six<2dev,>=1.6.1 in /usr/local/lib/python2.7/dist-packages (from google-api-python-client)
Collecting httplib2<1dev,>=0.9.2 (from google-api-python-client)
  Downloading httplib2-0.10.3.tar.gz (204kB)
    100% |################################| 204kB 3.0MB/s
Collecting oauth2client<5.0.0dev,>=1.5.0 (from google-api-python-client)
  Downloading oauth2client-4.1.1-py2.py3-none-any.whl (185kB)
    100% |################################| 194kB 3.5MB/s
Collecting uritemplate<4dev,>=3.0.0 (from google-api-python-client)
  Downloading uritemplate-3.0.0-py2.py3-none-any.whl
Requirement already up-to-date: rsa>=3.1.4 in /usr/local/lib/python2.7/dist-packages (from oauth2client<5.0.0dev,>=1.5.0->goog
le-api-python-client)
Requirement already up-to-date: pyasn1>=0.1.7 in /usr/local/lib/python2.7/dist-packages (from oauth2client<5.0.0dev,>=1.5.0->g
oogle-api-python-client)
Collecting pyasn1-modules>=0.0.5 (from oauth2client<5.0.0dev,>=1.5.0->google-api-python-client)
  Downloading pyasn1_modules-0.0.9-py2.py3-none-any.whl (60kB)
    100% |################################| 61kB 6.0MB/s
Building wheels for collected packages: httplib2
  Running setup.py bdist_wheel for httplib2 ... - done
  Stored in directory: /root/.cache/pip/wheels/ca/ac/5f/749651f7925b231103f5316cacca82a487810c22d30f011c0c
Successfully built httplib2
```

Invoke Translation API

```
from googleapiclient.discovery import build
service = build('translate', 'v2', developerKey=APIKEY)

# use the service
inputs = ['Today is a good day to learn Machine Learning ',
'It is an amazing technology', 'Great']
outputs = service.translations().list(source='en',
target='th', q=inputs).execute()

# print outputs
for input, output in zip(inputs, outputs['translations']):
  print u"{0} -> {1}".format(input, output['translatedText'])
```

```python
from googleapiclient.discovery import build
service = build('translate', 'v2', developerKey=APIKEY)
# use the service
inputs = ['Today is a good day to learn Machine Learning ', 'It is an amazing technology', 'Great']
outputs = service.translations().list(source='en', target='th', q=inputs).execute()
# print outputs
for input, output in zip(inputs, outputs['translations']):
  print u"{0} -> {1}".format(input, output['translatedText'])

```

```
Today is a good day to learn Machine Learning  -> วันนี้เป็นวันที่ดีในการเรียนรู้การเรียนรู้ด้วยเครื่อง
It is an amazing technology -> เป็นเทคโนโลยีที่น่าอัศจรรย์
Great -> ยิ่งใหญ่
```

Invoke Vision API

```
import base64
IMAGE="gs://cloud-training-demos/vision/sign2.jpg"
vservice = build('vision', 'v1', developerKey=APIKEY)
request = vservice.images().annotate(body={
        'requests': [{
                'image': {
                    'source': {
                        'gcs_image_uri': IMAGE
                    }
                },
                'features': [{
                    'type': 'TEXT_DETECTION',
                    'maxResults': 3,
                }]
            }],
        })
responses = request.execute(num_retries=3)
print responses

foreigntext =
responses['responses'][0]['textAnnotations'][0]['description']
foreignlang =
responses['responses'][0]['textAnnotations'][0]['locale']
print foreignlang, foreigntext
```

```
import base64
IMAGE="gs://cloud-training-demos/vision/sign2.jpg"
vservice = build('vision', 'v1', developerKey=APIKEY)
request = vservice.images().annotate(body={
        'requests': [{
                'image': {
                    'source': {
                        'gcs_image_uri': IMAGE
                    }
                },
                'features': [{
                    'type': 'TEXT_DETECTION',
                    'maxResults': 3,
                }]
            }],
        })
responses = request.execute(num_retries=3)
print responses
```

```
{u'responses': [{u'textAnnotations': [{u'locale': u'zh', u'description': u'\u8bf7\u60a8\u7231\u62a4\u548c\u4fdd\n\u62a4\u536b
\u751f\u521b\u5efa\u4f18\n\u7f8e\u6c34\u73af\u5883\n', u'boundingPoly': {u'vertices': [{u'y': 103, u'x': 150}, {u'y': 103,
 u'x': 1084}, {u'y': 654, u'x': 1084}, {u'y': 654, u'x': 150}]}}, {u'description': u'\u8bf7', u'boundingPoly': {u'vertices':
 [{u'y': 103, u'x': 178}, {u'y': 103, u'x': 322}, {u'y': 241, u'x': 322}, {u'y': 241, u'x': 178}]}}, {u'description': u'\u60a
8', u'boundingPoly': {u'vertices': [{u'y': 107, u'x': 327}, {u'y': 107, u'x': 471}, {u'y': 241, u'x': 471}, {u'y': 241, u'x':
 327}]}}, {u'description': u'\u7231\u62a4', u'boundingPoly': {u'vertices': [{u'y': 107, u'x': 481}, {u'y': 107, u'x': 774},
 {u'y': 245, u'x': 774}, {u'y': 245, u'x': 481}]}}, {u'description': u'\u548c', u'boundingPoly': {u'vertices': [{u'y': 110,
 u'x': 784}, {u'y': 110, u'x': 924}, {u'y': 245, u'x': 924}, {u'y': 245, u'x': 784}]}}, {u'description': u'\u4fdd', u'bounding
Poly': {u'vertices': [{u'y': 103, u'x': 933}, {u'y': 103, u'x': 1078}, {u'y': 237, u'x': 1078}, {u'y': 237, u'x': 933}]}}
```

```
foreigntext = responses['responses'][0]['textAnnotations'][0]['description']
foreignlang = responses['responses'][0]['textAnnotations'][0]['locale']
print foreignlang, foreigntext
```

```
zh 请您爱护和保
护卫生创建优
美水环境
```

Translate Sign

```
inputs=[foreigntext]
outputs = service.translations().list(source=foreignlang,
target='th', q=inputs).execute()
# print outputs
for input, output in zip(inputs, outputs['translations']):
  print u"{0} -> {1}".format(input,
output['translatedText'])
```

```
inputs=[foreigntext]
outputs = service.translations().list(source=foreignlang, target='th', q=inputs).execute()
# print outputs
for input, output in zip(inputs, outputs['translations']):
  print u"{0} -> {1}".format(input, output['translatedText'])
```

```
请您爱护和保
护卫生创建优
美水环境
 -> โปรดสร้างสภาพแวดล้อมการป้องกันน้ำที่สวยงามและการดูแลสุขภาพ
```