Project Milestone 4 - Data Processing: Dataflow- apache beam

Esam Uddin - 100711116, Mihir Patel - 100702168, Jane Coralde 100660214, Haider Sarmad

3/28/2022

Given Lab 4 Repository: https://github.com/goergedaoud/SOFE4630U-tut3

Group Project Repository:

https://github.com/esam191/Intelligent-Transportation-System

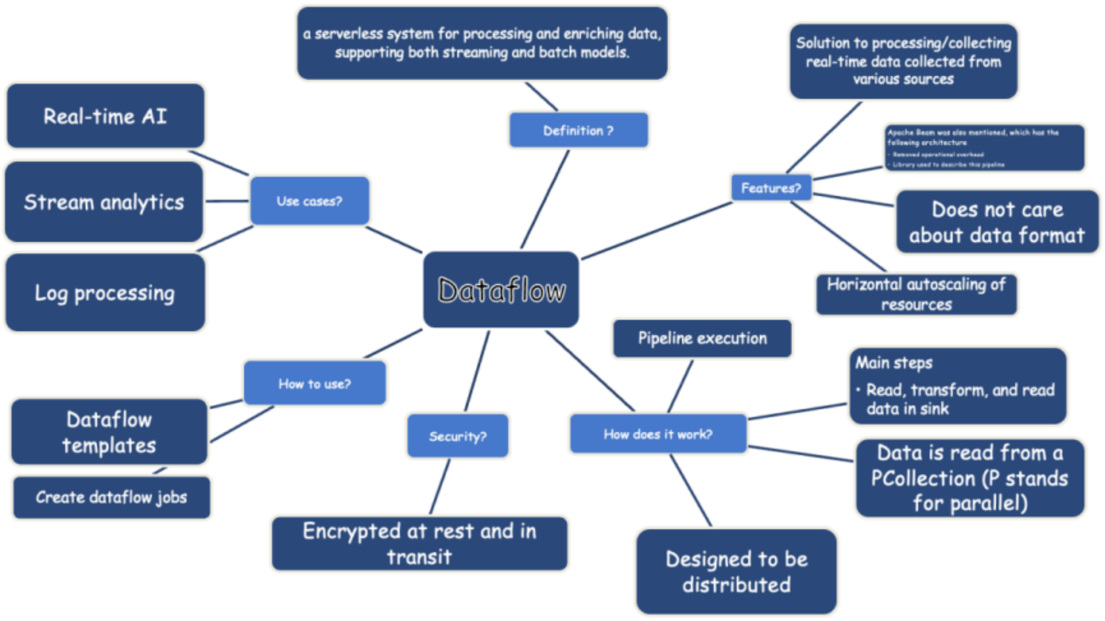
Objectives:

- Get familiar with Dataflow
- Understand MapReduce.
- Run batch and Stream Processing examples over GCP.

Procedure:

1. **Watch the following video about Google Cloud Dataflow**

   A: Please see the mindmap created below.

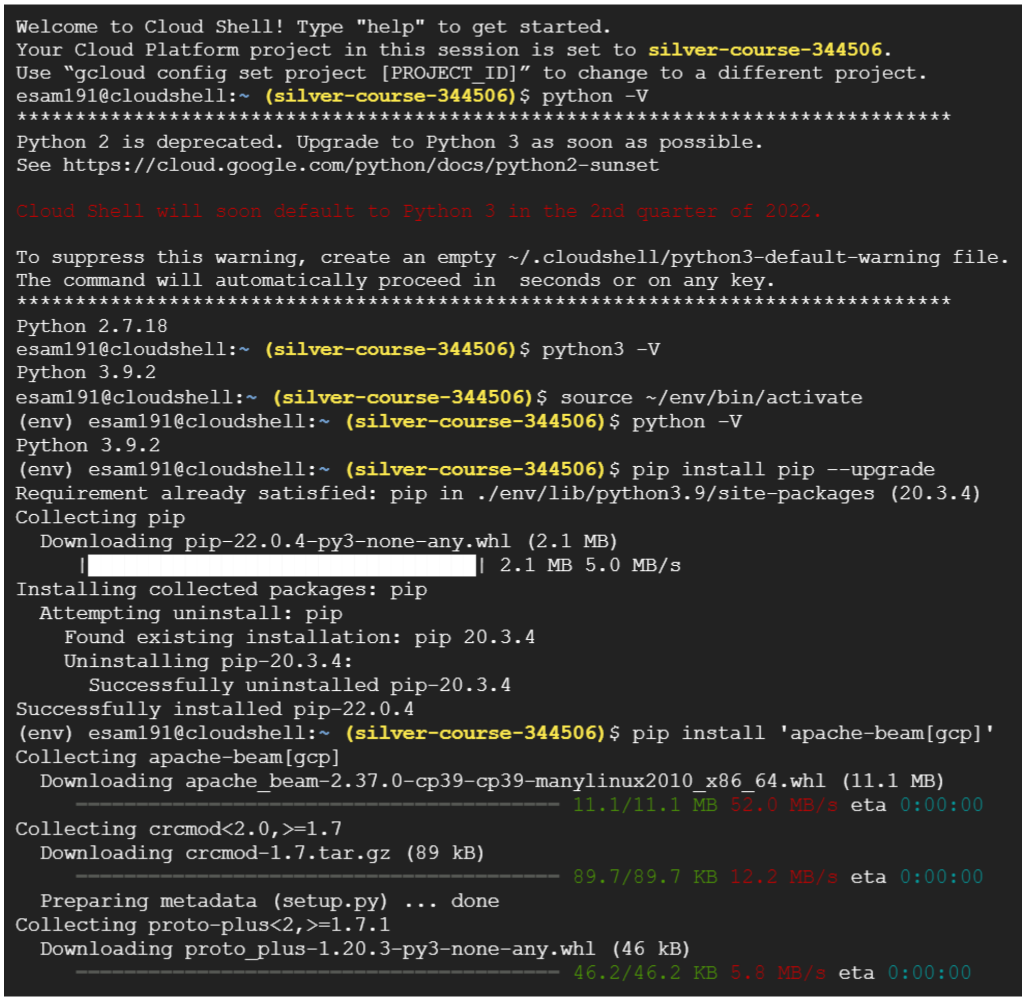

a serverless system for processing and enriching data, supporting both streaming and batch models.
Definition ?
Real-time AI
Stream analytics
Log processing
Use cases?
Solution to processing/collecting real-time data collected from various sources
Apache Beam was also mentioned, which has the following architecture
- Removed operational overhead
- Library used to describe this pipeline
Features?
Does not care about data format
Horizontal autoscaling of resources
Dataflow
Pipeline execution
Main steps
• Read, transform, and read data in sink
How to use?
Dataflow templates
Create dataflow jobs
Security?
Encrypted at rest and in transit
How does it work?
Data is read from a PCollection (P stands for parallel)
Designed to be distributed

2. **Watch the following video Describing how to apply MapReduce to count the words within a certain document.**

   A: The second output added in wordcount2.py is the part which applies MapReduce

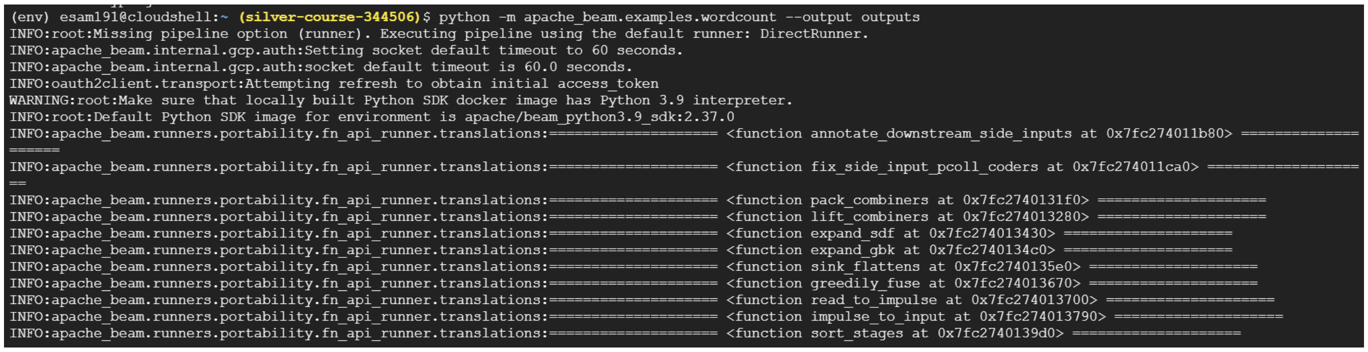
   https://www.youtube.com/watch?v=re6c_ee7uTc

   Part 1: creating python environment

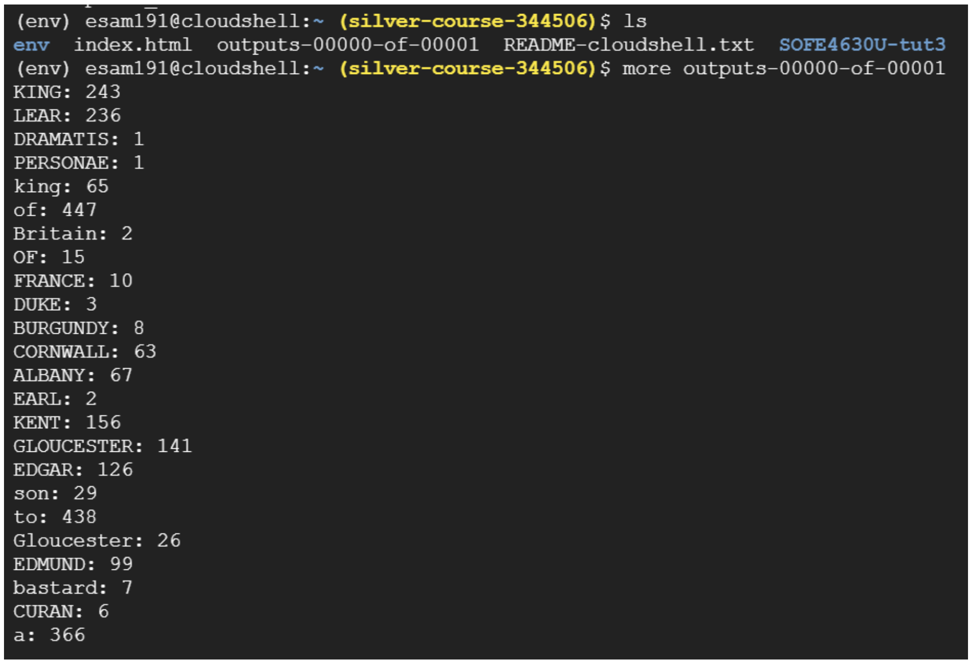


```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to silver-course-344506.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
esam191@cloudshell:~ (silver-course-344506)$ python -V
********************************************************************************
Python 2 is deprecated. Upgrade to Python 3 as soon as possible.
See https://cloud.google.com/python/docs/python2-sunset

Cloud Shell will soon default to Python 3 in the 2nd quarter of 2022.

To suppress this warning, create an empty ~/.cloudshell/python3-default-warning file.
The command will automatically proceed in  seconds or on any key.
********************************************************************************
Python 2.7.18
esam191@cloudshell:~ (silver-course-344506)$ python3 -V
Python 3.9.2
esam191@cloudshell:~ (silver-course-344506)$ source ~/env/bin/activate
(env) esam191@cloudshell:~ (silver-course-344506)$ python -V
Python 3.9.2
(env) esam191@cloudshell:~ (silver-course-344506)$ pip install pip --upgrade
Requirement already satisfied: pip in ./env/lib/python3.9/site-packages (20.3.4)
Collecting pip
  Downloading pip-22.0.4-py3-none-any.whl (2.1 MB)
     |████████████████████████████████| 2.1 MB 5.0 MB/s
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 20.3.4
    Uninstalling pip-20.3.4:
      Successfully uninstalled pip-20.3.4
Successfully installed pip-22.0.4
(env) esam191@cloudshell:~ (silver-course-344506)$ pip install 'apache-beam[gcp]'
Collecting apache-beam[gcp]
  Downloading apache_beam-2.37.0-cp39-cp39-manylinux2010_x86_64.whl (11.1 MB)
     ---------------------------------------- 11.1/11.1 MB 52.0 MB/s eta 0:00:00
Collecting crcmod<2.0,>=1.7
  Downloading crcmod-1.7.tar.gz (89 kB)
     ---------------------------------------- 89.7/89.7 KB 12.2 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting proto-plus<2,>=1.7.1
  Downloading proto_plus-1.20.3-py3-none-any.whl (46 kB)
     ---------------------------------------- 46.2/46.2 KB 5.8 MB/s eta 0:00:00
```

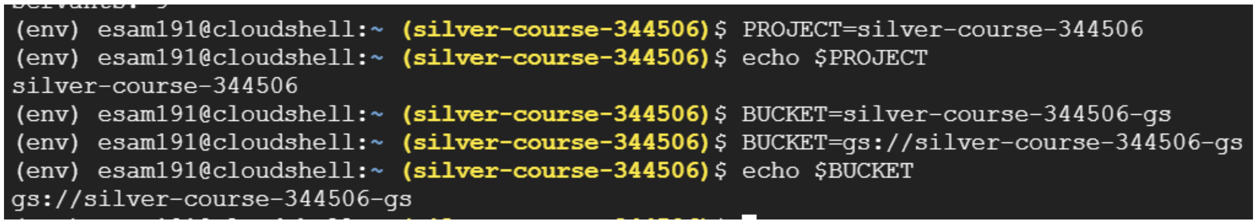
Part 2:testing wordcount example + creating cloud storage



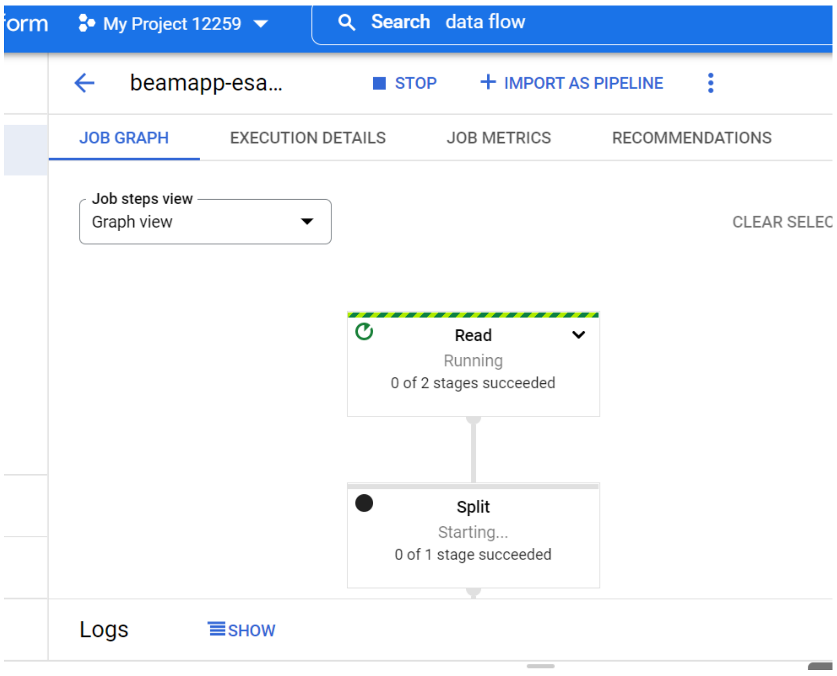
```
(env) esam191@cloudshell:~ (silver-course-344506)$ python -m apache_beam.examples.wordcount --output outputs
INFO:root:Missing pipeline option (runner). Executing pipeline using the default runner: DirectRunner.
INFO:apache_beam.internal.gcp.auth:Setting socket default timeout to 60 seconds.
INFO:apache_beam.internal.gcp.auth:socket default timeout is 60.0 seconds.
INFO:oauth2client.transport:Attempting refresh to obtain initial access_token
WARNING:root:Make sure that locally built Python SDK docker image has Python 3.9 interpreter.
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.37.0
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function annotate_downstream_side_inputs at 0x7fc274011b80> ===============
======
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function fix_side_input_pcoll_coders at 0x7fc274011ca0> ==================
==
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function pack_combiners at 0x7fc2740131f0> ====================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function lift_combiners at 0x7fc274013280> ====================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function expand_sdf at 0x7fc274013430> ====================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function expand_gbk at 0x7fc2740134c0> ====================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function sink_flattens at 0x7fc2740135e0> ====================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function greedily_fuse at 0x7fc274013670> ====================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function read_to_impulse at 0x7fc274013700> ====================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function impulse_to_input at 0x7fc274013790> ====================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function sort_stages at 0x7fc2740139d0> ====================
```

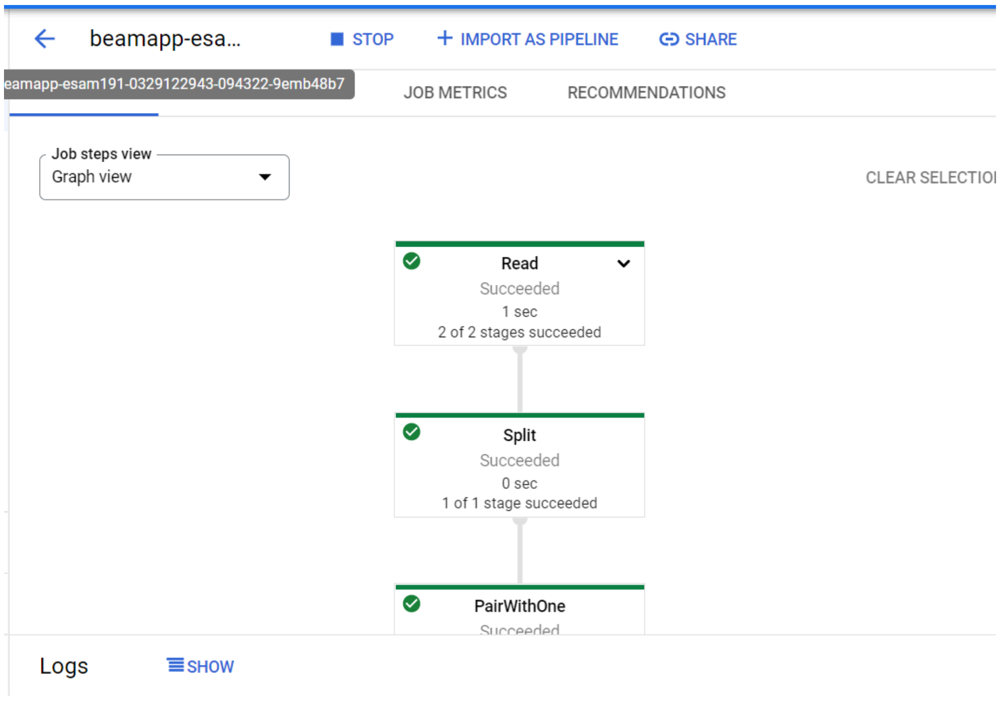


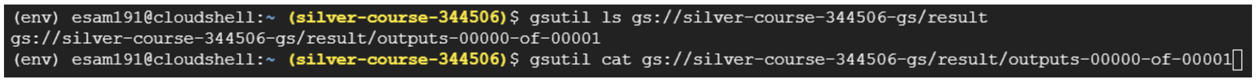
```
(env) esam191@cloudshell:~ (silver-course-344506)$ ls
env  index.html  outputs-00000-of-00001  README-cloudshell.txt  SOFE4630U-tut3
(env) esam191@cloudshell:~ (silver-course-344506)$ more outputs-00000-of-00001
KING: 243
LEAR: 236
DRAMATIS: 1
PERSONAE: 1
king: 65
of: 447
Britain: 2
OF: 15
FRANCE: 10
DUKE: 3
BURGUNDY: 8
CORNWALL: 63
ALBANY: 67
EARL: 2
KENT: 156
GLOUCESTER: 141
EDGAR: 126
son: 29
to: 438
Gloucester: 26
EDMUND: 99
bastard: 7
CURAN: 6
a: 366
```



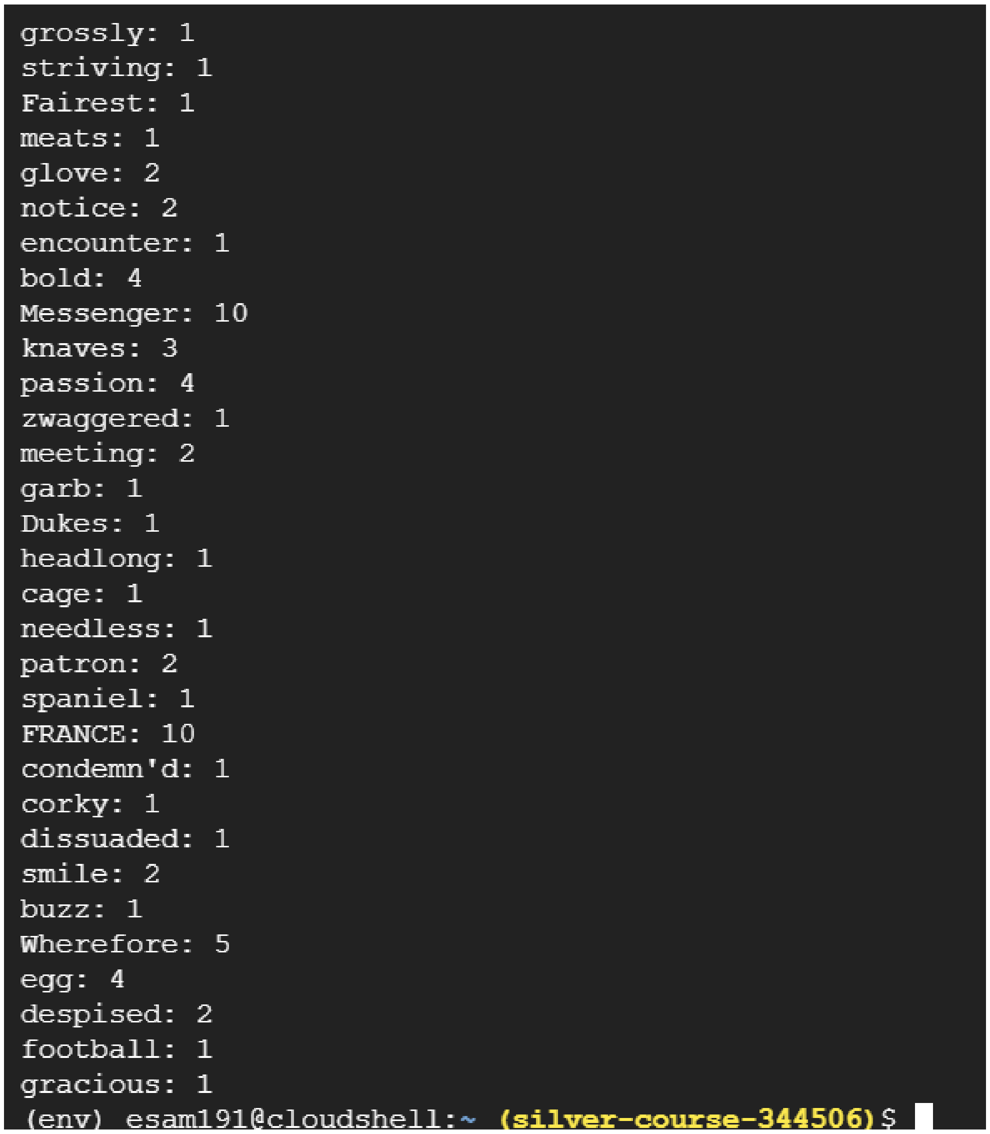
```
(env) esam191@cloudshell:~ (silver-course-344506)$ PROJECT=silver-course-344506
(env) esam191@cloudshell:~ (silver-course-344506)$ echo $PROJECT
silver-course-344506
(env) esam191@cloudshell:~ (silver-course-344506)$ BUCKET=silver-course-344506-gs
(env) esam191@cloudshell:~ (silver-course-344506)$ BUCKET=gs://silver-course-344506-gs
(env) esam191@cloudshell:~ (silver-course-344506)$ echo $BUCKET
gs://silver-course-344506-gs
```



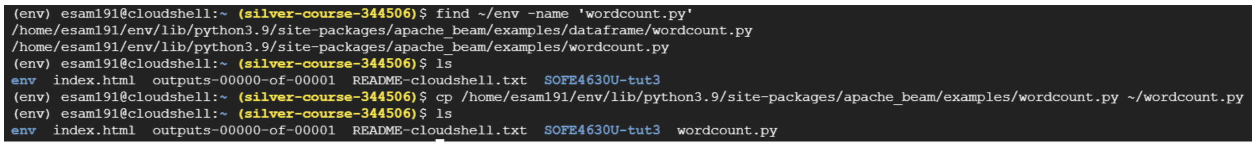




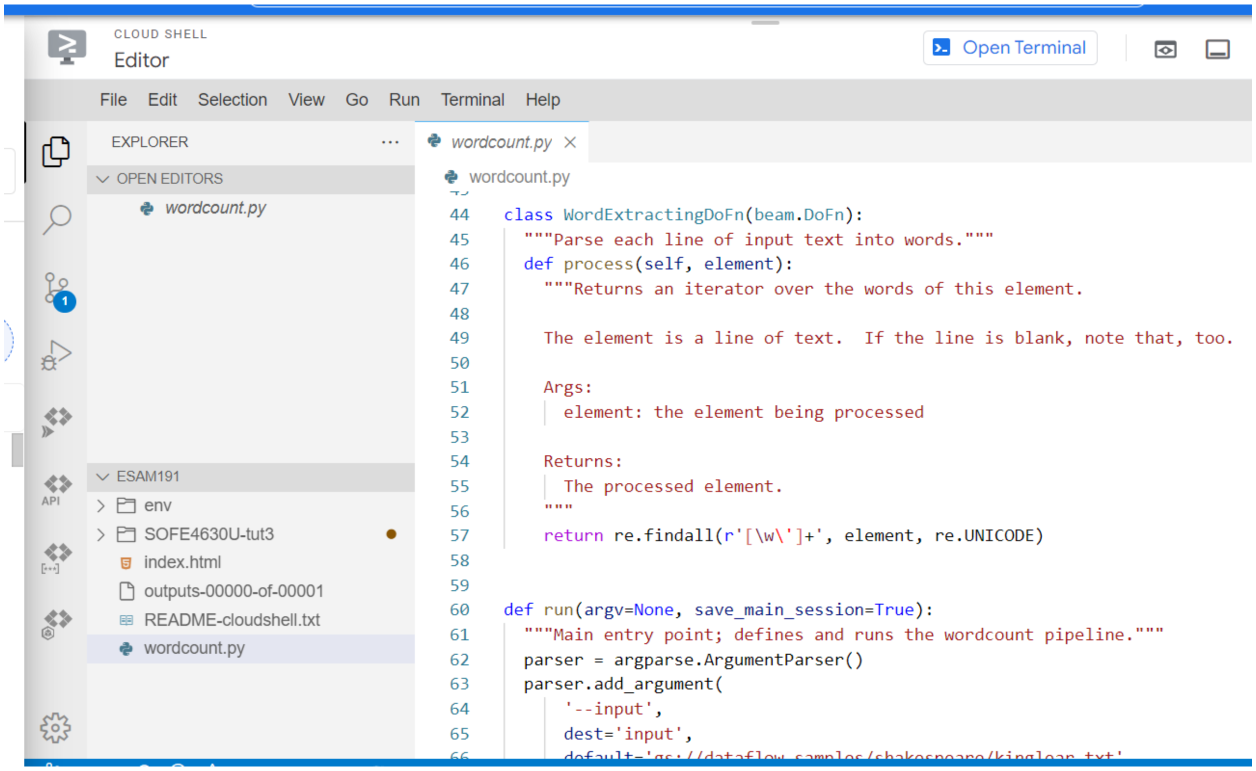
```
(env) esam191@cloudshell:~ (silver-course-344506)$ gsutil ls gs://silver-course-344506-gs/result
gs://silver-course-344506-gs/result/outputs-00000-of-00001
(env) esam191@cloudshell:~ (silver-course-344506)$ gsutil cat gs://silver-course-344506-gs/result/outputs-00000-of-00001
```



```
grossly: 1
striving: 1
Fairest: 1
meats: 1
glove: 2
notice: 2
encounter: 1
bold: 4
Messenger: 10
knaves: 3
passion: 4
zwaggered: 1
meeting: 2
garb: 1
Dukes: 1
headlong: 1
cage: 1
needless: 1
patron: 2
spaniel: 1
FRANCE: 10
condemn'd: 1
corky: 1
dissuaded: 1
smile: 2
buzz: 1
Wherefore: 5
egg: 4
despised: 2
football: 1
gracious: 1
(env) esam191@cloudshell:~ (silver-course-344506)$
```



```
(env) esam191@cloudshell:~ (silver-course-344506)$ find ~/env -name 'wordcount.py'
/home/esam191/env/lib/python3.9/site-packages/apache_beam/examples/dataframe/wordcount.py
/home/esam191/env/lib/python3.9/site-packages/apache_beam/examples/wordcount.py
(env) esam191@cloudshell:~ (silver-course-344506)$ ls
env  index.html  outputs-00000-of-00001  README-cloudshell.txt  SOFE4630U-tut3
(env) esam191@cloudshell:~ (silver-course-344506)$ cp /home/esam191/env/lib/python3.9/site-packages/apache_beam/examples/wordcount.py ~/wordcount.py
(env) esam191@cloudshell:~ (silver-course-344506)$ ls
env  index.html  outputs-00000-of-00001  README-cloudshell.txt  SOFE4630U-tut3  wordcount.py
```



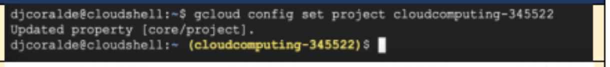
```
class WordExtractingDoFn(beam.DoFn):
  """Parse each line of input text into words."""
  def process(self, element):
    """Returns an iterator over the words of this element.

    The element is a line of text.  If the line is blank, note that, too.

    Args:
      element: the element being processed

    Returns:
      The processed element.
    """
    return re.findall(r'[\w\']+', element, re.UNICODE)


def run(argv=None, save_main_session=True):
  """Main entry point; defines and runs the wordcount pipeline."""
  parser = argparse.ArgumentParser()
  parser.add_argument(
      '--input',
      dest='input',
```

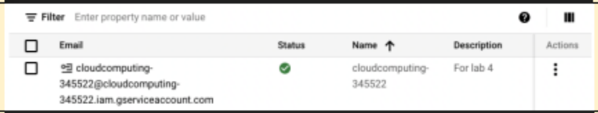
3. **Follow the following video to set up the GCP environment for Dataflow and run wordcount examples.**

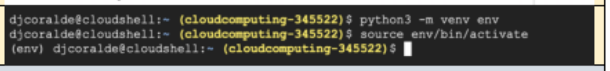
   https://www.youtube.com/watch?v=re6c_ee7uTc

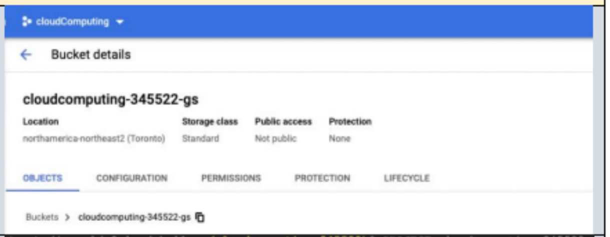
3.1 Activity Following the given video, the table below shows a summary of actions/commands/screenshots to reflect all tasks in the video.

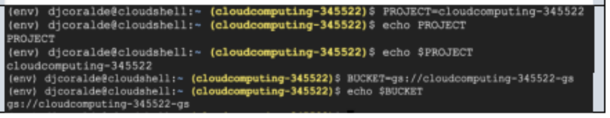
	Action	Command	Screenshot

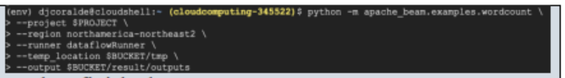
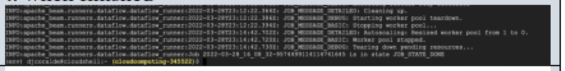
Setup	Gloud config	`Gcloud config set project <name_of_project>`	
			`djcoralde@cloudshell:~$ gcloud config set project cloudcomputing-345522`
			`Updated property [core/project].`
			`djcoralde@cloudshell:~ (cloudcomputing-345522)$`

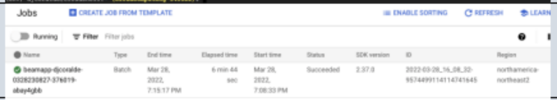
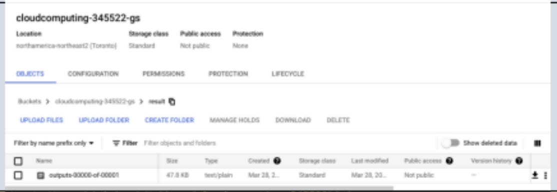
	Create services account		 Filter Enter property name or value Email | Status | Name | Description | Actions cloudcomputing-345522@cloudcomputing-345522.iam.gserviceaccount.com | ✓ | cloudcomputing-345522 | For lab 4

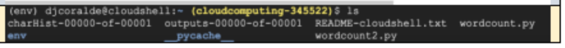
	Activate venv	`Python3 -m venv env`
		`Source env/bin/activate`
			
			`djcoralde@cloudshell:~ (cloudcomputing-345522)$ python3 -m venv env`
			`djcoralde@cloudshell:~ (cloudcomputing-345522)$ source env/bin/activate`
			`(env) djcoralde@cloudshell:~ (cloudcomputing-345522)$`

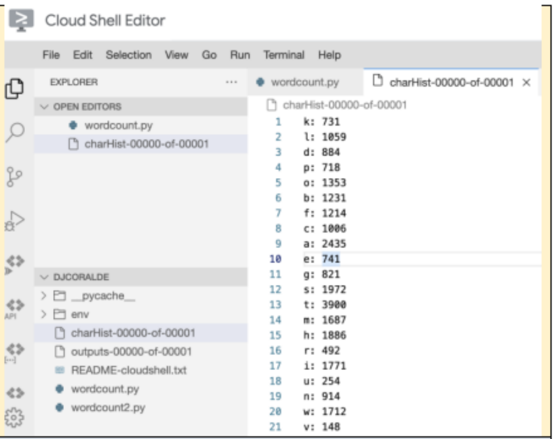
Install libraries	Upgrade pip	`pip install pip --upgrade`	
	Install apache-beam[gcp]	`pip install 'apache-beam[gcp]'`	
Run wordcount.py locally	Run code	`python -m apache_beam.examples.wordcount --output outputs`	
	View output file	`more outputs*`	
Run wordcount.py globally (over cloud)	Create cloud storage to be accessed (create storage bucket) To access storage bucket, use gs://<name_of_storage_bucket>		 cloudComputing Bucket details cloudcomputing-345522-gs Location | Storage class | Public access | Protection northamerica-northeast2 (Toronto) | Standard | Not public | None OBJECTS CONFIGURATION PERMISSIONS PROTECTION LIFECYCLE Buckets > cloudcomputing-345522-gs

	Set up PROJECT and BUCKET. These arguments will be used later when running the python script	`PROJECT = <name_of_project>`
		`BUCKET = gs://<link_to_bucket>`
			
			`(env) djcoralde@cloudshell:~ (cloudcomputing-345522)$ PROJECT=cloudcomputing-345522`
			`(env) djcoralde@cloudshell:~ (cloudcomputing-345522)$ echo PROJECT`
			`PROJECT`
			`(env) djcoralde@cloudshell:~ (cloudcomputing-345522)$ echo $PROJECT`
			`cloudcomputing-345522`
			`(env) djcoralde@cloudshell:~ (cloudcomputing-345522)$ BUCKET=gs://cloudcomputing-345522-gs`
			`(env) djcoralde@cloudshell:~ (cloudcomputing-345522)$ echo $BUCKET`
			`gs://cloudcomputing-345522-gs`

	Run	`python wordcount.py --region northamerica-northeast2 --runner DataflowRunner project $PROJECT --temp_location $BUCKET/tmp/ --output $BUCKET/result/outputs --experiment use_unsupported_python_version`	
			`(env) djcoralde@cloudshell:~ (cloudcomputing-345522)$ python -m apache_beam.examples.wordcount \`
			`> --project $PROJECT \`
			`> --region northamerica-northeast2 \`
			`> --runner dataflowRunner \`
			`> --temp_location $BUCKET/tmp \`
			`> --output $BUCKET/result/outputs`
			.. when finished
			
			`INFO:apache_beam.runners.dataflow.dataflow_runner:2022-03-28T23:12:22.366Z: JOB_MESSAGE_DETAILED: Cleaning up.`
			`INFO:apache_beam.runners.dataflow.dataflow_runner:2022-03-28T23:12:22.386Z: JOB_MESSAGE_DEBUG: Starting worker pool teardown.`
			`INFO:apache_beam.runners.dataflow.dataflow_runner:2022-03-28T23:12:22.394Z: JOB_MESSAGE_BASIC: Stopping worker pool...`
			`INFO:apache_beam.runners.dataflow.dataflow_runner:2022-03-28T23:14:42.702Z: JOB_MESSAGE_DETAILED: Autoscaling: Resized worker pool from 1 to 0.`
			`INFO:apache_beam.runners.dataflow.dataflow_runner:2022-03-28T23:14:42.720Z: JOB_MESSAGE_BASIC: Worker pool stopped.`
			`INFO:apache_beam.runners.dataflow.dataflow_runner:2022-03-28T23:14:42.730Z: JOB_MESSAGE_DEBUG: Tearing down pending resources...`
			`INFO:apache_beam.runners.dataflow.dataflow_runner:Job 2022-03-28_16_08_32-9574499114114741645 is in state JOB_STATE_DONE`
			`(env) djcoralde@cloudshell:~ (cloudcomputing-345522)$`

	View in dataflow		 Jobs Running | Filter Filter jobs Name | Type | End time | Elapsed time | Start time | Status | SDK version | ID | Region beamapp-djcoralde-0328230827-376019-abay4gbb | Batch | Mar 28, 2022, 7:15:17 PM | 6 min 44 sec | Mar 28, 2022, 7:08:33 PM | Succeeded | 2.37.0 | 2022-03-28_16_08_32-9574499114114741645 | northamerica-northeast2
	See in cloud storage		 cloudcomputing-345522-gs Location | Storage class | Public access | Protection northamerica-northeast2 (Toronto) | Standard | Not public | None OBJECTS CONFIGURATION PERMISSIONS PROTECTION LIFECYCLE Buckets > cloudcomputing-345522-gs > result Filter by name prefix only | Filter Filter objects and folders | Show deleted data Name | Size | Type | Created | Storage class | Last modified | Public access | Version history outputs-00000-of-00001 | 47.8 KB | text/plain | Mar 28, 2... | Standard | Mar 28, 20... | Not public | —

	Run	`python -m wordcount2 --output outputs --output2 charHist`	
			`(env) djcoralde@cloudshell:~ (cloudcomputing-345522)$ ls`
			`charHist-00000-of-00001  outputs-00000-of-00001  README-cloudshell.txt  wordcount.py`
			`env                      __pycache__             wordcount2.py`

Run wordcount2.py locally	See in editor		 Cloud Shell Editor File Edit Selection View Go Run Terminal Help EXPLORER OPEN EDITORS wordcount.py charHist-00000-of-00001 DJCORALDE __pycache__ env charHist-00000-of-00001 outputs-00000-of-00001 README-cloudshell.txt wordcount.py wordcount2.pycharHist-00000-of-00001 k: 731 l: 1059 d: 884 p: 718 o: 1353 b: 1231 f: 1214 c: 1006 a: 2435 e: 741 g: 821 s: 1972 t: 3900 m: 1687 h: 1886 r: 492 i: 1771 u: 254 n: 914 w: 1712 v: 148

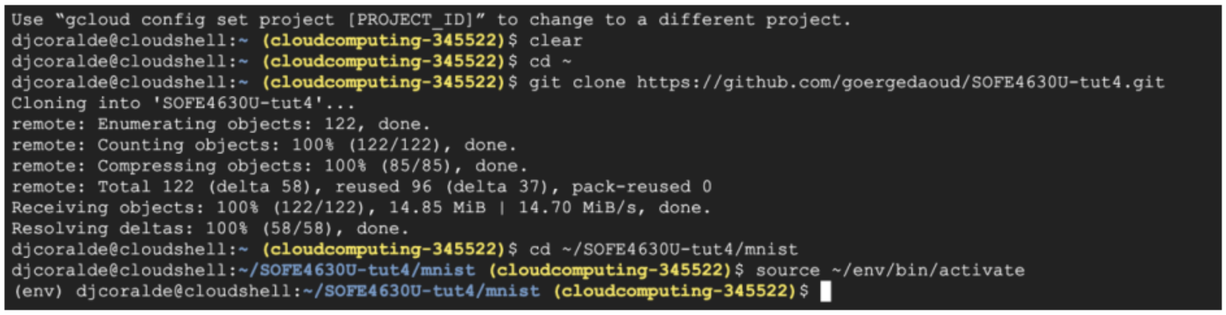
Run wordcount2.py over cloud	Set up arguments	`PROJECT=$(gcloud config list project --format "value(core.project)")`
		`echo $PROJECT`
		`BUCKET=gs://$PROJECT-gs`
		`echo $BUCKET`	

	Run	`python wordcount2.py --region northamerica-northeast2 --runner DataflowRunner --project $PROJECT --temp_location $BUCKET/tmp/ --input gs://dataflow-samples/shakespeare/winterstale.txt --output $BUCKET/result/outputs --output2 $BUCKET/result/outputs2 --experiment use_unsupported_python_version`	

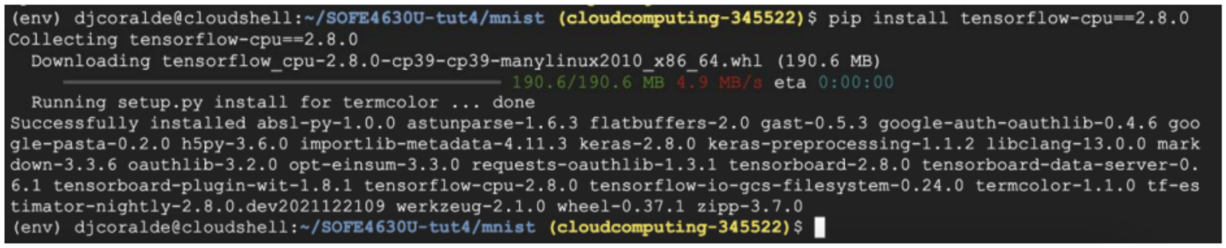
4. **Follow the following videos for various Dataflow examples for Batch and stream processing for the mnist dataset for various source and destination types; text file, MySQL database, and Kafka topics.**

   https://www.youtube.com/watch?v=9ZDj9KDGtEs

   Mnist dataset are handwritten images.

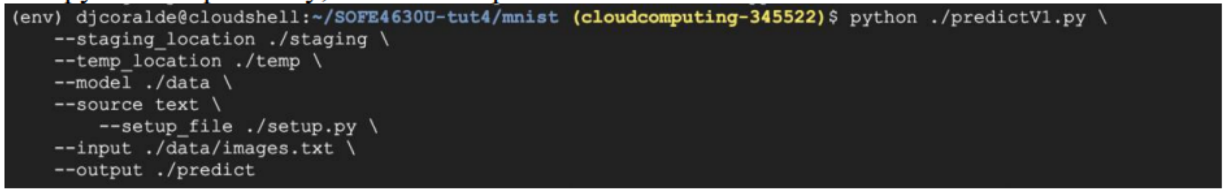


```
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
djcoralde@cloudshell:~ (cloudcomputing-345522)$ clear
djcoralde@cloudshell:~ (cloudcomputing-345522)$ cd ~
djcoralde@cloudshell:~ (cloudcomputing-345522)$ git clone https://github.com/goergedaoud/SOFE4630U-tut4.git
Cloning into 'SOFE4630U-tut4'...
remote: Enumerating objects: 122, done.
remote: Counting objects: 100% (122/122), done.
remote: Compressing objects: 100% (85/85), done.
remote: Total 122 (delta 58), reused 96 (delta 37), pack-reused 0
Receiving objects: 100% (122/122), 14.85 MiB | 14.70 MiB/s, done.
Resolving deltas: 100% (58/58), done.
djcoralde@cloudshell:~ (cloudcomputing-345522)$ cd ~/SOFE4630U-tut4/mnist
djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ source ~/env/bin/activate
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$
```

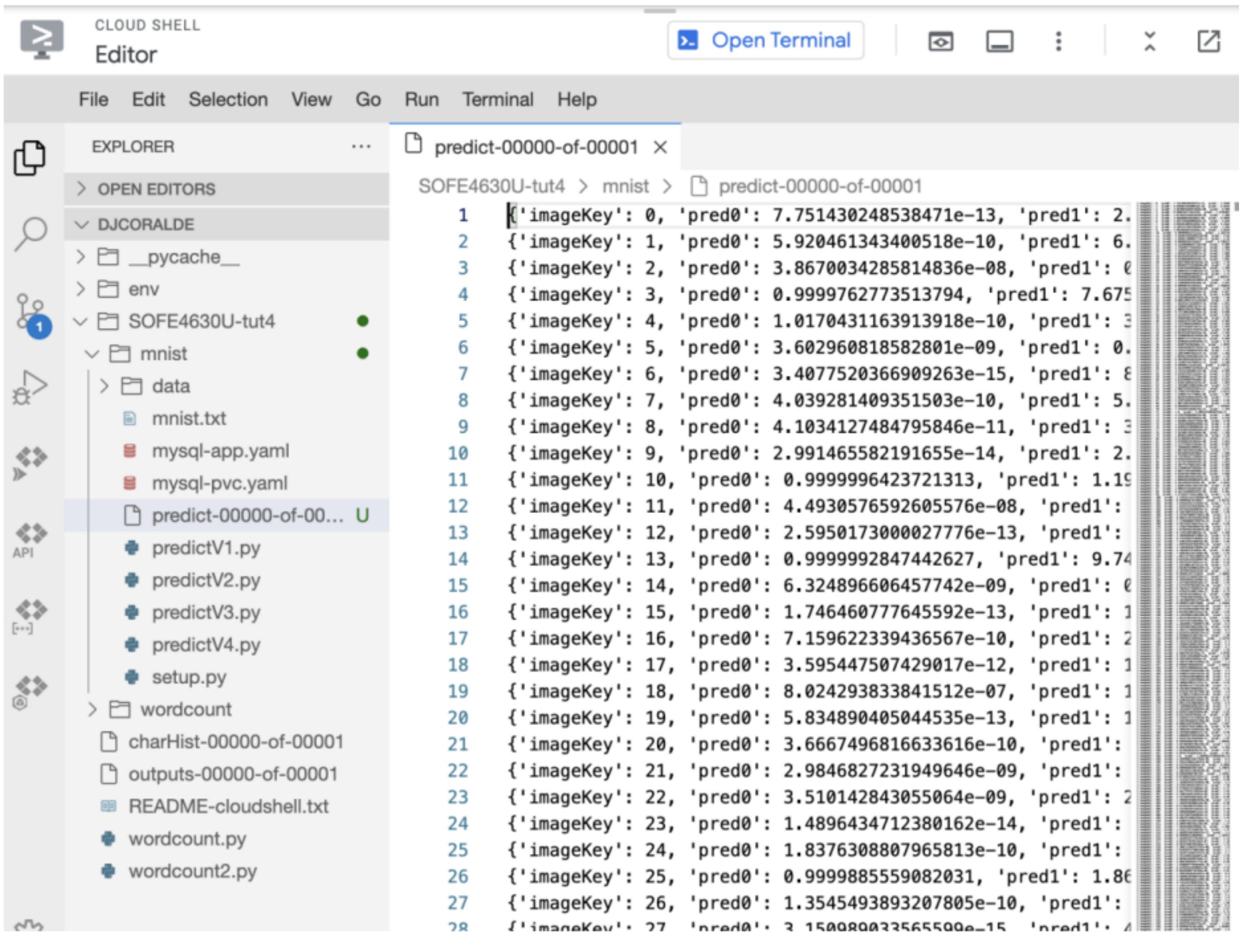


```
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ pip install tensorflow-cpu==2.8.0
Collecting tensorflow-cpu==2.8.0
  Downloading tensorflow_cpu-2.8.0-cp39-cp39-manylinux2010_x86_64.whl (190.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 190.6/190.6 MB 4.9 MB/s eta 0:00:00
  Running setup.py install for termcolor ... done
Successfully installed absl-py-1.0.0 astunparse-1.6.3 flatbuffers-2.0 gast-0.5.3 google-auth-oauthlib-0.4.6 goo
gle-pasta-0.2.0 h5py-3.6.0 importlib-metadata-4.11.3 keras-2.8.0 keras-preprocessing-1.1.2 libclang-13.0.0 mark
down-3.3.6 oauthlib-3.2.0 opt-einsum-3.3.0 requests-oauthlib-1.3.1 tensorboard-2.8.0 tensorboard-data-server-0.
6.1 tensorboard-plugin-wit-1.8.1 tensorflow-cpu-2.8.0 tensorflow-io-gcs-filesystem-0.24.0 termcolor-1.1.0 tf-es
timator-nightly-2.8.0.dev2021122109 werkzeug-2.1.0 wheel-0.37.1 zipp-3.7.0
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$
```

Run python script locally, see local output file in cloud editor

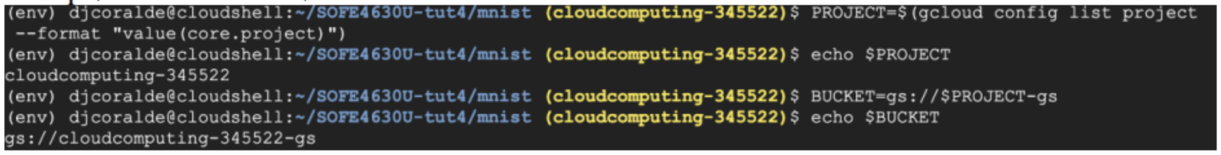


```
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ python ./predictV1.py \
    --staging_location ./staging \
    --temp_location ./temp \
    --model ./data \
    --source text \
        --setup_file ./setup.py \
    --input ./data/images.txt \
    --output ./predict
```



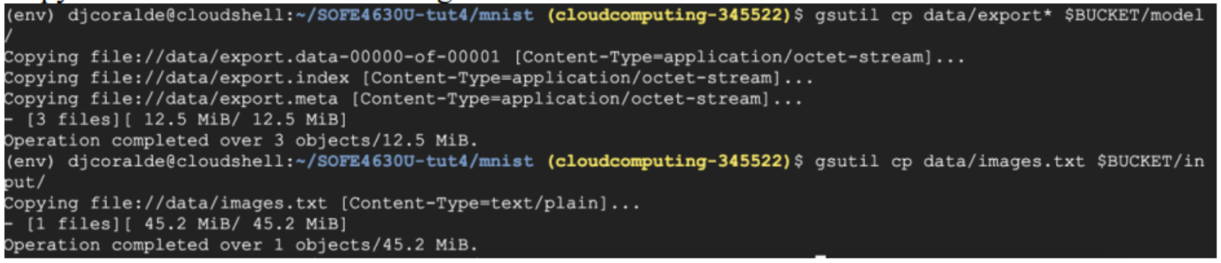
```
{'imageKey': 0, 'pred0': 7.751430248538471e-13, 'pred1': 2.
{'imageKey': 1, 'pred0': 5.920461343400518e-10, 'pred1': 6.
{'imageKey': 2, 'pred0': 3.8670034285814836e-08, 'pred1': 0
{'imageKey': 3, 'pred0': 0.9999762773513794, 'pred1': 7.675
{'imageKey': 4, 'pred0': 1.0170431163913918e-10, 'pred1': 3
{'imageKey': 5, 'pred0': 3.602960818582801e-09, 'pred1': 0.
{'imageKey': 6, 'pred0': 3.4077520366909263e-15, 'pred1': 8
{'imageKey': 7, 'pred0': 4.039281409351503e-10, 'pred1': 5.
{'imageKey': 8, 'pred0': 4.1034127484795846e-11, 'pred1': 3
{'imageKey': 9, 'pred0': 2.991465582191655e-14, 'pred1': 2.
{'imageKey': 10, 'pred0': 0.9999996423721313, 'pred1': 1.19
{'imageKey': 11, 'pred0': 4.4930576592605576e-08, 'pred1':
{'imageKey': 12, 'pred0': 2.5950173000027776e-13, 'pred1':
{'imageKey': 13, 'pred0': 0.9999992847442627, 'pred1': 9.74
{'imageKey': 14, 'pred0': 6.324896606457742e-09, 'pred1': 0
{'imageKey': 15, 'pred0': 1.746460777645592e-13, 'pred1': 1
{'imageKey': 16, 'pred0': 7.159622339436567e-10, 'pred1': 2
{'imageKey': 17, 'pred0': 3.595447507429017e-12, 'pred1': 1
{'imageKey': 18, 'pred0': 8.024293833841512e-07, 'pred1': 1
{'imageKey': 19, 'pred0': 5.834890405044535e-13, 'pred1': 1
{'imageKey': 20, 'pred0': 3.6667496816633616e-10, 'pred1':
{'imageKey': 21, 'pred0': 2.9846827231949646e-09, 'pred1':
{'imageKey': 22, 'pred0': 3.510142843055064e-09, 'pred1': 2
{'imageKey': 23, 'pred0': 1.4896434712380162e-14, 'pred1':
{'imageKey': 24, 'pred0': 1.8376308807965813e-10, 'pred1':
{'imageKey': 25, 'pred0': 0.9999885559082031, 'pred1': 1.86
{'imageKey': 26, 'pred0': 1.3545493893207805e-10, 'pred1':
{'imageKey': 27, 'pred0': 3.150989033565599e-15, 'pred1': 4
```

Set up $PROJECT and $BUCKET



```
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ PROJECT=$(gcloud config list project
 --format "value(core.project)")
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ echo $PROJECT
cloudcomputing-345522
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ BUCKET=gs://$PROJECT-gs
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ echo $BUCKET
gs://cloudcomputing-345522-gs
```

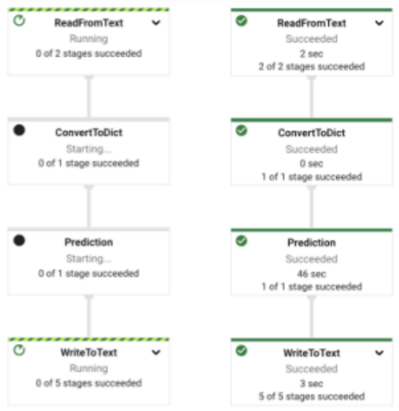
Copy files from data to cloud storage



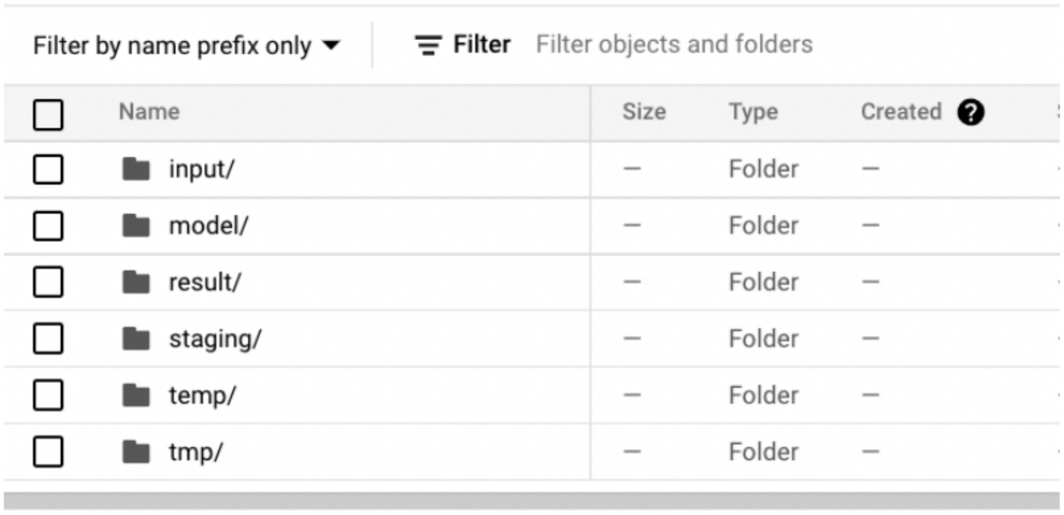
```
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ gsutil cp data/export* $BUCKET/model
/
Copying file://data/export.data-00000-of-00001 [Content-Type=application/octet-stream]...
Copying file://data/export.index [Content-Type=application/octet-stream]...
Copying file://data/export.meta [Content-Type=application/octet-stream]...
- [3 files][ 12.5 MiB/ 12.5 MiB]
Operation completed over 3 objects/12.5 MiB.
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ gsutil cp data/images.txt $BUCKET/in
put/
Copying file://data/images.txt [Content-Type=text/plain]...
- [1 files][ 45.2 MiB/ 45.2 MiB]
Operation completed over 1 objects/45.2 MiB.
```

Run code over cloud

- See job in dataflow>jobs

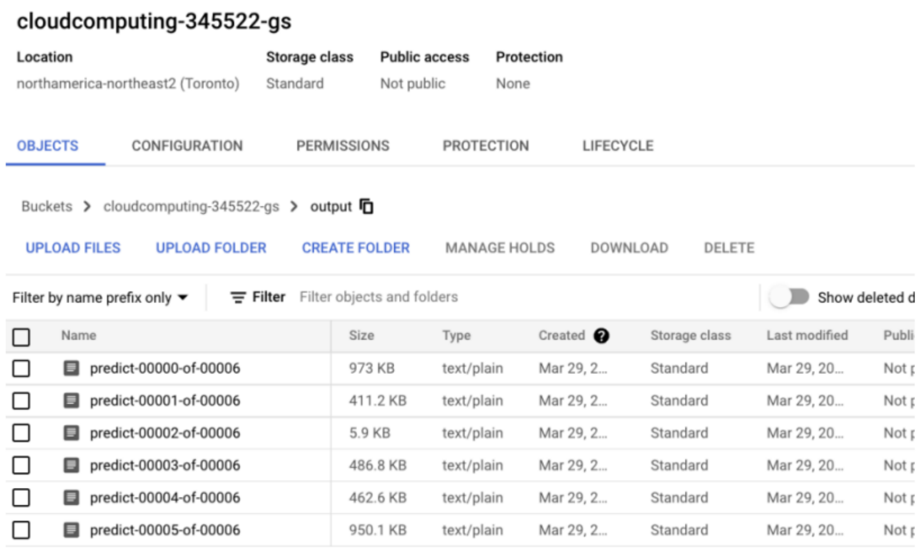


- See created files in cloud storage



Filter by name prefix only ▾ | Filter | Filter objects and folders

☐	Name	Size	Type	Created
☐	input/	—	Folder	—
☐	model/	—	Folder	—
☐	result/	—	Folder	—
☐	staging/	—	Folder	—
☐	temp/	—	Folder	—
☐	tmp/	—	Folder	—



cloudcomputing-345522-gs

Location	Storage class	Public access	Protection
northamerica-northeast2 (Toronto)	Standard	Not public	None

OBJECTS CONFIGURATION PERMISSIONS PROTECTION LIFECYCLE

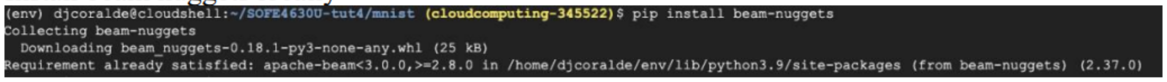
Buckets > cloudcomputing-345522-gs > output

UPLOAD FILES UPLOAD FOLDER CREATE FOLDER MANAGE HOLDS DOWNLOAD DELETE

Filter by name prefix only ▾ | Filter | Filter objects and folders | Show deleted d

☐	Name	Size	Type	Created	Storage class	Last modified	Publi
☐	predict-00000-of-00006	973 KB	text/plain	Mar 29, 2...	Standard	Mar 29, 20...	Not p
☐	predict-00001-of-00006	411.2 KB	text/plain	Mar 29, 2...	Standard	Mar 29, 20...	Not p
☐	predict-00002-of-00006	5.9 KB	text/plain	Mar 29, 2...	Standard	Mar 29, 20...	Not p
☐	predict-00003-of-00006	486.8 KB	text/plain	Mar 29, 2...	Standard	Mar 29, 20...	Not p
☐	predict-00004-of-00006	462.6 KB	text/plain	Mar 29, 2...	Standard	Mar 29, 20...	Not p
☐	predict-00005-of-00006	950.1 KB	text/plain	Mar 29, 2...	Standard	Mar 29, 20...	Not p

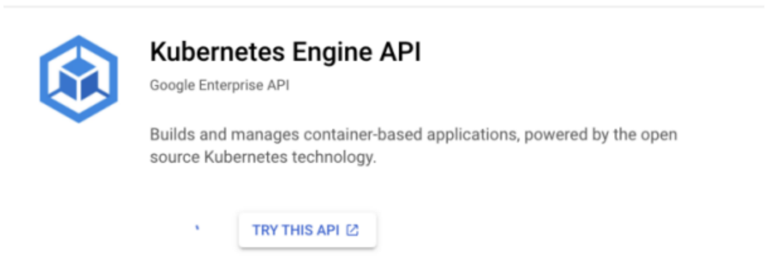
Install beam nuggets locally



```
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ pip install beam-nuggets
Collecting beam-nuggets
  Downloading beam_nuggets-0.18.1-py3-none-any.whl (25 kB)
Requirement already satisfied: apache-beam<3.0.0,>=2.8.0 in /home/djcoralde/env/lib/python3.9/site-packages (from beam-nuggets) (2.37.0)
```

Set up Kubernetes Engine

- Enable

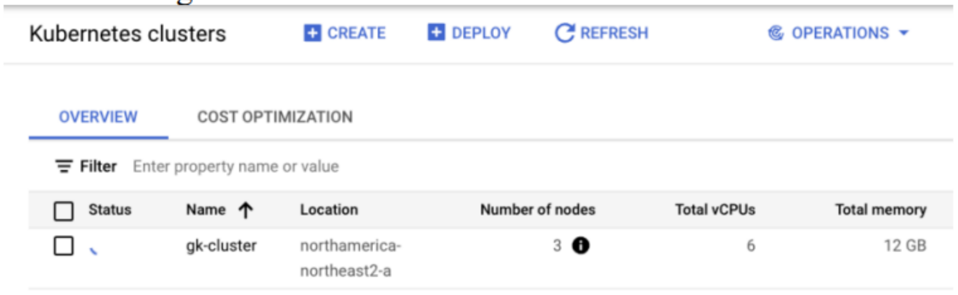


Kubernetes Engine API

Google Enterprise API

Builds and manages container-based applications, powered by the open source Kubernetes technology.

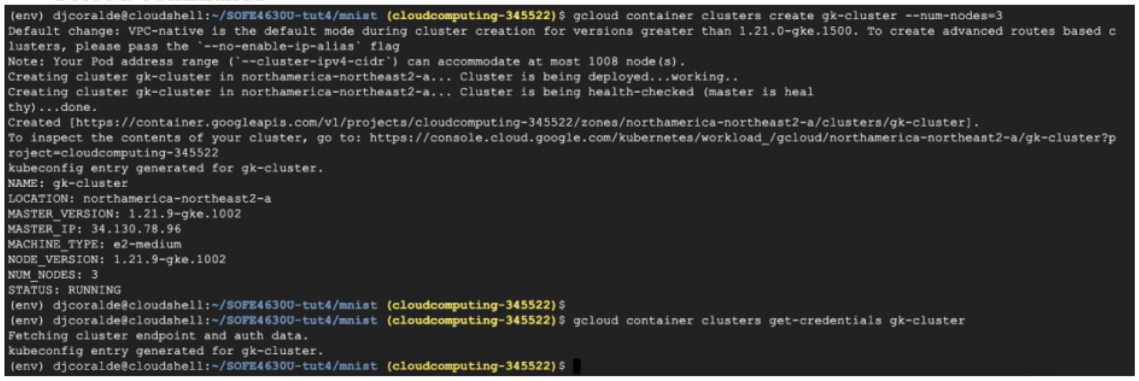
- Create cluster using commands



Kubernetes clusters

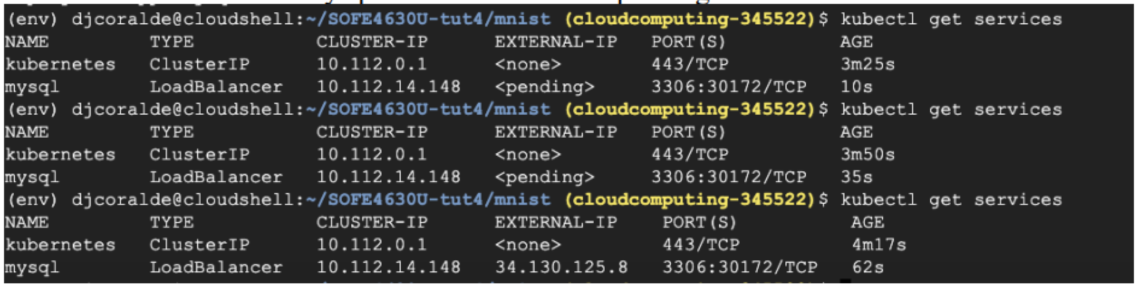
Status	Name	Location	Number of nodes	Total vCPUs	Total memory
	gk-cluster	northamerica-northeast2-a	3	6	12 GB

- Gcloud commands



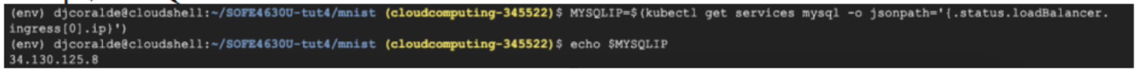
```
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ gcloud container clusters create gk-cluster --num-nodes=3
Default change: VPC-native is the default mode during cluster creation for versions greater than 1.21.0-gke.1500. To create advanced routes based c
lusters, please pass the `--no-enable-ip-alias` flag
Note: Your Pod address range (`--cluster-ipv4-cidr`) can accommodate at most 1008 node(s).
Creating cluster gk-cluster in northamerica-northeast2-a... Cluster is being deployed...working..
Creating cluster gk-cluster in northamerica-northeast2-a... Cluster is being health-checked (master is heal
thy)...done.
Created [https://container.googleapis.com/v1/projects/cloudcomputing-345522/zones/northamerica-northeast2-a/clusters/gk-cluster].
To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/workload_/gcloud/northamerica-northeast2-a/gk-cluster?p
roject=cloudcomputing-345522
kubeconfig entry generated for gk-cluster.
NAME: gk-cluster
LOCATION: northamerica-northeast2-a
MASTER_VERSION: 1.21.9-gke.1002
MASTER_IP: 34.130.78.96
MACHINE_TYPE: e2-medium
NODE_VERSION: 1.21.9-gke.1002
NUM_NODES: 3
STATUS: RUNNING
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ gcloud container clusters get-credentials gk-cluster
Fetching cluster endpoint and auth data.
kubeconfig entry generated for gk-cluster.
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$
```

Wait until external IP for mysql has value instead of pending.



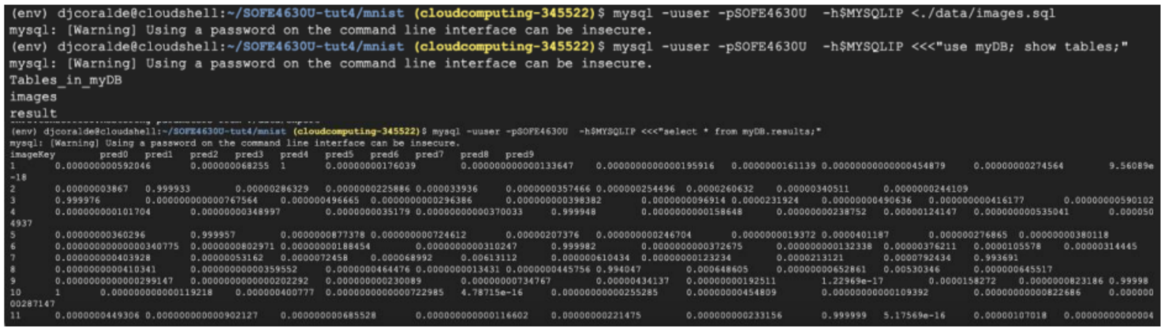
```
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ kubectl get services
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP    PORT(S)          AGE
kubernetes   ClusterIP      10.112.0.1      <none>         443/TCP          3m25s
mysql        LoadBalancer   10.112.14.148   <pending>      3306:30172/TCP   10s
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ kubectl get services
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP    PORT(S)          AGE
kubernetes   ClusterIP      10.112.0.1      <none>         443/TCP          3m50s
mysql        LoadBalancer   10.112.14.148   <pending>      3306:30172/TCP   35s
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ kubectl get services
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP    PORT(S)           AGE
kubernetes   ClusterIP      10.112.0.1      <none>          443/TCP           4m17s
mysql        LoadBalancer   10.112.14.148   34.130.125.8    3306:30172/TCP    62s
```

Set up $MYSQLIP



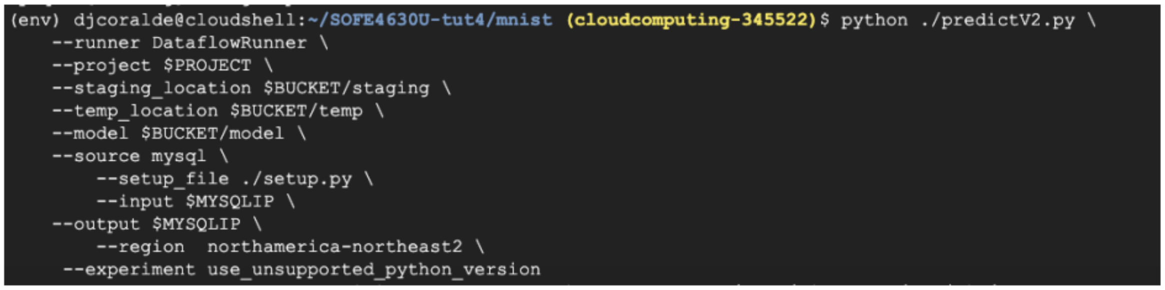
```
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ MYSQLIP=$(kubectl get services mysql -o jsonpath='{.status.loadBalancer.
ingress[0].ip}')
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ echo $MYSQLIP
34.130.125.8
```

MySQL commands



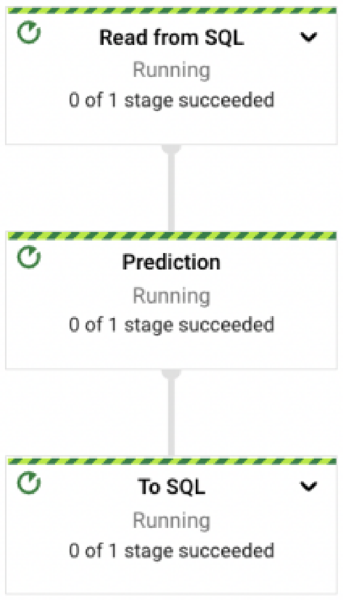
```
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ mysql -uuser -pSOFE4630U  -h$MYSQLIP <./data/images.sql
mysql: [Warning] Using a password on the command line interface can be insecure.
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ mysql -uuser -pSOFE4630U  -h$MYSQLIP <<<"use myDB; show tables;"
mysql: [Warning] Using a password on the command line interface can be insecure.
Tables_in_myDB
images
result
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ mysql -uuser -pSOFE4630U  -h$MYSQLIP <<<"select * from myDB.results;"
mysql: [Warning] Using a password on the command line interface can be insecure.
imageKey        pred0   pred1   pred2   pred3   pred4   pred5   pred6   pred7   pred8   pred9
1       0.000000000592046       0.000000068255  1       0.00000000176039        0.00000000000133647     0.0000000000000195916   0.0000000161139 0.0000000000000454879   0.0000000274564         9.56089e
-18
2       0.0000003867    0.999933        0.00000286329   0.000000225886 0.000033936      0.000000357466 0.000000254496  0.0000260632    0.00000340511       0.0000000244109
3       0.999976        0.0000000000000767564   0.000000496665  0.000000000296386       0.000000000398382       0.000000096914 0.0000231924    0.0000000490636   0.00000000416177        0.0000000590102
4       0.000000000101704       0.00000000348997        0.0000000035179 0.0000000000370033      0.999948        0.000000000158648       0.00000000238752   0.00000124147   0.000000000535041       0.000050
4937
5       0.0000000360296         0.999957        0.0000000877378 0.000000000724612       0.0000207376   0.000000000246704       0.000000019372 0.0000401187       0.000000276865  0.0000000380118
6       0.00000000000000340775  0.0000000802971 0.00000000188454        0.000000000310247       0.999982        0.000000000372675       0.000000000132338  0.00000376211   0.0000105578    0.00000314445
7       0.00000000403928        0.00000053162   0.0000072458    0.000068992     0.00613112      0.000000610434  0.00000000123234        0.0000213121      0.0000792434    0.993691
8       0.000000000410341       0.0000000000359552      0.000000464476 0.000000013431 0.000000445756 0.994047       0.000648605     0.00000000652861  0.00530346      0.000000645517
9       0.0000000000000299147   0.0000000000000202292   0.000000000230089       0.0000000734767         0.00000434137   0.00000000192511        1.22969e-17        0.0000158272    0.000000823186 0.99998
10      1       0.00000000000119218     0.000000400777  0.0000000000000722985   4.78715e-16     0.0000000000255285      0.00000000454809        0.0000000000109392        0.0000000000822686      0.000000
00287147
11      0.000000449306 0.00000000000902127      0.00000000685528        0.00000000000116602     0.0000000221475         0.000000000233156       0.999999  5.17569e-16     0.00000107018   0.0000000000000004
```

Run over cloud
- Command



```
(env) djcoralde@cloudshell:~/SOFE4630U-tut4/mnist (cloudcomputing-345522)$ python ./predictV2.py \
   --runner DataflowRunner \
   --project $PROJECT \
   --staging_location $BUCKET/staging \
   --temp_location $BUCKET/temp \
   --model $BUCKET/model \
   --source mysql \
       --setup_file ./setup.py \
       --input $MYSQLIP \
   --output $MYSQLIP \
       --region  northamerica-northeast2 \
    --experiment use_unsupported_python_version
```

- See DataFlow>Jobs (**got stuck here**)



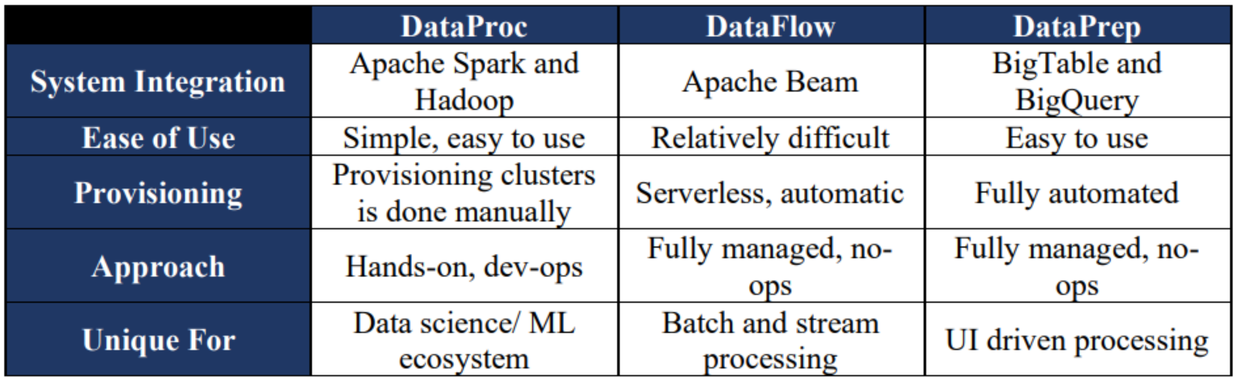
5. **(Optional) The following video describes how to use BigQuery and Google PubSub as sources and destinations for the Dataflow pipeline.**

6. **Google Cloud has another processing service called DataProc. Name another processing service that is usually used in the cloud environment (not necessarily GCP). Compare between it and both Dataflow and DataProc. Your comparison may include but is not limited to the major differences, advantages, disadvantages, and limitations.**

   A: The three most common data processing services that Google Cloud platform provides are DataFlow, DataPrep and DataProc.

   - DataFlow is a cloud-based data processing solution that can handle batch as well as real-time data streaming. It has capabilities such as resource auto-scaling and dynamic work rebalancing, as well as flexible scheduling and ready-to-use real-time AI patterns.
   - Google DataPrep is a data service that allows you to explore, clean, and prepare structured and unstructured data in the cloud. The main feachers are Active profiling, optimized processing throughput and Data quality rules.
   - Google Cloud DataProc is a managed service for Spark and ApacheHadoop jobs that enables batch processing, querying, streaming, and machine learning using open source data technologies.

   DataProc enables users to take advantage of open source tools for batch processing, querying, streaming, and machine learning [https://cloud.google.com/dataproc/docs/concepts/overview] To compare with DataProc and DataFlow, choose DataPrep. Below is a table comparing the three.



	DataProc	DataFlow	DataPrep
System Integration	Apache Spark and Hadoop	Apache Beam	BigTable and BigQuery
Ease of Use	Simple, easy to use	Relatively difficult	Easy to use
Provisioning	Provisioning clusters is done manually	Serverless, automatic	Fully automated
Approach	Hands-on, dev-ops	Fully managed, no-ops	Fully managed, no-ops
Unique For	Data science/ ML ecosystem	Batch and stream processing	UI driven processing

Google Cloud has another processing service called DataPrep.

- Dataflow
  - Streaming analytics service
  - Minimizes:
    - Latency
    - Cost
    - Processing time
  - Uses
    - Autoscaling
    - Batch processing
- DataProc
  - Highly scalable service
    - Runs:
      - Apache Spark
      - Apache Flink
      - Presto
- DataPrep
  - Cloud google data service
    - By Trifacta
    - Prepare data for
      - Analysis
      - Machine learning

Another processing service example is Trifacta which is under the google cloud platform. Some comparison is that the dataflow is an easier way to stream the analytic service which prides itself on reducing the cost, process as well as time/latency. Dataproc can scale the purpose and goal by itself. This allows multiple analytical and or data during its processing work.

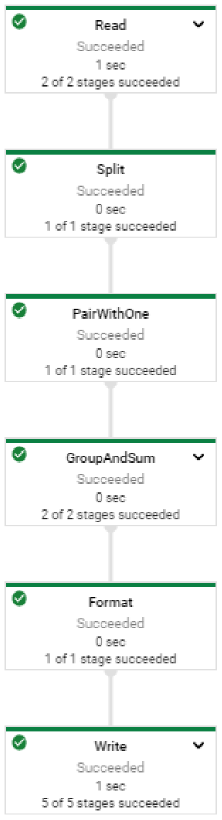
An advantage would have to be that the dataflow is able to specialize in allowing minimum aspects of the data processor which also allows for a cheaper price tag and its efficiency also increases. Dataproc's main advantages include its ability to manipulate data in an organized fashion. Additionally, it has a sophisticated ability to allow the acceptance of new data using ML.

The disadvantages of Dataflows include a lack of features that may impair their ability in the way it's supposed to adapt and be flexible, this is a disadvantage because other services may include these traits. Additionally, another disadvantage seen with Dataflow is its popularity, since it’s a fairly new concept, it hasn’t gained a mass following yet which makes it a little more difficult to find the answers online. Dataproc’s main disadvantage is mostly focused on its latency and processing due to its large-scale factor, however, this is balanced out because the system is affordable and gives companies a cheaper alternative.

7. **Suggest a practical application using both stream and batch processing that can be applied to a given dataset. It's expected to use the dataset uploaded in the third milestone but you can use any other dataset. If you decide to use another dataset, It should maintain both variety and huge volume. Your report should include but not limited to**

   Sample practical application

   - **The application**
     - Image/Face Recognition
       - Most common applications in machine learning
       - Uses both stream and batch processing
   - **Its impact.**
     - Identifies objects, faces, people, etc.
     - Apple uses face recognition technology in their products
     - Facebook uses face/image recognition technology to provide auto tagging feature
   - **The used dataset (size, schema/structure).**
     - Used dataset is a set of images in relation to the problem domain
       - Ex. for a facial recognition system: face images
         - Face images of triplets
         - Folders containing different facial expressions
   - **A graph showing the proposed pipeline(s).**



- **List of other tools (AI, clustering,…) needed to implement that application.**
    - Machine learning libraries
    - OpenFace
    - Facial_recognition library
    - OpenCV

Application focus: Self-driving cars

Dataset

- The dataset for autonomous driving cars will be taken from Google Open Images. The dataset includes both labeled and unlabeled objects such as cars, pedestrians, traffic lights, fire hydrants, buses, trucks, and signs, among other things. Over 400,000 photos are included in the dataset, which depict a wide range of environmental changes. Sensor data is also included in the dataset, which is essential for calculating the distance between the objects and the device. This dataset can be utilized in the creation and design of object detection systems.

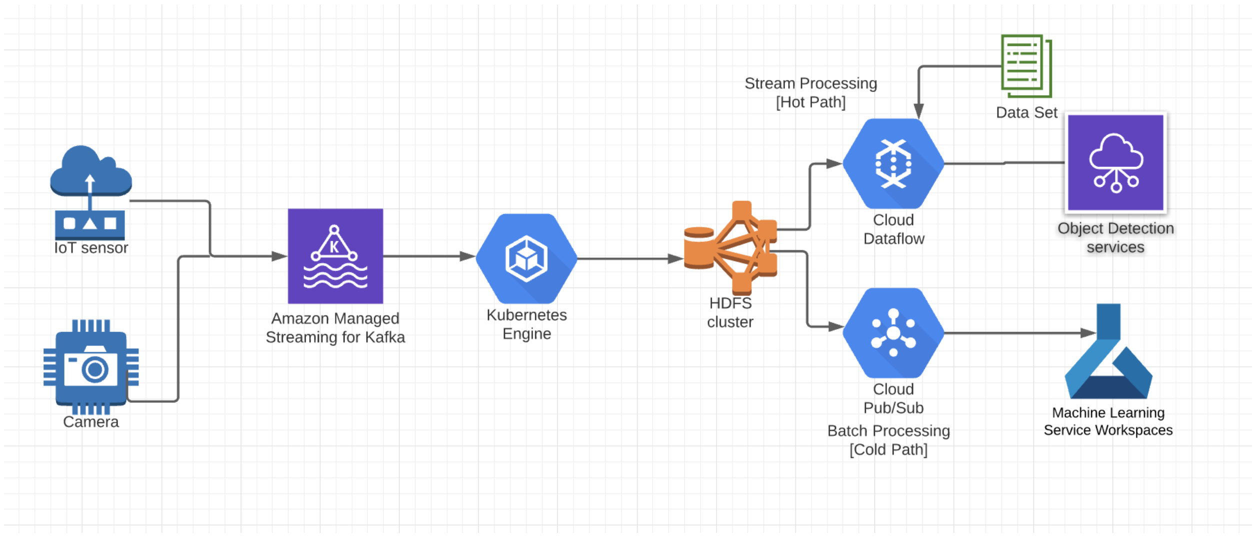
The application

- Using stream and batch processing, the above dataset will be used to create a portable object detection system that may be used in cars to assist drivers in driving safely by detecting objects in real time.

Its impact

- Object detection is becoming more important as the frequency of car accidents in Ontario continues to rise. According to statistics, there is a 26% increase in car accidents every year. As a result, object detection methods are required to resolve the dilemma. The device would employ Big Data to construct an object detection model that would be able to successfully detect objects and assist drivers in driving safely on the road.The steam processing [hot path] will be utilized to detect objects and alert users when they approach them. To accomplish this the system will use the linked sensors to process data such as distance and speed. As of batch processing [cold path], the primary goal is to gather data that can be used to improve object detection. It will also be utilized to provide system upgrades, which will improve the user experience.

A graph showing the proposed pipeline(s)



List of other tools

- Machine learning library for example tensorflow and tensorflow lite
- Big Data, Apache Beam, Dataflow
- HDFS for handling large data sets running on commodity hardware