



Cloud Computing
Lab 4: Batch and Stream processing

Jane Coralde | 100660214
March 29, 2022

Objective:

1. Get familiar with Dataflow
2. Understand MapReduce.
3. Run batch and Stream Processing examples over GCP.

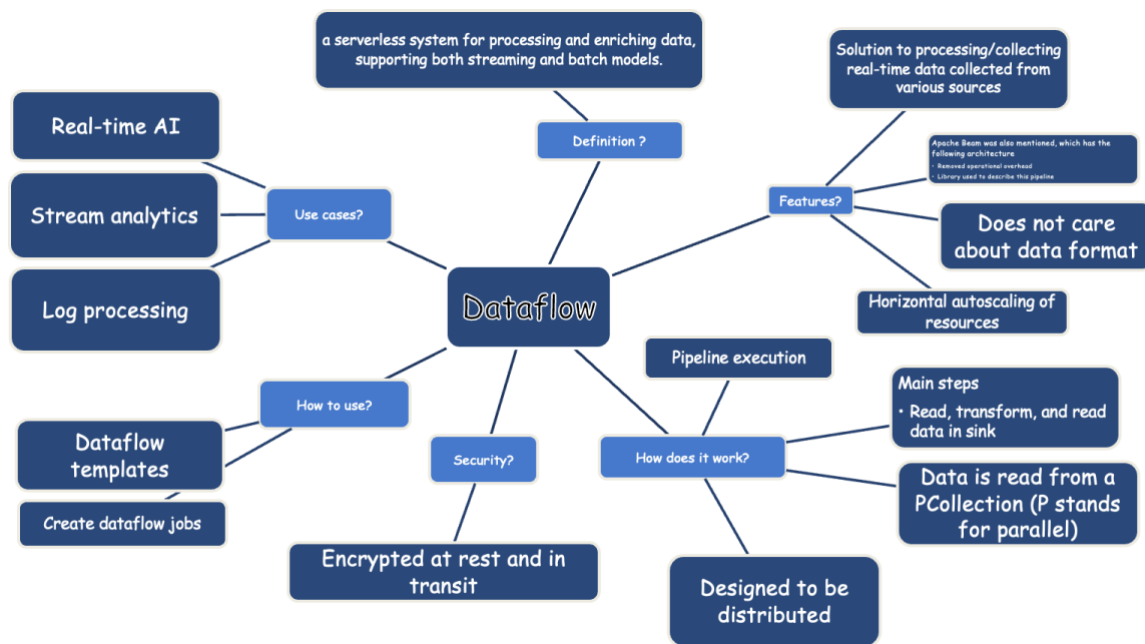
Lab4 Repository: <https://github.com/goergedaoud/SOFE4630U-tut4.git> (Links to an external site.)

Procedure: There are 7 items/tasks given in Lab4. Answer all.

1 Q and A

Q: Watch the following [video](#) about Google Cloud Dataflow

A: Please see mindmap created below.



2 Q and A

Q: Watch the following [video](#) describing how to apply MapReduce to count the words within a certain document

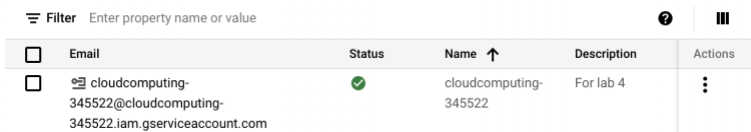
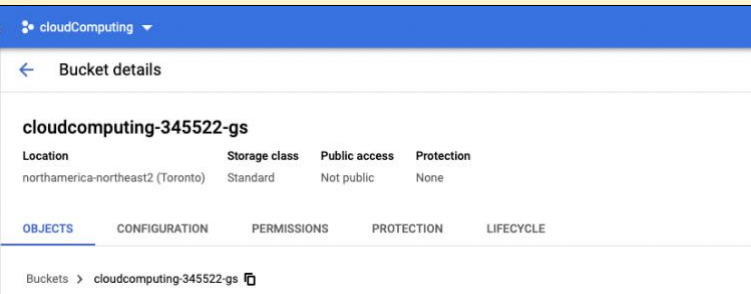
A: The second output added in wordcount2.py is the part which applies MapReduce.

3 Coding Exercise

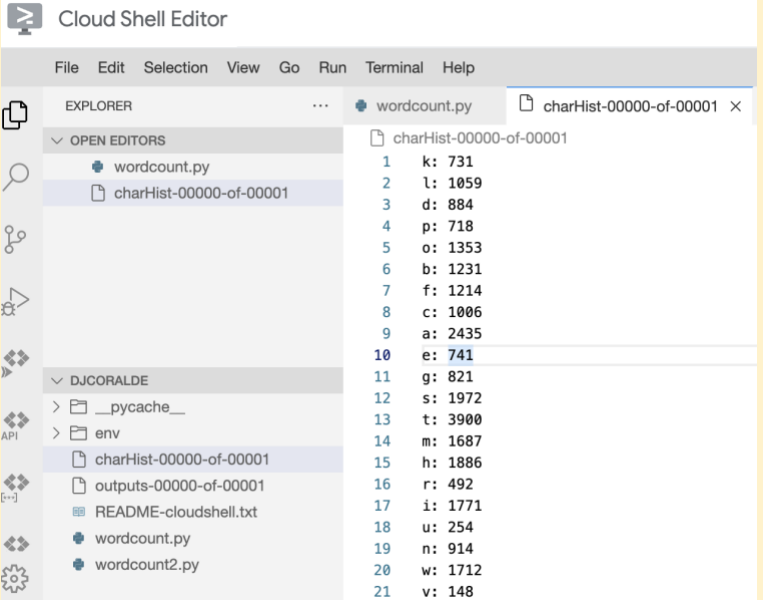
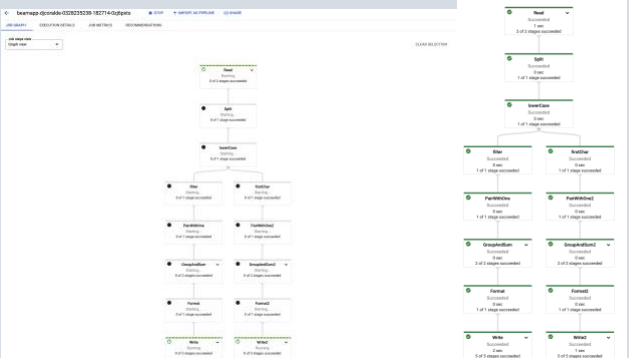
Follow the following [video](#) to set up the GCP environment for Dataflow and **run wordcount examples**.

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		
	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>		
	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	<pre>python wordcount.py --region northamerica-northeast2 --runner DataflowRunner project \$PROJECT --temp_location \$BUCKET/tmp/ --output \$BUCKET/result/outputs --experiment use_unsupported_python_version</pre>	<pre>(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:2022-03-28T23:12:22.364Z: JOB_MESSAGE_DETAILED: Cleaning up. INFO:apache_beam.runners.dataflow.dataflow_runner:2022-03-28T23:12:22.384Z: 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: Realized worker pool from 1 to 0. INFO:apache_beam.runners.dataflow.dataflow_runner:2022-03-28T23:14:42.722Z: JOB_MESSAGE_BASIC: Worker pool stopped. INFO:apache_beam.runners.dataflow.dataflow_runner:2022-03-28T23:14:42.732Z: JOB_MESSAGE_DEBUG: Tearing down pending resources... INFO:apache_beam.runners.dataflow.dataflow_runner:2022-03-28_16_08_32-9574499114114741645 is in state JOB_STATE_DONE (env) djcoralde@cloudshell:~ (cloudcomputing-345522)\$</pre>																		
View in dataflow		<div>Jobs CREATE JOB FROM TEMPLATE ENABLE SORTING REFRESH LEARN</div> <div><input type="radio"/> Running Filter Filter jobs</div> <table><thead><tr><th>Name</th><th>Type</th><th>End time</th><th>Elapsed time</th><th>Start time</th><th>Status</th><th>SDK version</th><th>ID</th><th>Region</th></tr></thead><tbody><tr><td>beamapp-djcoralde-0328230827-376019-abay4gbb</td><td>Batch</td><td>Mar 28, 2022, 7:15:17 PM</td><td>6 min 44 sec</td><td>Mar 28, 2022, 7:08:33 PM</td><td>Succeeded</td><td>2.37.0</td><td>2022-03-28_16_08_32-9574499114114741645</td><td>northamerica-northeast2</td></tr></tbody></table>	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
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		<div>cloudcomputing-345522-gs</div> <div>Location: northamerica-northeast2 (Toronto) Storage class: Standard Public access: Not public Protection: None</div> <div>OBJECTS CONFIGURATION PERMISSIONS PROTECTION LIFECYCLE</div> <div>Buckets > cloudcomputing-345522-gs > result</div> <div>UPLOAD FILES UPLOAD FOLDER CREATE FOLDER MANAGE HOLDS DOWNLOAD DELETE</div> <div>Filter by name prefix only Filter Filter objects and folders Show deleted data</div> <table><thead><tr><th>Name</th><th>Size</th><th>Type</th><th>Created</th><th>Storage class</th><th>Last modified</th><th>Public access</th><th>Version history</th></tr></thead><tbody><tr><td>outputs-00000-of-00001</td><td>47.8 KB</td><td>text/plain</td><td>Mar 28, 2022</td><td>Standard</td><td>Mar 28, 2022</td><td>Not public</td><td>—</td></tr></tbody></table>	Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	outputs-00000-of-00001	47.8 KB	text/plain	Mar 28, 2022	Standard	Mar 28, 2022	Not public	—		
Name	Size	Type	Created	Storage class	Last modified	Public access	Version history													
outputs-00000-of-00001	47.8 KB	text/plain	Mar 28, 2022	Standard	Mar 28, 2022	Not public	—													
Run	<pre>python -m wordcount2 --output outputs --output2 charHist</pre>	<pre>(env) djcoralde@cloudshell:~ (cloudcomputing-345522)\$ ls charHist-00000-of-00001 outputs-00000-of-00001 README-cloudshell.txt wordcount.py env __pycache__ wordcount2.py</pre>																		

<p>Run wordcount2.py locally</p>	<p>See in editor</p>		
<p>Run wordcount2.py over cloud</p>	<p>Set up arguments</p>	<pre>PROJECT=\$(gcloud config list project --format "value(core.project)") echo \$PROJECT BUCKET=gs://\$PROJECT-gs echo \$BUCKET</pre>	
	<p>Run</p>	<pre>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</pre>	

4 Q and A

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

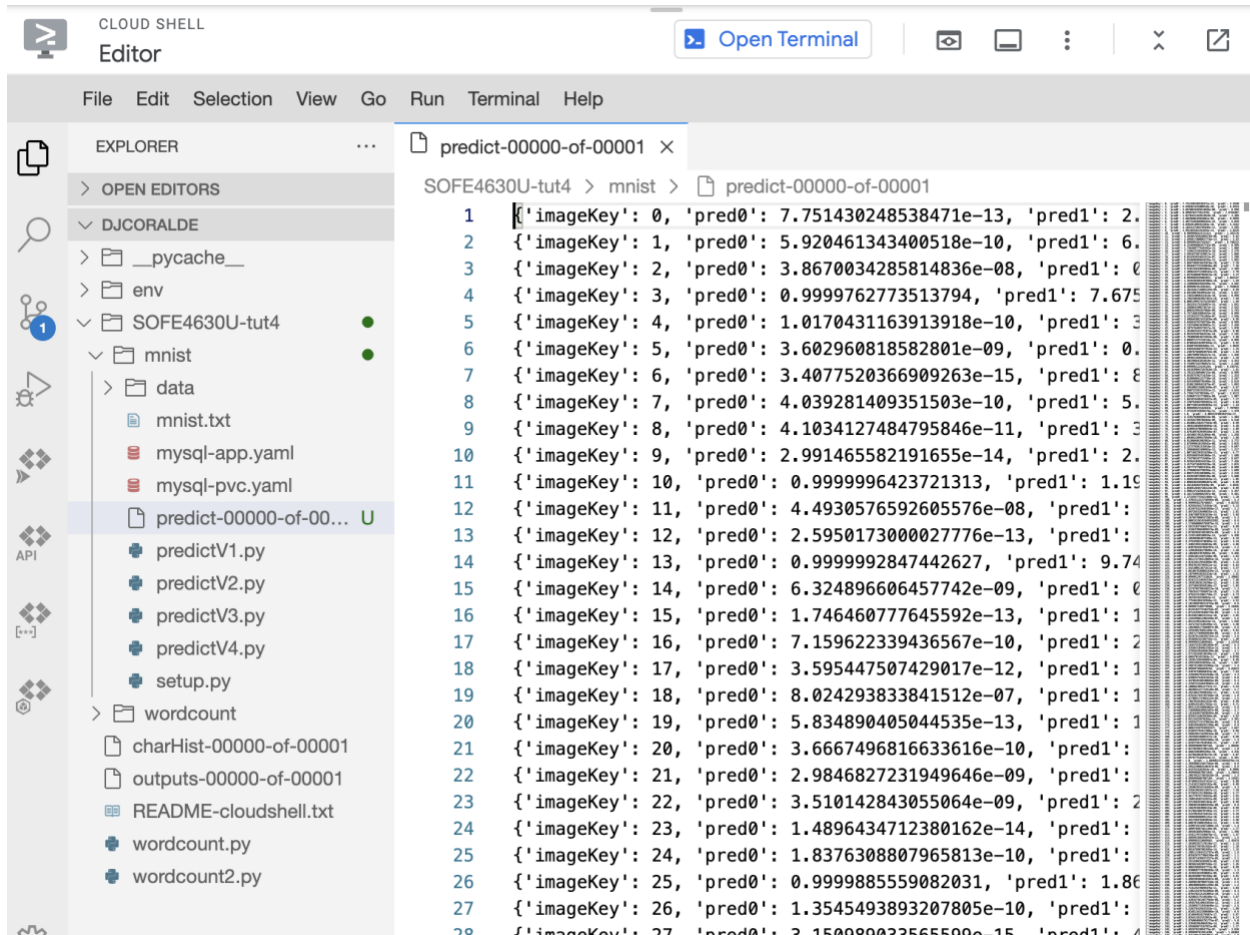
Mnist dataset are handwritten images.

```
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 google-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-estimator-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
```



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/input/
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				
<input type="checkbox"/>	Name	Size	Type	Created ?
<input type="checkbox"/>	input/	—	Folder	—
<input type="checkbox"/>	model/	—	Folder	—
<input type="checkbox"/>	result/	—	Folder	—
<input type="checkbox"/>	staging/	—	Folder	—
<input type="checkbox"/>	temp/	—	Folder	—
<input type="checkbox"/>	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					<input type="checkbox"/> Show deleted d		
<input type="checkbox"/>	Name	Size	Type	Created ?	Storage class	Last modified	Publi
<input type="checkbox"/>	predict-00000-of-00006	973 KB	text/plain	Mar 29, 2...	Standard	Mar 29, 20...	Not p
<input type="checkbox"/>	predict-00001-of-00006	411.2 KB	text/plain	Mar 29, 2...	Standard	Mar 29, 20...	Not p
<input type="checkbox"/>	predict-00002-of-00006	5.9 KB	text/plain	Mar 29, 2...	Standard	Mar 29, 20...	Not p
<input type="checkbox"/>	predict-00003-of-00006	486.8 KB	text/plain	Mar 29, 2...	Standard	Mar 29, 20...	Not p
<input type="checkbox"/>	predict-00004-of-00006	462.6 KB	text/plain	Mar 29, 2...	Standard	Mar 29, 20...	Not p
<input type="checkbox"/>	predict-00005-of-00006	950.1 KB	text/plain	Mar 29, 2...	Standard	Mar 29, 20...	Not p

Install beam nuggets locally

```
(env) djcoral@cloudshell:~/SOF24630U-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/djcoral/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.

[TRY THIS API](#)

- Create cluster using commands

Kubernetes clusters

+

CREATE

+

DEPLOY

↻

REFRESH

⚙

OPERATIONS

OVERVIEW

COST OPTIMIZATION

Filter

 Enter property name or value

<input type="checkbox"/>	Status	Name	Location	Number of nodes	Total vCPUs	Total memory
<input type="checkbox"/>		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 0.00000000592046 0.000000068255 1 0.0000000176039 0.00000000000133647 0.000000000000195916 0.0000000161139 0.00000000000000454879 0.00000000274564 9.56089e
-18
2 0.00000003867 0.999933 0.00000286329 0.000000225886 0.000033936 0.0000000357466 0.00000254496 0.000026032 0.00000340511 0.0000000244109
3 0.999976 0.000000000000767564 0.00000496665 0.000000000296386 0.000000000398382 0.0000000096914 0.0000231924 0.00000000496636 0.000000000416177 0.00000000590102
4 0.00000000101704 0.0000000348997 0.0000000035179 0.000000000370033 0.999948 0.000000000158648 0.0000000238752 0.00000124147 0.00000000535041 0.000050
4937
5 0.00000000360296 0.999957 0.0000000877378 0.00000000724612 0.00000207376 0.000000000246704 0.0000000019372 0.0000401187 0.000000276865 0.00000000380118
6 0.0000000000000340775 0.0000000802971 0.0000000188454 0.0000000000318247 0.999982 0.000000000072675 0.000000000132338 0.00000376211 0.0000105578 0.00000314445
7 0.0000000000403928 0.00000053162 0.0000072458 0.0000068992 0.00813112 0.000000610434 0.0000000123234 0.0000213121 0.0000792434 0.998691
8 0.00000000000410341 0.00000000000359552 0.0000000464476 0.00000000013431 0.0000000445756 0.994047 0.000648605 0.00000000652861 0.00530346 0.000000645517
9 0.0000000000000299147 0.000000000000202292 0.000000000230089 0.00000000734767 0.00000434137 0.00000000192511 1.22969e-17 0.0000158272 0.0000000823186 0.99998
10 1 0.000000000000119218 0.000000400777 0.0000000000000722985 4.78715e-16 0.000000000000255285 0.00000000454809 0.00000000000109392 0.00000000000822686 0.000000
00287147
11 0.0000000449306 0.000000000000002127 0.000000000685528 0.000000000000116602 0.00000000221475 0.000000000233156 0.999999 5.17569e-16 0.00000107018 0.000000000000004
```

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)

Q: The following [video](#) describes how to use BigQuery and Google PubSub as sources and destinations for the Dataflow pipeline

6 Q and A

Q: 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: 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, chose **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

7 Q and A

Q: 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 decided to use another dataset, It should maintain both variety and huge volume. Your report should include but not limited to

- The application.
- Its impact.
- The used dataset (size, schema/structure).
- A graph showing the proposed pipeline(s).
- List of other tools (AI, clustering,...) needed to implement that application.

A: ---