



Google Cloud ML Examples

Sung Kim <hunkim+ml@gmail.com>

<https://github.com/hunkim/GoogleCloudMLExamples>

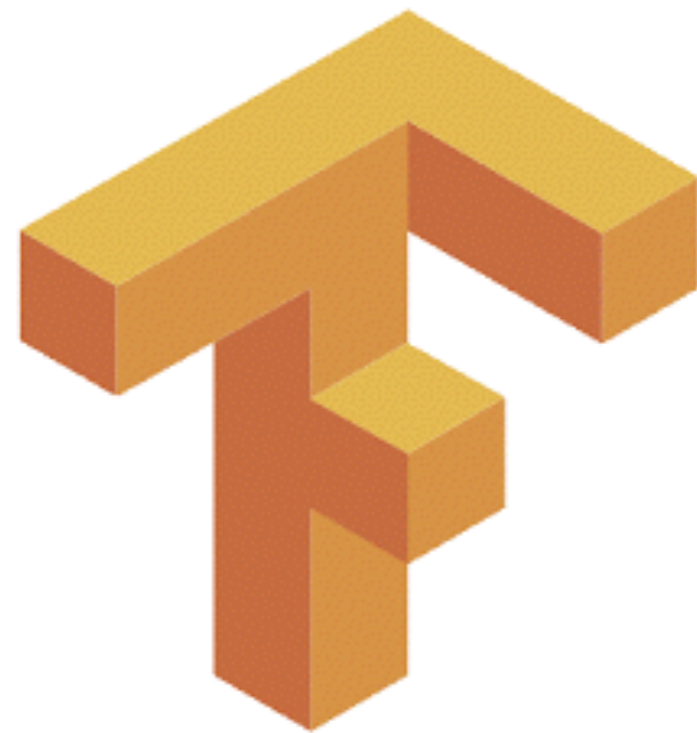


Launch your next project today.

Use Google Cloud Platform and get started with a \$300 free trial.

TRY IT FREE

Local TensorFlow Tasks



TensorFlow Task

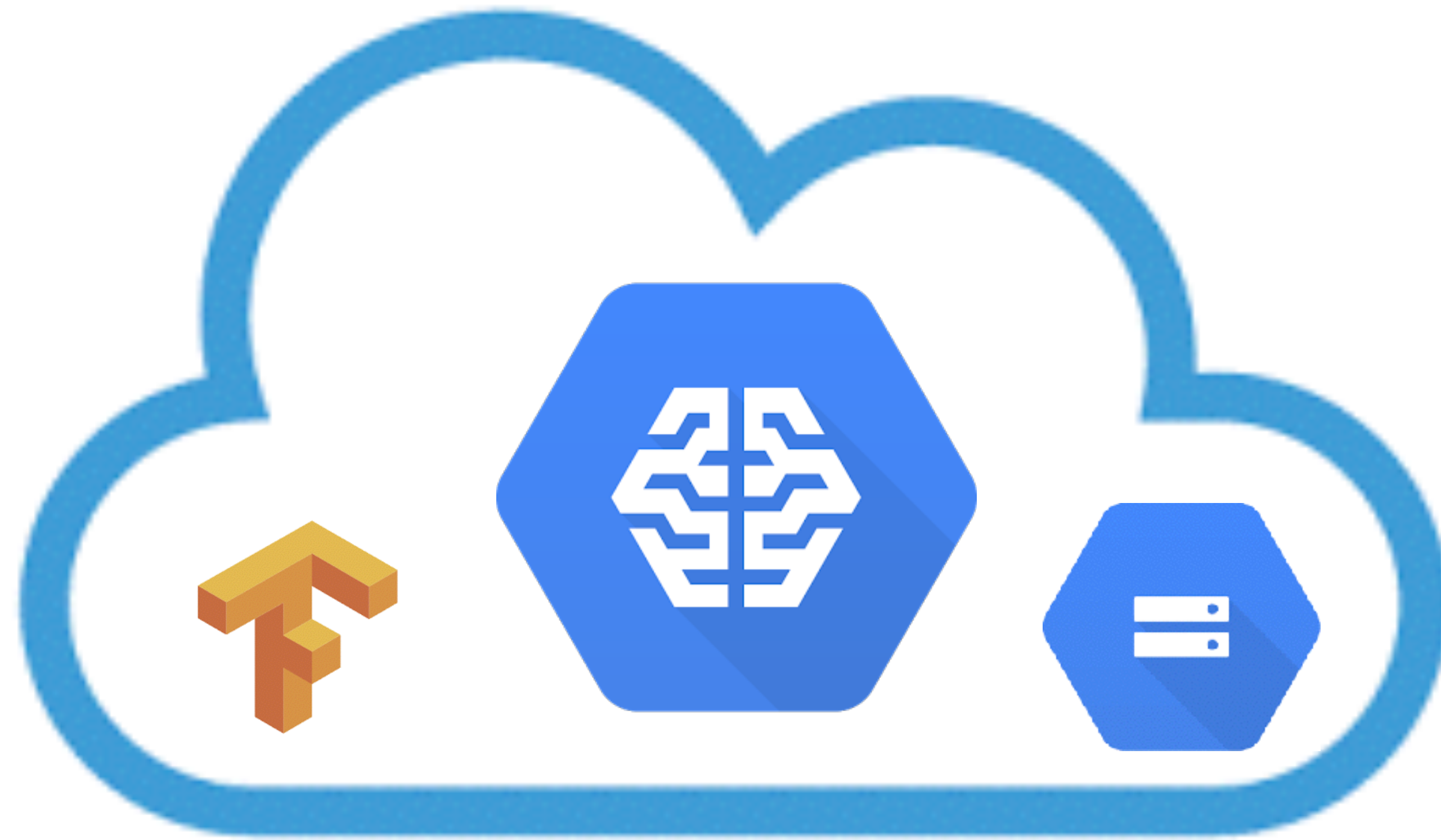


Local Disk

Cloud ML TensorFlow Tasks




TensorFlow Task



Google Cloud ML

Setup *your* environment



Choose this "Cloud Shell" option if you **do not** already have [Docker](#)  installed.

This is the fastest way to try out Cloud ML without installing any software.

All changes will be isolated in the Cloud Shell virtual machine. (You can disconnect and reconnect later without losing your changes.)

Works with: macOS, Linux, and Windows

1. [Start Cloud Shell](#). You should run all of the following commands inside of this command line.
2. Install required tools and dependencies (inside of Cloud Shell) by pasting the following script into the command line:

```
curl https://storage.googleapis.com/cloud-ml/scripts/setup_cloud_shell.sh | bash
```

You should see "Success! Your environment has the required tools and dependencies." when the script finishes successfully.

3. Add the newly-installed tools to your PATH:

```
export PATH=${HOME}/.local/bin:${PATH}
```


Google Cloud Console

The screenshot shows the Google Cloud Platform console interface. At the top, a blue header bar contains the 'Google Cloud Platform' logo, a dropdown menu for the 'tensorflow' project, a search bar, and several utility icons (chat, help, notifications, and a user profile). Below the header, a left-hand navigation pane lists 'Home', 'Dashboard' (which is selected and highlighted in blue), and 'Activity'. The main content area is titled 'Dashboard' and features three primary widgets. The first widget, 'Project: tensorflow', displays the project ID 'tensorflow-145707 (#603879767620)' and includes a button to 'Manage project settings'. The second widget, 'App Engine', shows a 'Summary (count/sec)' dropdown and a message stating 'There is no data for this chart.' The third widget, 'Google Cloud status', indicates that 'All services normal' and provides a link to 'Go to Cloud status dashboard'. In the top right corner of the dashboard area, there is a button to 'Activate Google Cloud Shell' and a 'CUSTOMIZE' link.

Google Cloud Platform tensorflow

Home Dashboard

Activate Google Cloud Shell CUSTOMIZE

Project: tensorflow

ID: tensorflow-145707 (#603879767620)

→ Manage project settings

App Engine

Summary (count/sec)

There is no data for this chart.

Google Cloud status

All services normal

→ Go to Cloud status dashboard

Google Cloud Platform tensorflow

Home

Dashboard

Activity

Project: tensorflow

ID: tensorflow-145707 (# 603879767620)

Manage project settings

App Engine

Summary (count/sec)

There is no data for this chart.

Google Cloud status

All services normal

Go to Cloud status dashboard

tensorflow-145707

Welcome to Cloud Shell! Type "help" to get started.


hunkim@tensorflow-145707:~\$ uname -a

Linux cs-6564-devshell-vm-f34a87e3-efeb-4ed1-9a87-2afc36fc4fed-1d 3.16.0-4-amd64 #1 SMP Debian 3.16.36-1+deb8u1 (2016-09-03) x86_64 GNU/Linux

hunkim@tensorflow-145707:~\$

<https://console.cloud.google.com/>



Choose this "Cloud Shell" option if you **do not** already have [Docker](#)  installed.

This is the fastest way to try out Cloud ML without installing any software.

All changes will be isolated in the Cloud Shell virtual machine. (You can disconnect and reconnect later without losing your changes.)

Works with: macOS, Linux, and Windows

1. [Start Cloud Shell](#). You should run all of the following commands inside of this command line.
2. Install required tools and dependencies (inside of Cloud Shell) by pasting the following script into the command line:

```
curl https://storage.googleapis.com/cloud-ml/scripts/setup_cloud_shell.sh | bash
```

You should see "Success! Your environment has the required tools and dependencies." when the script finishes successfully.

3. Add the newly-installed tools to your PATH:

```
export PATH=${HOME}/.local/bin:${PATH}
```

Verifying your environment

If everything installed without error, you should have a properly-configured development environment. Paste the following script into the command line:

```
curl https://storage.googleapis.com/cloud-ml/scripts/check_environment.py | python
```

You should see "Success! Your environment is configured correctly." when the script finishes successfully.

Initializing your Cloud ML project

Allow the Cloud ML service accounts to access resources in your Google Cloud project:

```
gcloud beta ml init-project
```

Setting up your Cloud Storage bucket

The Cloud ML services need to access Cloud Storage locations to read and write data during model training and batch prediction. This section shows you how to create a new bucket. You may use an existing bucket, but if it is not part of the project you are using to run Cloud ML, you must explicitly [grant access to the Cloud ML service accounts](#).

★ **Note:** The instructions in the [quickstarts](#) assume that you will name your bucket your Cloud Platform project name appended with '-ml'. If you use a different bucket name, you will need to adjust the instructions.

Create a [Google Cloud Storage](#) bucket for reading and writing data during model training and batch prediction:

1. Set a name for your new bucket.

If you want to use the project name with '-ml' appended, get your project name:

```
PROJECT_ID=$(gcloud config list project --format "value(core.project)")  
BUCKET_NAME=${PROJECT_ID}-ml
```

Otherwise, use whatever name you want:

```
BUCKET_NAME="your_bucket_name"
```

2. Create the new bucket:

```
gsutil mb -l us-central1 gs://$BUCKET_NAME
```


Google Cloud commands

- **gcloud**: command-line interface to Google Cloud Platform
 - Google Cloud ML jobs (`gcloud beta ml``)
 - Google Compute Engine virtual machine instances and other resources
 - Google Cloud Dataproc clusters and jobs
 - Google Cloud Deployment manager deployments
 - ...
- **gsutil**: command-line interface to Google Cloud Storage

<https://cloud.google.com/storage/docs/gsutil>

<https://cloud.google.com/sdk/gcloud/>

Example

Example git repository


git clone <https://github.com/hunkim/GoogleCloudMLExamples.git>

▼  **CloudMLExample** (~/.CloudMLExample)


▶  input

▶  output

▼  **train**

 1-multiply.py

 2-input.py

 3-output.py

 __init__.py

 .gitignore

 README.md

Simple Multiplication

```
def run_training():  
    x = tf.placeholder("float") # Create a placeholder 'x'  
    w = tf.Variable(5.0, name="weights")  
    y = tf.mul(w, x)  
  
    with tf.Session() as sess:  
        # Add the variable initializer Op.  
        tf.initialize_all_variables().run()  
        print(sess.run(y, feed_dict={x: 1.0}))  
        print(sess.run(y, feed_dict={x: 2.0}))  
  
def main(_):  
    run_training()  
  
if __name__ == '__main__':  
    tf.app.run()
```

Run locally

```
✓ ~/CloudMLExample  
21:33 $ python -m train.1-multiply  
5.0  
10.0
```

Run on Cloud ML

```
JOB_NAME=<your job name>
```


```
JOB_NAME="task8"
```

```
PROJECT_ID=`gcloud config list project --format "value(core.project)"`
```


```
STAGING_BUCKET=gs://${PROJECT_ID}-ml
```


```
12:38 $ gcloud beta ml jobs submit training ${JOB_NAME} \  
> --package-path=train \  
> --staging-bucket="${STAGING_BUCKET}" \  
> --module-name=train.1-multiply  
createTime: '2016-10-29T04:38:31Z'  
jobId: t8  
state: QUEUED  
trainingInput:  
  packageUri:  
    - gs://tensorflow-145707-ml/cloudml/dist/1477715908/train-0.0.0.tar.gz  
pythonModule: train.1-multiply  
region: us-central1
```


Machine Learning Console


 Google Cloud Platform

TOOLS


 Development


 Deployment Manager


 Test Lab


 Endpoints



BIG DATA

 BigQuery

 Pub/Sub

 Dataproc


 Dataflow







 Machine Learning 


Models [+ CREATE MODEL](#)


	Default version	
^ Lsung	v1	Delete


Jobs

 Google Cloud Platform tensorflow ▾



 Machine Learning

 Models

 Jobs

Jobs

Job ID	Creation time ▾	Status
<u>task7</u>	Oct 27, 2016, 9:38:59 PM	✔ Completed (52 sec)
task5	Oct 27, 2016, 9:02:09 PM	✔ Completed (19 sec)
task2	Oct 27, 2016, 7:49:27 PM	✔ Completed (56 sec)
task1	Oct 27, 2016, 7:15:36 PM	✔ Completed (54 sec)
mul9	Oct 27, 2016, 4:53:25 PM	✔ Completed (55 sec)

Jobs/Task

Google Cloud Platform

tensorflow

Machine Learning

Models

Jobs

← Job details

task7

✔ Completed (52 sec)

Creation time

Oct 27, 2016, 9:38:59 PM

End time

Oct 27, 2016, 9:39:51 PM


Logs

[View logs](#)


Training input







```
{
  "packageUris": [
    "gs://tensorflow-145707-ml/cloudmldist/1477575537/train-0.0.0.tar.gz"
  ],
  "pythonModule": "train.1-multiply",
  "region": "us-central1"
}
```



Jobs/task7/logs


 Google Cloud Platform


tensorflow ▾




 Stackdriver Logging

 Logs

 Logs-based Metrics

 Exports

Logs

Filter by label or text search ▾

Create Metric ?

Cloud Machine Learning, task7 ▾

All logs ▾


Any log level ▾


Jump to date ▾


▶


↺


2016-10-27 View Options ▾


▶  21:39:19.944 OpKernel ('op: "Round" device_type: "CPU" constraint { name: "T" allowed_values { list { type: DT_INT64 ..


▶  21:39:19.945 OpKernel ('op: "Round" device_type: "CPU" constraint { name: "T" allowed_values { list { type: DT_INT32 ..


▶  21:39:19.945 OpKernel ('op: "Round" device_type: "CPU" constraint { name: "T" allowed_values { list { type: DT_DOUBLE..


▶  21:39:19.946 OpKernel ('op: "Round" device_type: "CPU" constraint { name: "T" allowed_values { list { type: DT_FLOAT ..


▶  21:39:19.946 OpKernel ('op: "Round" device_type: "CPU" constraint { name: "T" allowed_values { list { type: DT_HALF }..


▶  21:39:19.955 5.0


▶  21:39:19.956 10.0


▶  21:39:19.995 Module completed; cleaning up.

▶  21:39:19.995 Clean up finished.

▶  21:39:19.996 Task completed successfully.

▶  21:39:51.694 Tearing down TensorFlow.

▶  21:39:59.116 Finished tearing down TensorFlow.

▶  21:40:05.036 Job completed successfully.

Input Example

```

# Basic model parameters as external flags.
flags = tf.app.flags
FLAGS = flags.FLAGS
flags.DEFINE_string('input_dir', 'input', 'Input Directory.')

def run_training():
    csv_file = os.path.join(FLAGS.input_dir, 'input.csv');
    filename_queue = tf.train.string_input_producer([csv_file])
    key, value = tf.TextLineReader().read(filename_queue)

    col1, col2 = tf.decode_csv(value, record_defaults=[[1], [1]])

    x = tf.placeholder("float") # Create a placeholder 'x'
    w = tf.Variable(5.0, name="weights")
    y = tf.mul(w, x)

    with tf.Session() as sess:
        # Add the variable initializer Op.
        tf.initialize_all_variables().run()

        # Start populating the filename queue.
        coord = tf.train.Coordinator()
        threads = tf.train.start_queue_runners(coord=coord)

        for i in range(2):
            # Retrieve a single instance:
            x1, x2 = sess.run([col1, col2])
            print(sess.run(y, feed_dict={x: x1}))
            print(sess.run(y, feed_dict={x: x2}))

        coord.request_stop()
        coord.join(threads)

```

CSV File Reading

```

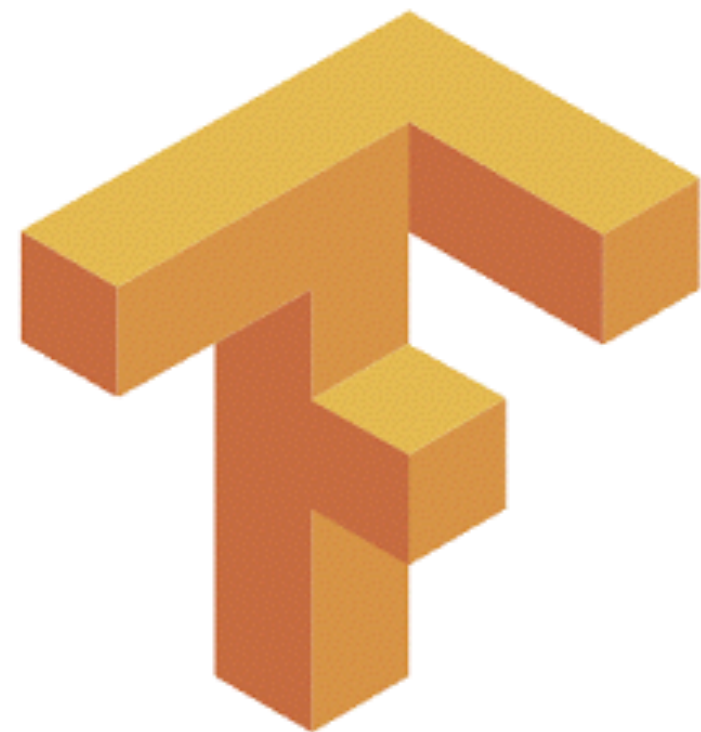
✓ ~/CloudMLExample
23:05 $ cat input/input.csv
1,2
3,4

```

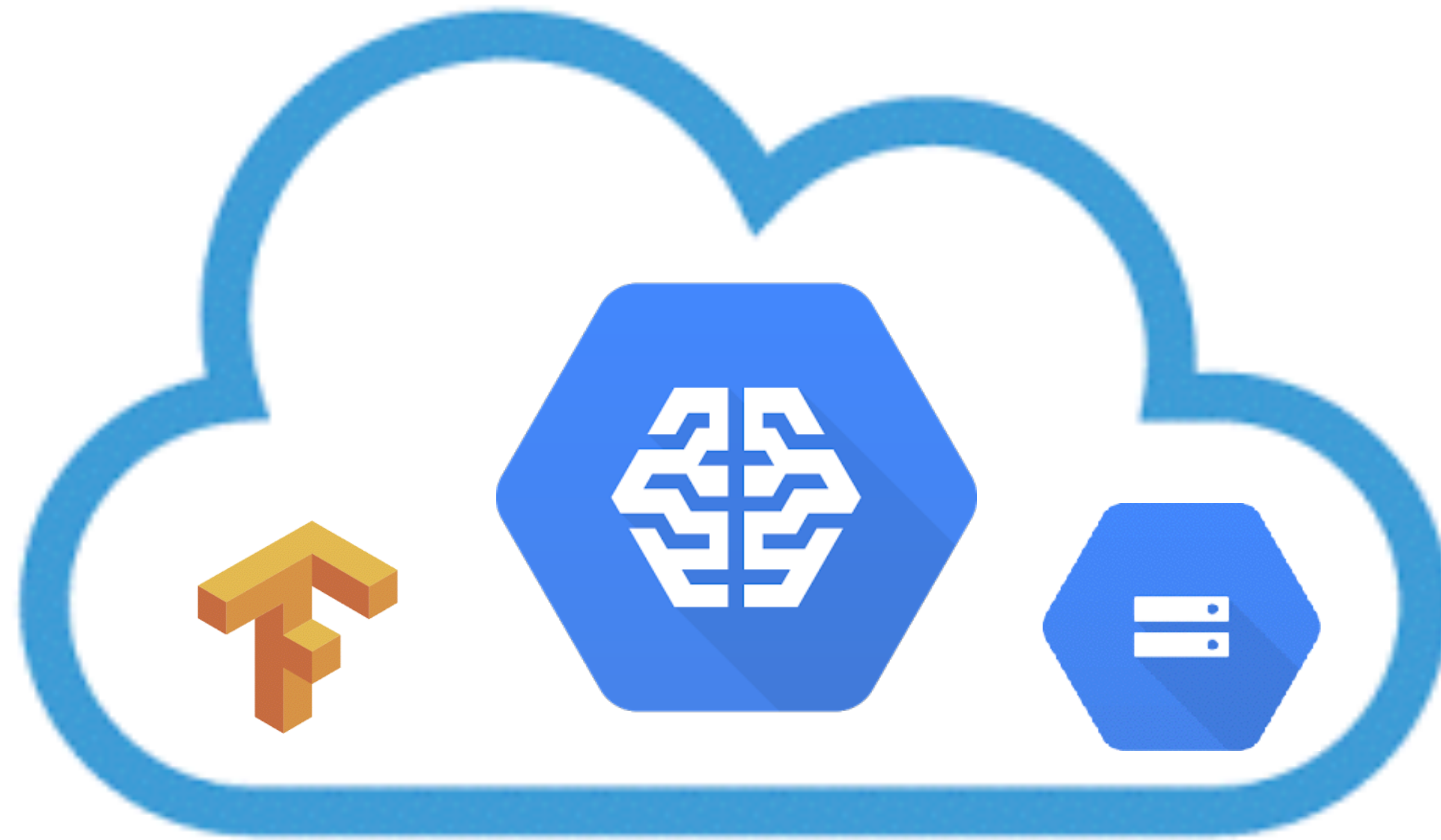
Run locally

```
23:08 $ cat input/input.csv
1,2
3,4
✓ ~/CloudMLExample
08:49 $ python -m train.2-input
5.0
10.0
15.0
20.0
```


Cloud ML TensorFlow Tasks



TensorFlow Task



Google Cloud ML

Setting and file copy

```
JOB_NAME="task9"
```

```
PROJECT_ID=`gcloud config list project --format "value(core.project)"`
```

```
STAGING_BUCKET=gs://${PROJECT_ID}-ml
```

```
INPUT_PATH=${STAGING_BUCKET}/input
```

```
gsutil cp input/input.csv $INPUT_PATH/input.csv
```

```
09:10 $ gsutil cp input/input.csv $INPUT_PATH/input.csv
Copying file:///input/input.csv [Content-Type=text/csv]...
/ [1 files][ 8.0 B/ 8.0 B]
Operation completed over 1 objects/8.0 B.
```


Google Storage

Google Cloud Platform

Networking

STORAGE

Bigtable

SQL

Datastore

Storage

STACKDRIVER

ser

CREATE BUCKET

REFRESH

DELETE

standard Storage buckets are now listed as Regional or Multi-Regional, depending on their location settings. [Learn more](#)

Dismiss

Filter by prefix...

Name	Default storage class ?	Location	
tensorflow-145707-ml	Multi-Regional	US	
tensorflow-145707.appspot.com	Multi-Regional	US	

Storage

Browser

Transfer

Settings

Browser

UPLOAD FILES

UPLOAD FOLDER

CREATE FOLDER


REFRESH

SHARE PUBLICLY

DELETE

Buckets / tensorflow-145707-ml / input

Filter by prefix...

<input type="checkbox"/>	Name	Size	Type	Storage class	Last modified	Share publicly
<input type="checkbox"/>	 input.csv	8 B	text/csv	Multi-Regional	10/28/16, 9:11 AM	<input type="checkbox"/>

Run on Cloud ML

```
12:51 $ gcloud beta ml jobs submit training ${JOB_NAME} \  
> --package-path=train \  
> --staging-bucket="${STAGING_BUCKET}" \  
> --module-name=train.2-input \  
> -- --input_dir="${INPUT_PATH}"  
createTime: '2016-10-29T04:51:17Z'  
jobId: task_10  
state: QUEUED  
trainingInput:  
  args:  
    - --input_dir=gs://tensorflow-145707-ml/input  
  packageUri:  
    - gs://tensorflow-145707-ml/cloudml/dist/1477716675/train-0.0.0.tar.gz  
pythonModule: train.2-input  
region: us-central1
```

Jobs

Google Cloud Platform tensorflow

Machine Learning

Models

Jobs

Job details

task10

Completed (48 sec)

Creation time

Oct 28, 2016, 9:13:05 AM

End time

Oct 28, 2016, 9:13:53 AM

Logs

[View logs](#)

Training input

```
{
  "packageUri": [
    "gs://tensorflow-145707-ml/cloudmldist/1477617183/train-0.0.0.tar.gz"
  ],
  "pythonModule": "train.2-input",
  "args": [
    "--input_dir=gs://tensorflow-145707-ml/input"
  ],
  "region": "us-central1"
}
```

Logs



Logs



Logs-based Metrics



Exports

Filter by label or text search

Create Metric



Cloud Machine Learning, task10

All logs

Any log level

Jump to date



2016-10-28

View Options

```
▶ [i] 09:13:23.971 Stored in directory: /root/.cache/pip/wheels/45/48/75/d931bb937bb1cb4558cc210e4cdc3be905e66eb958d7942922
▶ [i] 09:13:23.974 Successfully built train
▶ [i] 09:13:23.974 Installing collected packages: train
▶ [i] 09:13:24.036 Successfully installed train
▶ [i] 09:13:24.351 Running command: python -m train.2-input --input_dir=gs://tensorflow-145707-ml/input
▶ [!!] 09:13:25.124 OpKernel ('op: "Round" device_type: "CPU" constraint { name: "T" allowed_values { list { type: DT_INT64 } } }) fo..
▶ [!!] 09:13:25.125 OpKernel ('op: "Round" device_type: "CPU" constraint { name: "T" allowed_values { list { type: DT_INT32 } } }) fo..
▶ [!!] 09:13:25.125 OpKernel ('op: "Round" device_type: "CPU" constraint { name: "T" allowed_values { list { type: DT_DOUBLE } } }) f..
▶ [!!] 09:13:25.125 OpKernel ('op: "Round" device_type: "CPU" constraint { name: "T" allowed_values { list { type: DT_FLOAT } } }) fo..
▶ [!!] 09:13:25.125 OpKernel ('op: "Round" device_type: "CPU" constraint { name: "T" allowed_values { list { type: DT_HALF } } }) for..
▶ [i] 09:13:25.251 5.0
▶ [i] 09:13:25.252 10.0
▶ [i] 09:13:25.252 15.0
▶ [i] 09:13:25.252 20.0
▶ [i] 09:13:25.279 Module completed; cleaning up.
```

Output Example


```

# Basic model parameters as external flags.
flags = tf.app.flags
FLAGS = flags.FLAGS
flags.DEFINE_string('output_dir', 'output', 'Output Directory.')

def run_training():
    x = tf.placeholder("float") # Create a placeholder 'x'
    w = tf.Variable(5.0, name="weights")
    y = tf.mul(w, x)

    with tf.Session() as sess:
        # Add the variable initializer Op.
        tf.initialize_all_variables().run()

        print(sess.run(y, feed_dict={x: 1.0}))
        print(sess.run(y, feed_dict={x: 2.0}))

        # Create a saver for writing training checkpoints.
        saver = tf.train.Saver()
        checkpoint_file = os.path.join(FLAGS.output_dir, 'checkpoint')
        saver.save(sess, checkpoint_file, global_step=0)

def main(_):
    run_training()

if __name__ == '__main__':
    tf.app.run()

```

TensorFlow Saver

Local Run

```
16:47 $ python -m train.3-output
```

```
5.0
```

```
10.0
```

```
✓ ~/CloudMLExample
```

```
16:47 $ ls -la output/
```

```
total 24
```

```
drwxr-xr-x  5 hunkim  staff   170 Oct 28 16:47 .
```

```
drwxr-xr-x  7 hunkim  staff   238 Oct 28 16:47 ..
```

```
-rw-r--r--  1 hunkim  staff    81 Oct 28 16:47 checkpoint
```

```
-rw-r--r--  1 hunkim  staff   154 Oct 28 16:47 checkpoint-0
```

```
-rw-r--r--  1 hunkim  staff  2684 Oct 28 16:47 checkpoint-0.meta
```

Configuration


Set variables


```
JOB_NAME="task20"  
PROJECT_ID=`gcloud config list project --format "value(core.project)"`  
STAGING_BUCKET=gs://${PROJECT_ID}-ml  
OUTPUT_PATH=${STAGING_BUCKET}/output/
```


Create the output folder (Copy an empty file to the GS path with trailing slash, `/`)


```
gsutil cp /dev/null $OUTPUT_PATH
```

Create/Check the output folder

Storage

Browser

Transfer

Settings

Browser

↑ UPLOAD FILES

↑ UPLOAD FOLDER

+ CREATE FOLDER

↻ REFRESH

👤 SHARE PUBLICLY

🗑 DELETE

Buckets / tensorflow-145707-ml





Filter by prefix...

<input type="checkbox"/>	Name	Size	Type	Storage class	Last modified	Share publicly
<input type="checkbox"/>	cloudmldist/	—	Folder	—	—	
<input type="checkbox"/>	input/	—	Folder	—	—	
<input type="checkbox"/>	<u>output/</u>	—	Folder	—	—	
<input type="checkbox"/>	shell_test/	—	Folder	—	—	
<input type="checkbox"/>	test_dist/	—	Folder	—	—	


Run on Cloud ML


```
:53 $ gcloud beta ml jobs submit training ${JOB_NAME} \
--package-path=train \
--staging-bucket="${STAGING_BUCKET}" \
--module-name=train.3-output \
-- --output_dir="${OUTPUT_PATH}"
createTime: '2016-10-29T04:53:37Z'
jobId: t20
state: QUEUED
trainingInput:
args:
- --output_dir=gs://tensorflow-145707-ml/output/
packageUri:
- gs://tensorflow-145707-ml/cloudml/dist/1477716814/train-0.0.0.tar.gz
pythonModule: train.3-output
region: us-central1
```


Job completed


 Machine Learning	← Job details								
 Models	task20								
 Jobs	<div> Completed (51 sec)</div> <table><tr><td>Creation time</td><td>Oct 28, 2016, 5:14:27 PM</td></tr><tr><td>End time</td><td>Oct 28, 2016, 5:15:18 PM</td></tr><tr><td>Logs</td><td>View logs</td></tr><tr><td>Training input</td><td><pre>{ "packageUri": ["gs://tensorflow-145707-ml/cloudmldist/1477646065/train-0.0.0.tar.gz"], "pythonModule": "train.3-output", "args": ["--output_dir=gs://tensorflow-145707-ml/output"], "region": "us-central1" }</pre></td></tr></table>	Creation time	Oct 28, 2016, 5:14:27 PM	End time	Oct 28, 2016, 5:15:18 PM	Logs	View logs	Training input	<pre>{ "packageUri": ["gs://tensorflow-145707-ml/cloudmldist/1477646065/train-0.0.0.tar.gz"], "pythonModule": "train.3-output", "args": ["--output_dir=gs://tensorflow-145707-ml/output"], "region": "us-central1" }</pre>
Creation time	Oct 28, 2016, 5:14:27 PM								
End time	Oct 28, 2016, 5:15:18 PM								
Logs	View logs								
Training input	<pre>{ "packageUri": ["gs://tensorflow-145707-ml/cloudmldist/1477646065/train-0.0.0.tar.gz"], "pythonModule": "train.3-output", "args": ["--output_dir=gs://tensorflow-145707-ml/output"], "region": "us-central1" }</pre>								

Generated checkpoint files

Storage

Browser

Transfer

Settings

Browser

⬆️ UPLOAD FILES

⬆️ UPLOAD FOLDER

📁 CREATE FOLDER




🔄 REFRESH

👥 SHARE PUBLICLY

🗑️ DELETE

Buckets / tensorflow-145707-ml / output

🔍 Filter by prefix...

<input type="checkbox"/>	Name	Size	Type	Storage class	Last modified	Share publicly	
<input type="checkbox"/>	 checkpoint	81 B	—	Multi-Regional	10/28/16, 5:14 PM	<input type="checkbox"/>	⋮
<input type="checkbox"/>	 checkpoint-0	154 B	—	Multi-Regional	10/28/16, 5:14 PM	<input type="checkbox"/>	⋮
<input type="checkbox"/>	 checkpoint-0.meta	2.56 KB	—	Multi-Regional	10/28/16, 5:14 PM	<input type="checkbox"/>	⋮

With Great Power Comes Great Responsibility

ML PRICING

Cloud Machine Learning charges for training ML models and running predictions with trained models. For detailed pricing information, please view the [pricing guide](#).

ITEM	US	EUROPE/ASIA
Training Clusters		
Basic Tier	\$0.49/hour	\$0.54/hour
Standard Tier	\$4.90/hour	\$5.40/hour
Premium Tier	\$36.75/hour	\$40.50/hour
Custom Cluster Configuration	\$0.49/hour per ML training unit	\$0.54/hour per ML training unit
Prediction Requests		
Up to 100M per Month	\$0.10 / 1K +\$0.40/Node Hour	\$0.11 / 1K +\$0.44/Node Hour
Requests over 100M per month	\$0.05 / 1K +\$0.40/Node Hour	\$0.05 / 1K +\$0.44/Node Hour

Check your bills!

Google Cloud Platform tensorflow

Billing

Overview

Budgets & alerts

History

Billing export

Profile

Payment settings

HistoryMy Billing Account

Current balance

\$0.00

Estimated tax of \$0.00 has been deducted

Make a payment

How you pay

Manage

Visa •••3638

No backup

Monthly automatic payments

Profile

Manage

Sung Kim

HKUST

Kowloon City

Hong Kong

All transactionsDetailedExportLast 3 months

Oct 1, 2016 - Oct 28, 2016

Ending balance: \$0.00

Date	Description	Debits (\$)	Credits (\$)	Balance (\$)
Oct 1, 2016 - Oct 27, 2016	Cloud Machine Learning Training Unit-Hours: 2.025 Hours (Source:tensorflow [tensorflow-145707])	0.99		

Next

- Could ML deploy
- Hyper-parameter tuning
- Distributed training tasks