SOFE 4630U: Cloud Computing

Date: February 16, 2023

Project Milestone 2: Data Processing Service (Dataflow)

Group T4
Name: Preet Patel
Student Id: 100708239

Links:

Github:
https://github.com/preetpatel87/Cloud-Computing-Project-Group-T4/tree/main/Project%20Milestone%20Dataflow/Preet%20Patel

Demonstration Links:

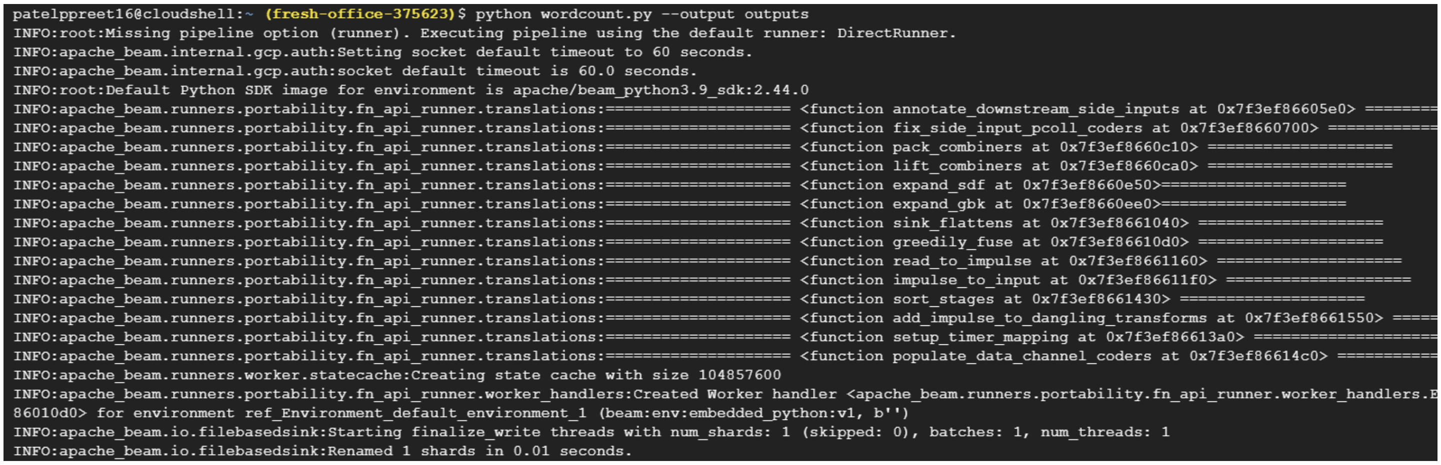
- **Wordcount and MNIST Dataflow Jobs Demo:** https://drive.google.com/file/d/1ZXILVFSoQoQrARABXP61vRDdFHcMlnG1/view?usp=sharing
- **Design Task - SmartMeter Dataflow Job Demo:** https://drive.google.com/file/d/1CnePrT04BtB4e2lXuJROsx1u9Q6rG0TW/view?usp=sharing

Discussion:

A report that includes the description of the second wordcount example (wordcount2.py) and the pipeline you used in the Design section. It should have snapshots of the job and results of the four examples (wordcount and MNIST) as well as the design part.

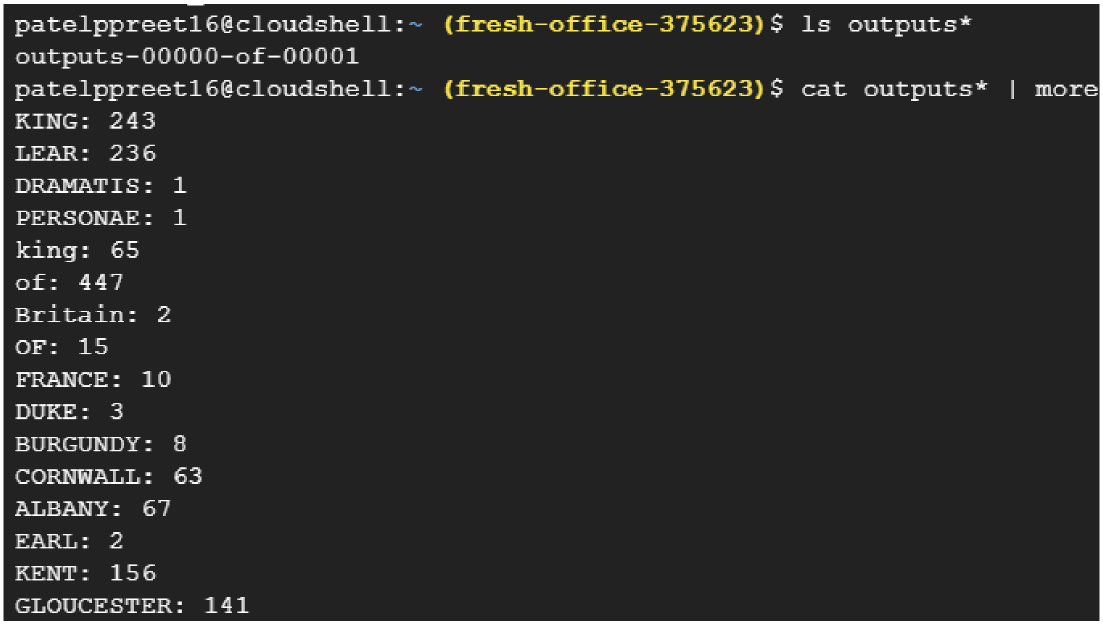
1. wordcount.py

Executed wordcount.py and storing the output into outputs



```
patelppreet16@cloudshell:~ (fresh-office-375623)$ python wordcount.py --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:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.44.0
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function annotate_downstream_side_inputs at 0x7f3ef86605e0> ========
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function fix_side_input_pcoll_coders at 0x7f3ef8660700> ===========
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function pack_combiners at 0x7f3ef8660c10> ====================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function lift_combiners at 0x7f3ef8660ca0> ====================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function expand_sdf at 0x7f3ef8660e50>====================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function expand_gbk at 0x7f3ef8660ee0>====================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function sink_flattens at 0x7f3ef8661040> ====================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function greedily_fuse at 0x7f3ef86610d0> ====================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function read_to_impulse at 0x7f3ef8661160> ====================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function impulse_to_input at 0x7f3ef86611f0> ====================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function sort_stages at 0x7f3ef8661430> ====================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function add_impulse_to_dangling_transforms at 0x7f3ef8661550> =====
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function setup_timer_mapping at 0x7f3ef86613a0> ===================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function populate_data_channel_coders at 0x7f3ef86614c0> ==========
INFO:apache_beam.runners.worker.statecache:Creating state cache with size 104857600
INFO:apache_beam.runners.portability.fn_api_runner.worker_handlers:Created Worker handler <apache_beam.runners.portability.fn_api_runner.worker_handlers.E
86010d0> for environment ref_Environment_default_environment_1 (beam:env:embedded_python:v1, b'')
INFO:apache_beam.io.filebasedsink:Starting finalize_write threads with num_shards: 1 (skipped: 0), batches: 1, num_threads: 1
INFO:apache_beam.io.filebasedsink:Renamed 1 shards in 0.01 seconds.
```

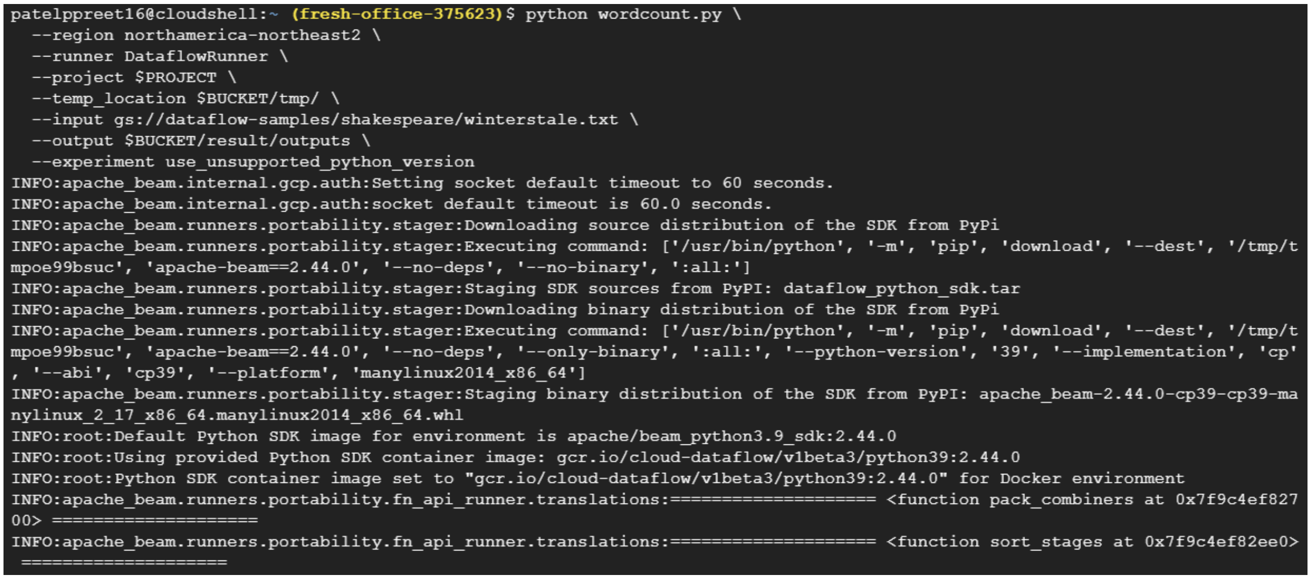
Filtered for output files and observed the output of wordcount.py by printing content of the file in the terminal.



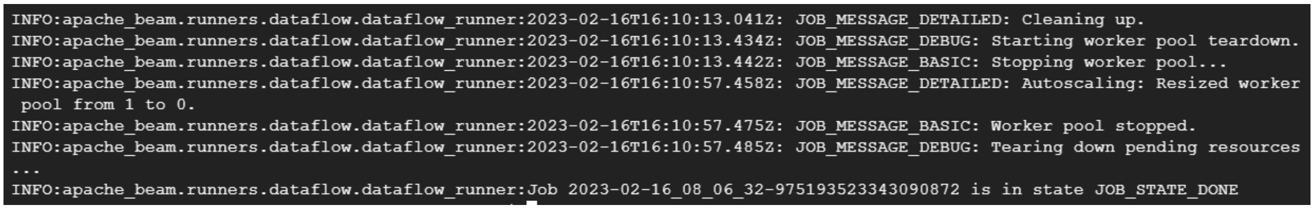
```
patelppreet16@cloudshell:~ (fresh-office-375623)$ ls outputs*
outputs-00000-of-00001
patelppreet16@cloudshell:~ (fresh-office-375623)$ cat outputs* | more
KING: 243
LEAR: 236
DRAMATIS: 1
PERSONAE: 1
king: 65
of: 447
Britain: 2
OF: 15
FRANCE: 10
DUKE: 3
BURGUNDY: 8
CORNWALL: 63
ALBANY: 67
EARL: 2
KENT: 156
GLOUCESTER: 141
```

2. Dataflow Job - wordcount.py

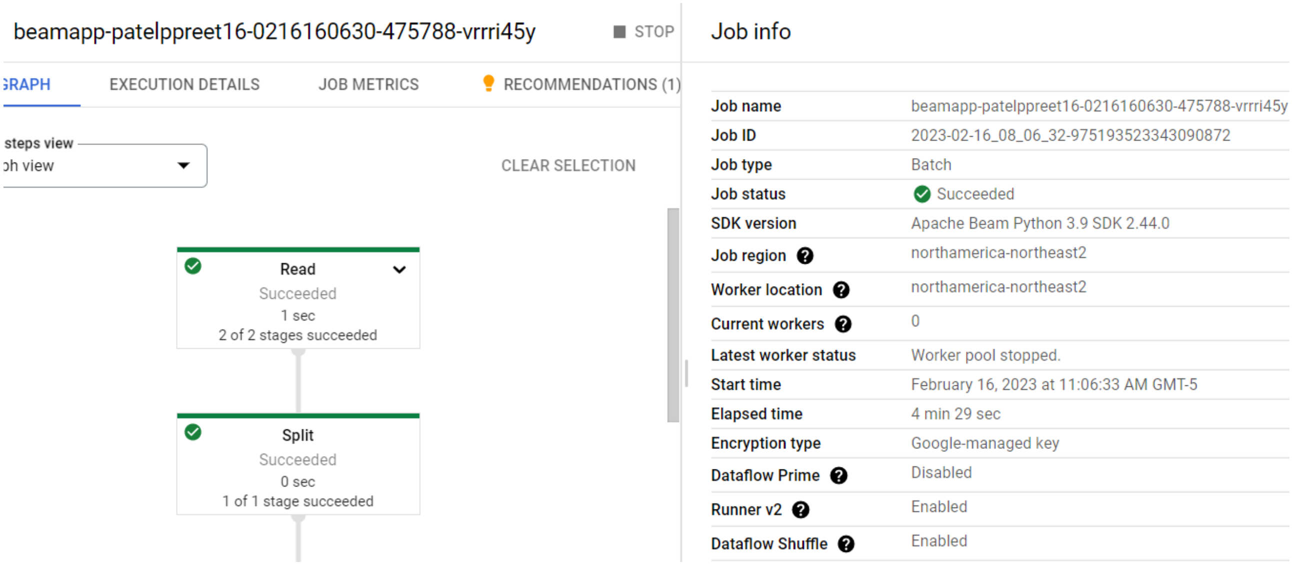
Created a dataflow job using wordcount.py by executing the following command. Provided the input as winterstale.txt file and the output was set to the result/outputs folder in the created project bucket.



```
patelppreet16@cloudshell:~ (fresh-office-375623)$ python wordcount.py \
  --region northamerica-northeast2 \
  --runner DataflowRunner \
  --project $PROJECT \
  --temp_location $BUCKET/tmp/ \
  --input gs://dataflow-samples/shakespeare/winterstale.txt \
  --output $BUCKET/result/outputs \
  --experiment use_unsupported_python_version
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:apache_beam.runners.portability.stager:Downloading source distribution of the SDK from PyPi
INFO:apache_beam.runners.portability.stager:Executing command: ['/usr/bin/python', '-m', 'pip', 'download', '--dest', '/tmp/t
mpoe99bsuc', 'apache-beam==2.44.0', '--no-deps', '--no-binary', ':all:']
INFO:apache_beam.runners.portability.stager:Staging SDK sources from PyPI: dataflow_python_sdk.tar
INFO:apache_beam.runners.portability.stager:Downloading binary distribution of the SDK from PyPi
INFO:apache_beam.runners.portability.stager:Executing command: ['/usr/bin/python', '-m', 'pip', 'download', '--dest', '/tmp/t
mpoe99bsuc', 'apache-beam==2.44.0', '--no-deps', '--only-binary', ':all:', '--python-version', '39', '--implementation', 'cp'
, '--abi', 'cp39', '--platform', 'manylinux2014_x86_64']
INFO:apache_beam.runners.portability.stager:Staging binary distribution of the SDK from PyPI: apache_beam-2.44.0-cp39-cp39-ma
nylinux_2_17_x86_64.manylinux2014_x86_64.whl
INFO:root:Default Python SDK image for environment is apache/beam_python3.9_sdk:2.44.0
INFO:root:Using provided Python SDK container image: gcr.io/cloud-dataflow/v1beta3/python39:2.44.0
INFO:root:Python SDK container image set to "gcr.io/cloud-dataflow/v1beta3/python39:2.44.0" for Docker environment
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function pack_combiners at 0x7f9c4ef827
00> ====================
INFO:apache_beam.runners.portability.fn_api_runner.translations:==================== <function sort_stages at 0x7f9c4ef82ee0>
 ====================
```



```
INFO:apache_beam.runners.dataflow.dataflow_runner:2023-02-16T16:10:13.041Z: JOB_MESSAGE_DETAILED: Cleaning up.
INFO:apache_beam.runners.dataflow.dataflow_runner:2023-02-16T16:10:13.434Z: JOB_MESSAGE_DEBUG: Starting worker pool teardown.
INFO:apache_beam.runners.dataflow.dataflow_runner:2023-02-16T16:10:13.442Z: JOB_MESSAGE_BASIC: Stopping worker pool...
INFO:apache_beam.runners.dataflow.dataflow_runner:2023-02-16T16:10:57.458Z: JOB_MESSAGE_DETAILED: Autoscaling: Resized worker
 pool from 1 to 0.
INFO:apache_beam.runners.dataflow.dataflow_runner:2023-02-16T16:10:57.475Z: JOB_MESSAGE_BASIC: Worker pool stopped.
INFO:apache_beam.runners.dataflow.dataflow_runner:2023-02-16T16:10:57.485Z: JOB_MESSAGE_DEBUG: Tearing down pending resources
...
INFO:apache_beam.runners.dataflow.dataflow_runner:Job 2023-02-16_08_06_32-975193523343090872 is in state JOB_STATE_DONE
```



beamapp-patelppreet16-0216160630-475788-vrrri45y

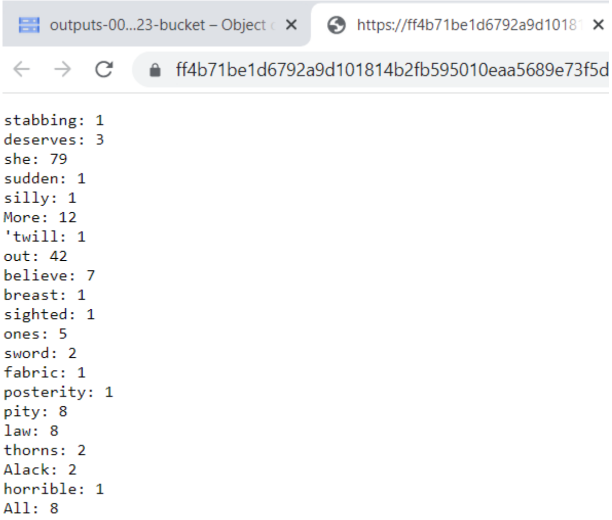
Read
Succeeded
1 sec
2 of 2 stages succeeded

Split
Succeeded
0 sec
1 of 1 stage succeeded

Job info

Job name	beamapp-patelppreet16-0216160630-475788-vrrri45y
Job ID	2023-02-16_08_06_32-975193523343090872
Job type	Batch
Job status	✓ Succeeded
SDK version	Apache Beam Python 3.9 SDK 2.44.0
Job region ?	northamerica-northeast2
Worker location ?	northamerica-northeast2
Current workers ?	0
Latest worker status	Worker pool stopped.
Start time	February 16, 2023 at 11:06:33 AM GMT-5
Elapsed time	4 min 29 sec
Encryption type	Google-managed key
Dataflow Prime ?	Disabled
Runner v2 ?	Enabled
Dataflow Shuffle ?	Enabled

Downloaded the output files from the created project bucket and opened it to view the output of the job. The word count for the entire winterstale.txt file was in the output file.

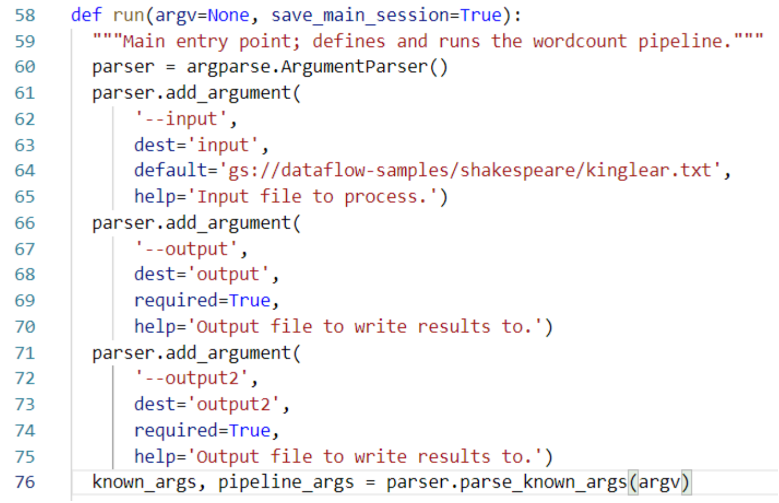

outputs-00...23-bucket – Object
https://ff4b71be1d6792a9d1018
ff4b71be1d6792a9d101814b2fb595010eaa5689e73f5d

```
stabbing: 1
deserves: 3
she: 79
sudden: 1
silly: 1
More: 12
'twill: 1
out: 42
believe: 7
breast: 1
sighted: 1
ones: 5
sword: 2
fabric: 1
posterity: 1
pity: 8
law: 8
thorns: 2
Alack: 2
horrible: 1
All: 8
```

3. Dataflow Job - wordcount2.py

wordcount2.py file description:

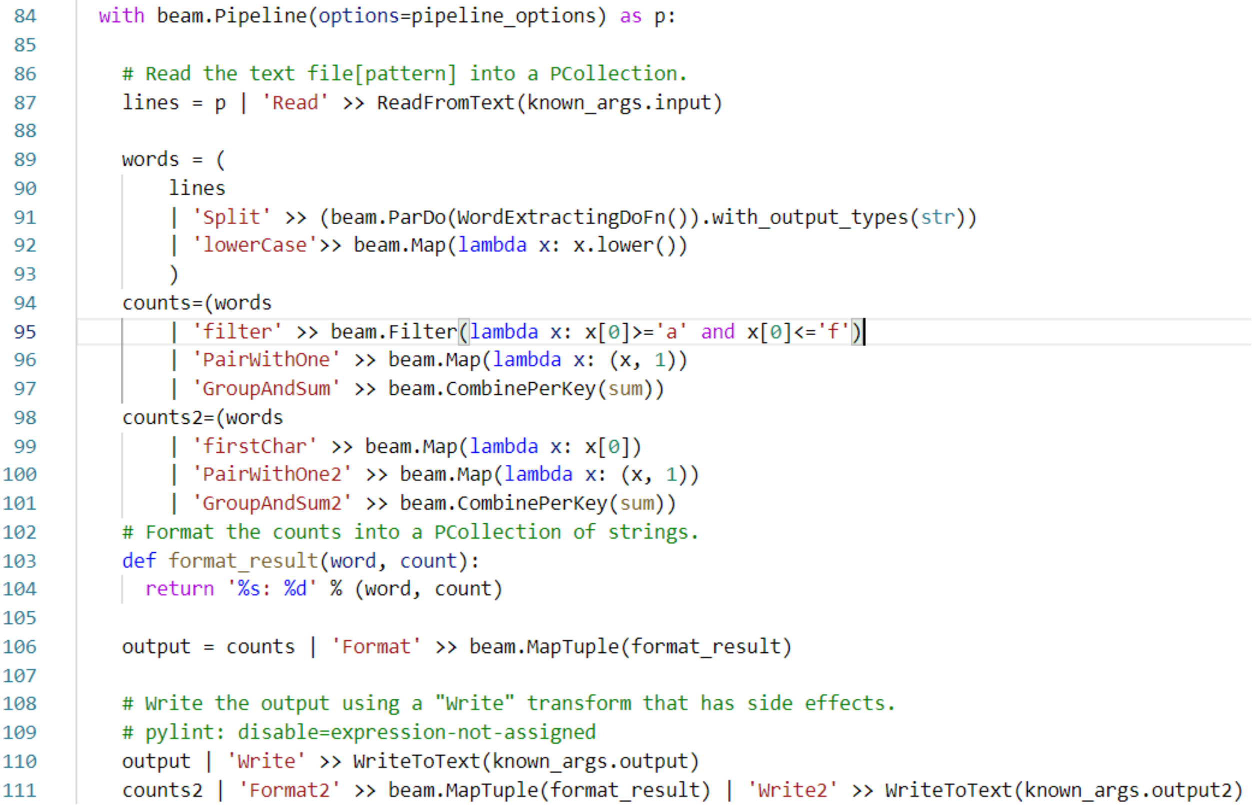
The wordcount2.py script is an upgraded version of the wordcount.py file that was executed in the previous two steps. The wordcount2.py script contains both a word count (similar to the wordcount.py script) and also a character count as well. The following image describes the different parameters required by the script. It requires an input file provided in the --input parameter (if the input file is not provided it defaults the input to kinglear.txt file). It also requires two output locations. First output will be used for word count whereas the second will be used for character count.



```python
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',
      default='gs://dataflow-samples/shakespeare/kinglear.txt',
      help='Input file to process.')
  parser.add_argument(
      '--output',
      dest='output',
      required=True,
      help='Output file to write results to.')
  parser.add_argument(
      '--output2',
      dest='output2',
      required=True,
      help='Output file to write results to.')
  known_args, pipeline_args = parser.parse_known_args(argv)
```

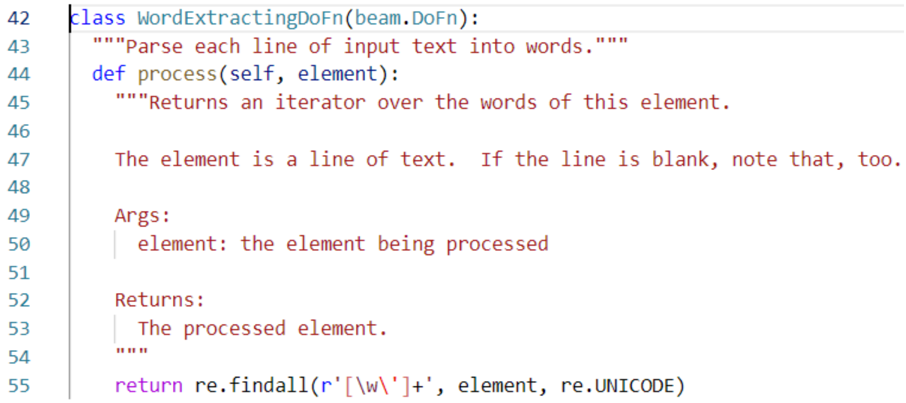
The next image below shows all the different stages of the pipeline for the Dataflow job. The first stage is ‘Read’ which involves reading the input text file. The second stage ‘Split’ stage involves parsing the input file and converting each line of the input into words which is performed by the WordExtractingDoFn function (provided in the images below). The third stage ‘lowerCase’ will convert all the words to lowercase and map them to a dictionary. The result of these stages are stored to variable words which is then the input for the next two split parts of the pipeline. After the lowerCase stage the pipeline will split into 2 paths. The first path result which is stored in counts variable, starts with the ‘Filter’ stage that filters all the words that begin with any letters from a to f in the alphabet. All other words that start with letters outside of that range will be filtered out. Next, in the ‘PairWithOne’ the words will be used to create a key/value pair with the key being the word and the value being its count (currently initialized to 1). Then, the ‘GroupAndSum’ stage will count the number of times the key (word) occurs by combining the map.

The second path also functions similarly to the word count path. Firstly, in the ‘firstChar’ stage the first letter of each word is extracted and a map (dictionary) is created of those letters. Next, similar to the first path the ‘PairWithOne2’ and ‘GroupAndSum2’ stages will create a key/value pair for each entry(letter) in the map and then combine the map to count the number of times each key (letter) occurs. Finally, the outputs for both the paths will be appropriately formatted in the ‘Format’ and ‘Format2’ stages respectively and written to output files in the location defined in the parameter of the command to execute this job.



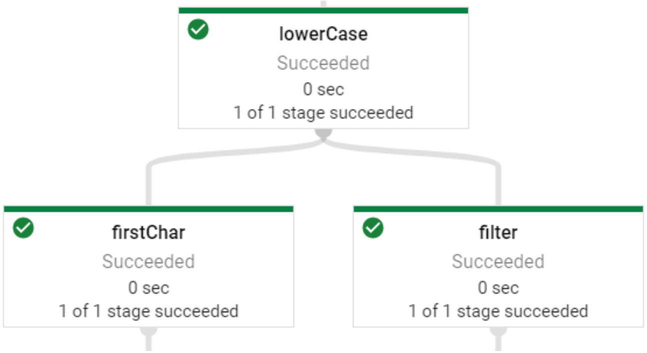
```
with beam.Pipeline(options=pipeline_options) as p:

  # Read the text file[pattern] into a PCollection.
  lines = p | 'Read' >> ReadFromText(known_args.input)

  words = (
      lines
      | 'Split' >> (beam.ParDo(WordExtractingDoFn()).with_output_types(str))
      | 'lowerCase'>> beam.Map(lambda x: x.lower())
      )
  counts=(words
      | 'filter' >> beam.Filter(lambda x: x[0]>='a' and x[0]<='f')
      | 'PairWithOne' >> beam.Map(lambda x: (x, 1))
      | 'GroupAndSum' >> beam.CombinePerKey(sum))
  counts2=(words
      | 'firstChar' >> beam.Map(lambda x: x[0])
      | 'PairWithOne2' >> beam.Map(lambda x: (x, 1))
      | 'GroupAndSum2' >> beam.CombinePerKey(sum))
  # Format the counts into a PCollection of strings.
  def format_result(word, count):
    return '%s: %d' % (word, count)

  output = counts | 'Format' >> beam.MapTuple(format_result)

  # Write the output using a "Write" transform that has side effects.
  # pylint: disable=expression-not-assigned
  output | 'Write' >> WriteToText(known_args.output)
  counts2 | 'Format2' >> beam.MapTuple(format_result) | 'Write2' >> WriteToText(known_args.output2)
```

WordExtractingDoFn Function:



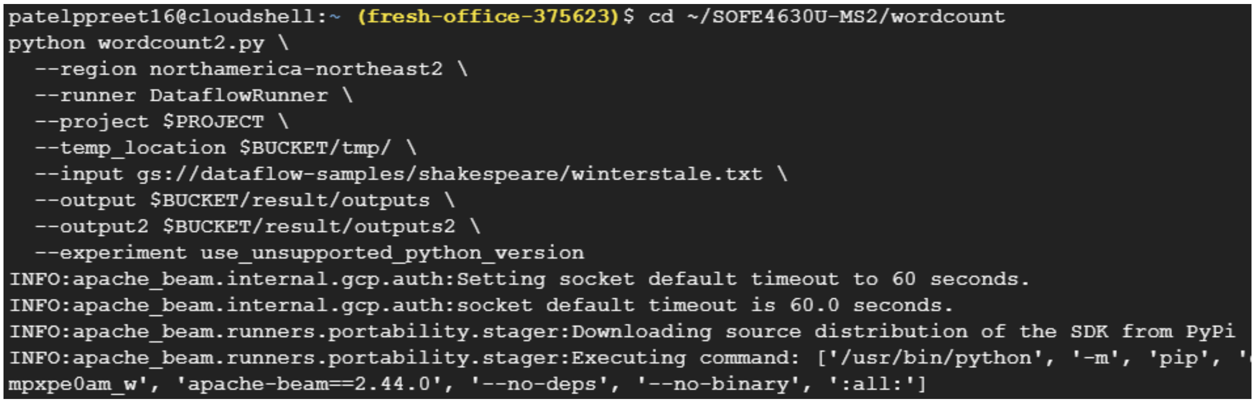
```
class WordExtractingDoFn(beam.DoFn):
  """Parse each line of input text into words."""
  def process(self, element):
    """Returns an iterator over the words of this element.

    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)
```

Job Pipeline split after lowerCase stage:

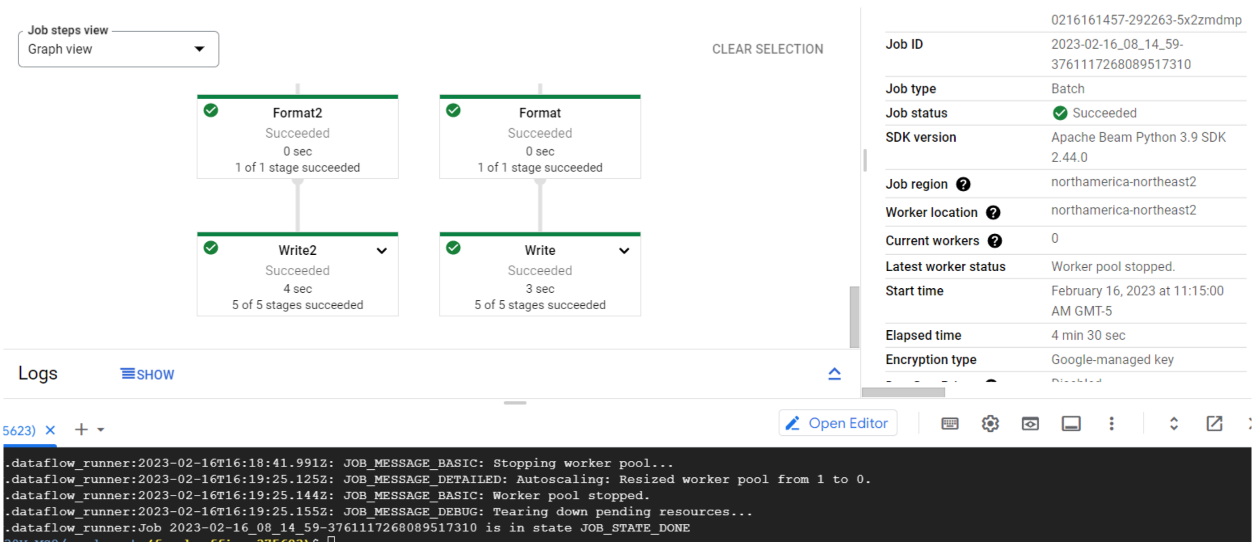


Task:

Created a dataflow job using the provided wordcount2.py by executing the following command. Provided the input as winterstale.txt file and the output was set to the result/outputs folder whereas the output2 was set to the results/outputs2 folder in the created project bucket.



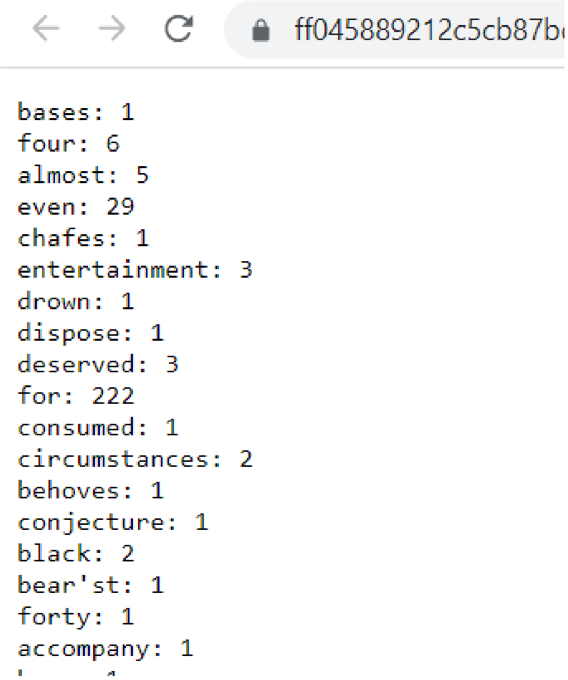
```
patelppreet16@cloudshell:~ (fresh-office-375623)$ cd ~/SOFE4630U-MS2/wordcount
python wordcount2.py \
  --region northamerica-northeast2 \
  --runner DataflowRunner \
  --project $PROJECT \
  --temp_location $BUCKET/tmp/ \
  --input gs://dataflow-samples/shakespeare/winterstale.txt \
  --output $BUCKET/result/outputs \
  --output2 $BUCKET/result/outputs2 \
  --experiment use_unsupported_python_version
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:apache_beam.runners.portability.stager:Downloading source distribution of the SDK from PyPi
INFO:apache_beam.runners.portability.stager:Executing command: ['/usr/bin/python', '-m', 'pip', '
mpxpe0am_w', 'apache-beam==2.44.0', '--no-deps', '--no-binary', ':all:']
```



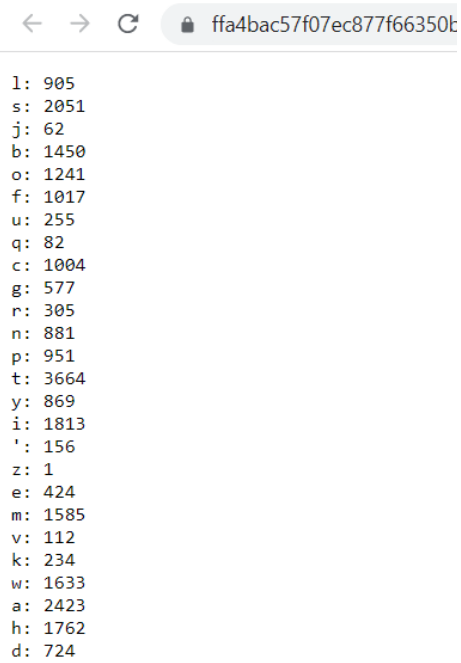
	0216161457-292263-5x2zmdmp
Job ID	2023-02-16_08_14_59-3761117268089517310
Job type	Batch
Job status	Succeeded
SDK version	Apache Beam Python 3.9 SDK 2.44.0
Job region	northamerica-northeast2
Worker location	northamerica-northeast2
Current workers	0
Latest worker status	Worker pool stopped.
Start time	February 16, 2023 at 11:15:00 AM GMT-5
Elapsed time	4 min 30 sec
Encryption type	Google-managed key

```
.dataflow_runner:2023-02-16T16:18:41.991Z: JOB_MESSAGE_BASIC: Stopping worker pool...
.dataflow_runner:2023-02-16T16:19:25.125Z: JOB_MESSAGE_DETAILED: Autoscaling: Resized worker pool from 1 to 0.
.dataflow_runner:2023-02-16T16:19:25.144Z: JOB_MESSAGE_BASIC: Worker pool stopped.
.dataflow_runner:2023-02-16T16:19:25.155Z: JOB_MESSAGE_DEBUG: Tearing down pending resources...
.dataflow_runner:Job 2023-02-16_08_14_59-3761117268089517310 is in state JOB_STATE_DONE
```

Downloaded the two output files from the created project bucket and opened both of them to view the output of the job. The word count for the entire winterstale.txt file was in the first output file. The letter count was in the second output file.



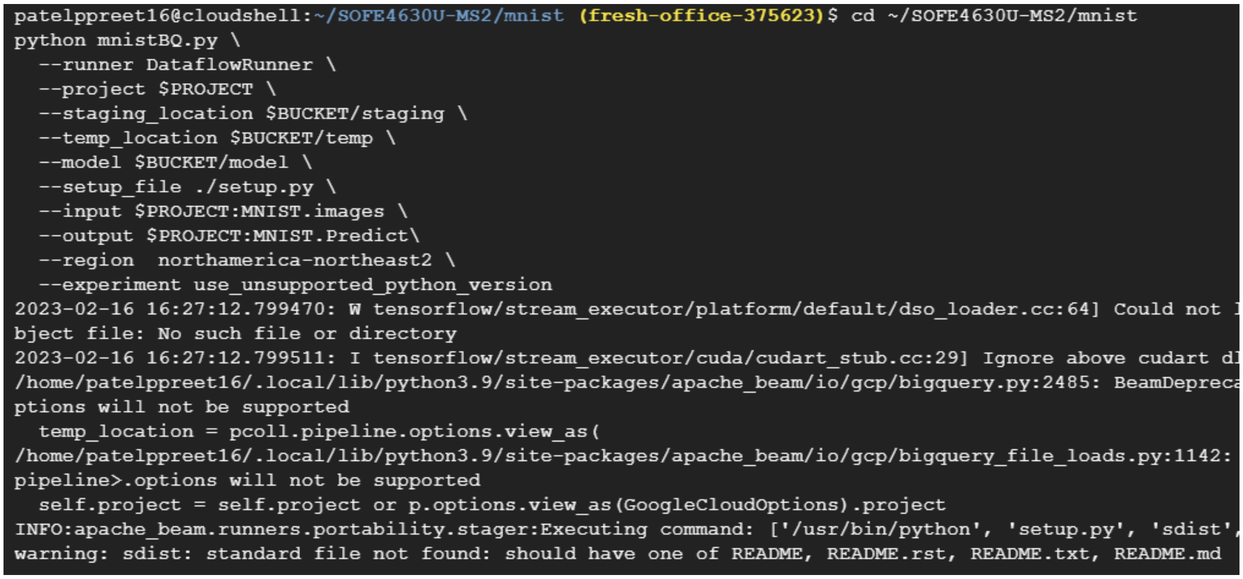
```
bases: 1
four: 6
almost: 5
even: 29
chafes: 1
entertainment: 3
drown: 1
dispose: 1
deserved: 3
for: 222
consumed: 1
circumstances: 2
behoves: 1
conjecture: 1
black: 2
bear'st: 1
forty: 1
accompany: 1
```



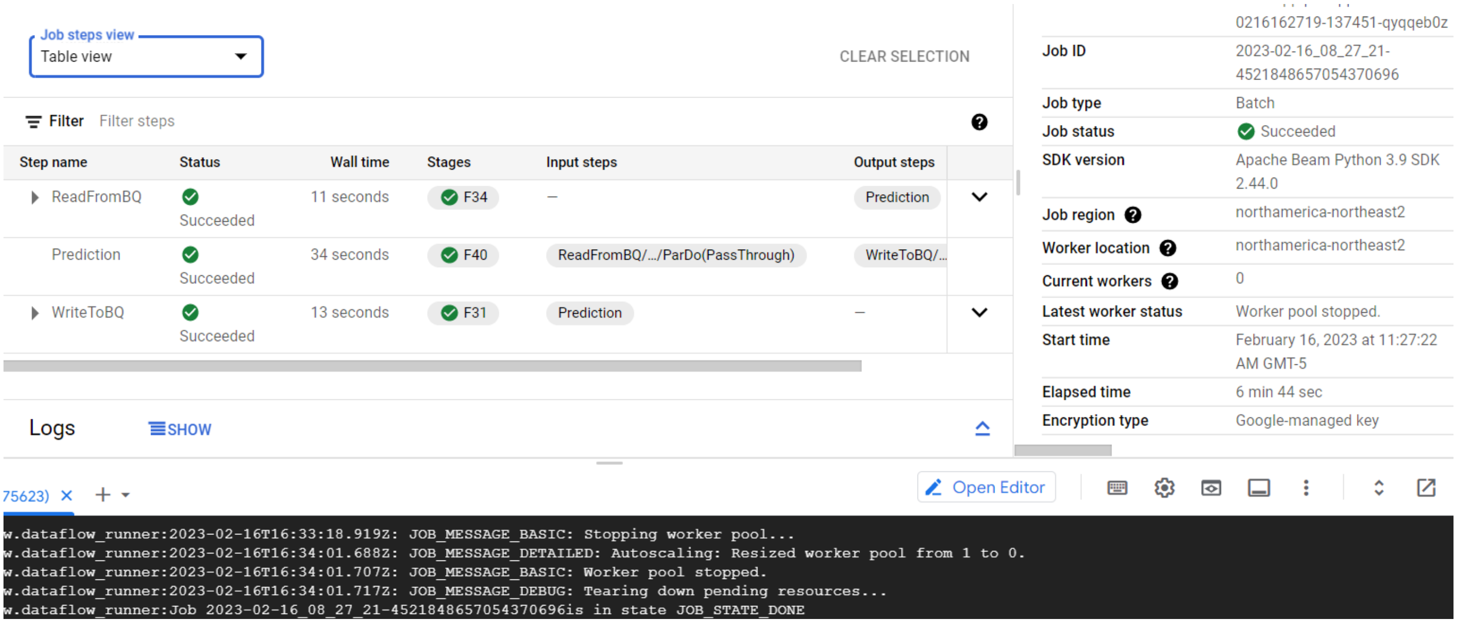
```
l: 905
s: 2051
j: 62
b: 1450
o: 1241
f: 1017
u: 255
q: 82
c: 1004
g: 577
r: 305
n: 881
p: 951
t: 3664
y: 869
i: 1813
': 156
z: 1
e: 424
m: 1585
v: 112
k: 234
w: 1633
a: 2423
h: 1762
d: 724
```

4. Dataflow Job with Big Query - MNIST

Created a dataflow job using the provided mnistBQ.py script by executing the following command. Provided the required parameters such as the staging location, location of the model to detect the digits from images, the setup file which downloads tensorflow, the input data which was extracted from a csv file and stored into a BigQuery table named images and the output was set to a BigQuery table named Predict which will be created by the job.



```
patelppreet16@cloudshell:~/SOFE4630U-MS2/mnist (fresh-office-375623)$ cd ~/SOFE4630U-MS2/mnist
python mnistBQ.py \
  --runner DataflowRunner \
  --project $PROJECT \
  --staging_location $BUCKET/staging \
  --temp_location $BUCKET/temp \
  --model $BUCKET/model \
  --setup_file ./setup.py \
  --input $PROJECT:MNIST.images \
  --output $PROJECT:MNIST.Predict\
  --region  northamerica-northeast2 \
  --experiment use_unsupported_python_version
2023-02-16 16:27:12.799470: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not l
bject file: No such file or directory
2023-02-16 16:27:12.799511: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dl
/home/patelppreet16/.local/lib/python3.9/site-packages/apache_beam/io/gcp/bigquery.py:2485: BeamDepreca
ptions will not be supported
  temp_location = pcoll.pipeline.options.view_as(
/home/patelppreet16/.local/lib/python3.9/site-packages/apache_beam/io/gcp/bigquery_file_loads.py:1142:
pipeline>.options will not be supported
  self.project = self.project or p.options.view_as(GoogleCloudOptions).project
INFO:apache_beam.runners.portability.stager:Executing command: ['/usr/bin/python', 'setup.py', 'sdist',
warning: sdist: standard file not found: should have one of README, README.rst, README.txt, README.md
```

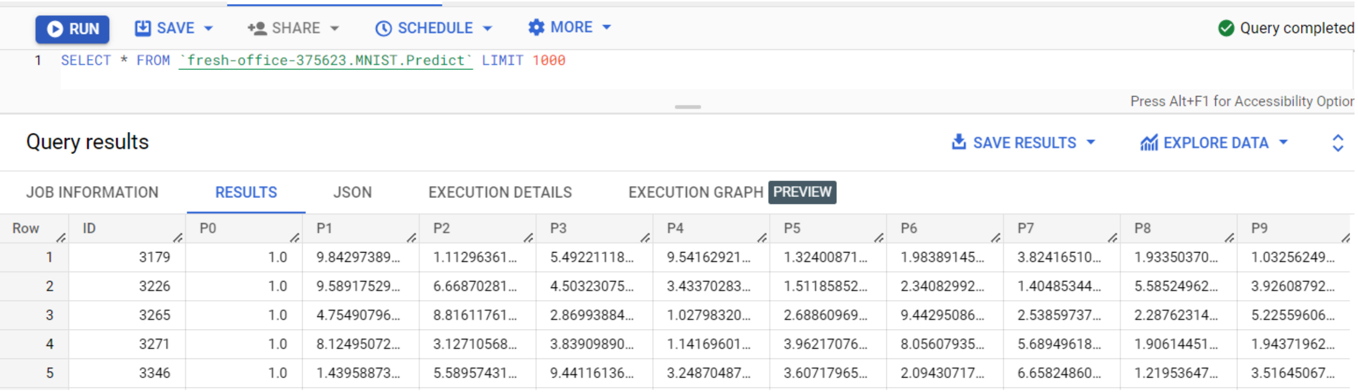


Step name	Status	Wall time	Stages	Input steps	Output steps
ReadFromBQ	✓ Succeeded	11 seconds	✓ F34	–	Prediction
Prediction	✓ Succeeded	34 seconds	✓ F40	ReadFromBQ/.../ParDo(PassThrough)	WriteToBQ/...
WriteToBQ	✓ Succeeded	13 seconds	✓ F31	Prediction	–

	0216162719-137451-qyqqeb0z
Job ID	2023-02-16_08_27_21-4521848657054370696
Job type	Batch
Job status	✓ Succeeded
SDK version	Apache Beam Python 3.9 SDK 2.44.0
Job region	northamerica-northeast2
Worker location	northamerica-northeast2
Current workers	0
Latest worker status	Worker pool stopped.
Start time	February 16, 2023 at 11:27:22 AM GMT-5
Elapsed time	6 min 44 sec
Encryption type	Google-managed key

```
w.dataflow_runner:2023-02-16T16:33:18.919Z: JOB_MESSAGE_BASIC: Stopping worker pool...
w.dataflow_runner:2023-02-16T16:34:01.688Z: JOB_MESSAGE_DETAILED: Autoscaling: Resized worker pool from 1 to 0.
w.dataflow_runner:2023-02-16T16:34:01.707Z: JOB_MESSAGE_BASIC: Worker pool stopped.
w.dataflow_runner:2023-02-16T16:34:01.717Z: JOB_MESSAGE_DEBUG: Tearing down pending resources...
w.dataflow_runner:Job 2023-02-16_08_27_21-4521848657054370696is in state JOB_STATE_DONE
```

The BigQuery table named Predict in the database which is created by the job was queried to view the output of the job (model) which are the predictions for what the digit in the image is.

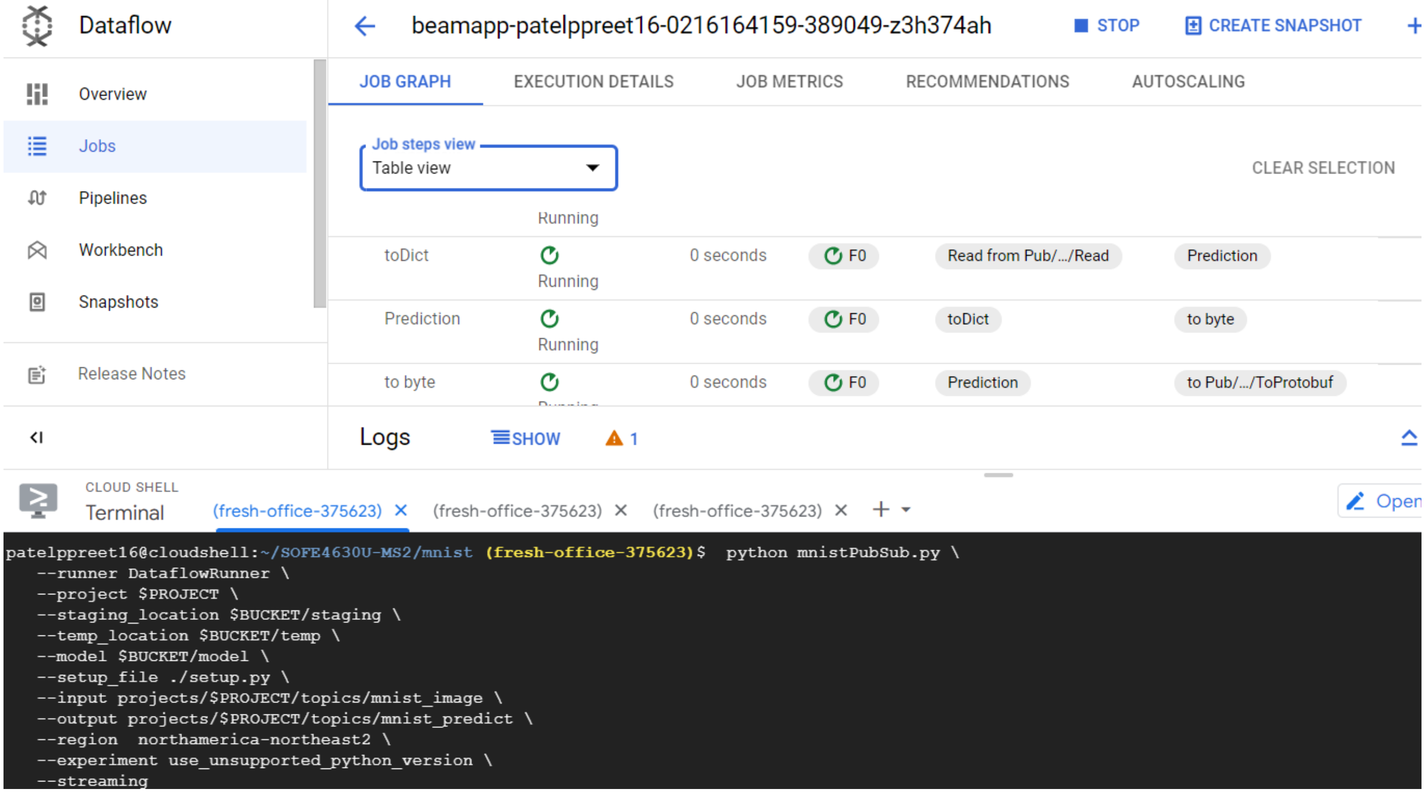


```
1  SELECT * FROM `fresh-office-375623.MNIST.Predict` LIMIT 1000
```

Row	ID	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9
1	3179	1.0	9.84297389...	1.11296361...	5.49221118...	9.54162921...	1.32400871...	1.98389145...	3.82416510...	1.93350370...	1.03256249...
2	3226	1.0	9.58917529...	6.66870281...	4.50323075...	3.43370283...	1.51185852...	2.34082992...	1.40485344...	5.58524962...	3.92608792...
3	3265	1.0	4.75490796...	8.81611761...	2.86993884...	1.02798320...	2.68860969...	9.44295086...	2.53859737...	2.28762314...	5.22559606...
4	3271	1.0	8.12495072...	3.12710568...	3.83909890...	1.14169601...	3.96217076...	8.05607935...	5.68949618...	1.90614451...	1.94371962...
5	3346	1.0	1.43958873...	5.58957431...	9.44116136...	3.24870487...	3.60717965...	2.09430717...	6.65824860...	1.21953647...	3.51645067...

5. Dataflow Job using Pub/Sub - MNIST

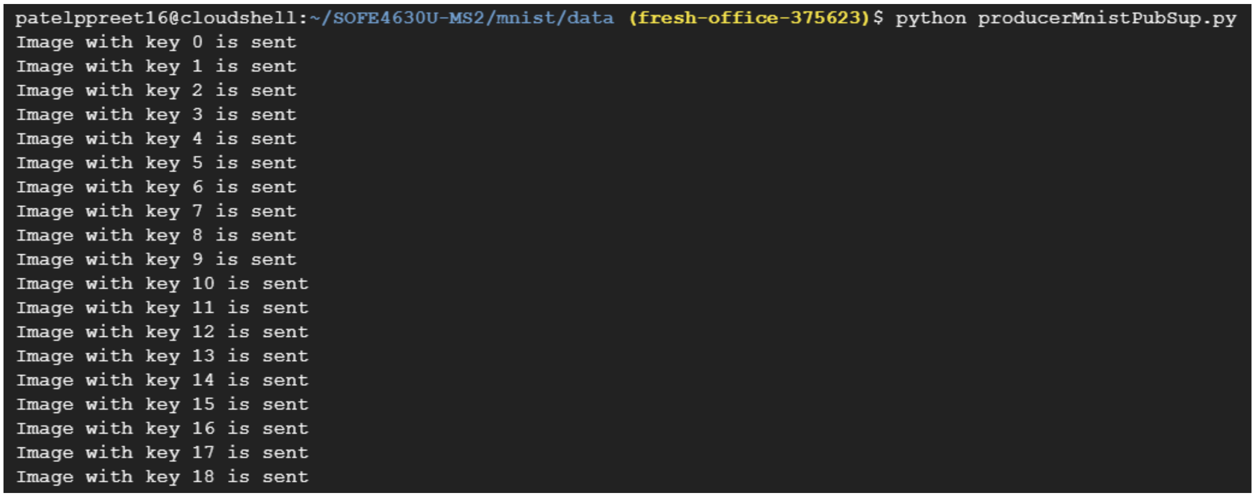
Created a dataflow job using the provided mnistPubSub.py script by executing the following command. Provided the required parameters such as the staging location, location of the model that detects the digits from images, the setup file which downloads tensorflow, the input which is a topic that the job will subscribe to/consume that will provided the input data in messages sent to the topic and the output which is also a topic that the job will publish the output data from the job to.



	Running				
toDict	Running	0 seconds	F0	Read from Pub/.../Read	Prediction
Prediction	Running	0 seconds	F0	toDict	to byte
to byte		0 seconds	F0	Prediction	to Pub/.../ToProtobuf

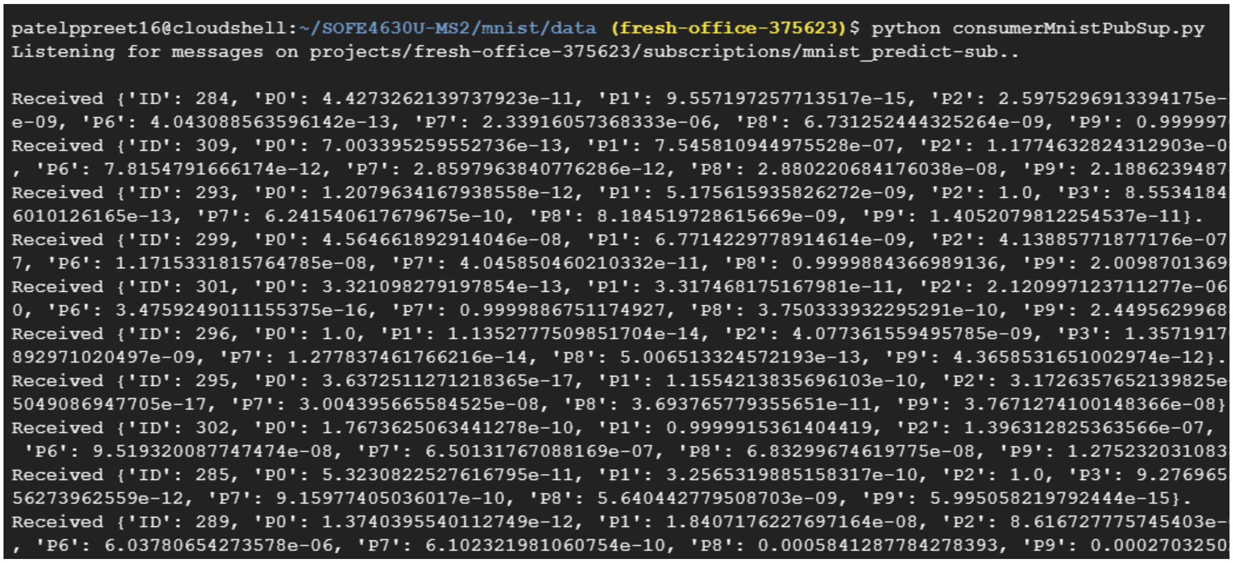
```
patelppreet16@cloudshell:~/SOFE4630U-MS2/mnist (fresh-office-375623)$  python mnistPubSub.py \
   --runner DataflowRunner \
   --project $PROJECT \
   --staging_location $BUCKET/staging \
   --temp_location $BUCKET/temp \
   --model $BUCKET/model \
   --setup_file ./setup.py \
   --input projects/$PROJECT/topics/mnist_image \
   --output projects/$PROJECT/topics/mnist_predict \
   --region  northamerica-northeast2 \
   --experiment use_unsupported_python_version \
   --streaming
```

The following image shows the producer script that will publish input data for the job to the input topic.



```
patelppreet16@cloudshell:~/SOFE4630U-MS2/mnist/data (fresh-office-375623)$ python producerMnistPubSup.py
Image with key 0 is sent
Image with key 1 is sent
Image with key 2 is sent
Image with key 3 is sent
Image with key 4 is sent
Image with key 5 is sent
Image with key 6 is sent
Image with key 7 is sent
Image with key 8 is sent
Image with key 9 is sent
Image with key 10 is sent
Image with key 11 is sent
Image with key 12 is sent
Image with key 13 is sent
Image with key 14 is sent
Image with key 15 is sent
Image with key 16 is sent
Image with key 17 is sent
Image with key 18 is sent
```

The following image shows the consumer script that will consume the output topic to print the output of the job to the terminal.



```
patelppreet16@cloudshell:~/SOFE4630U-MS2/mnist/data (fresh-office-375623)$ python consumerMnistPubSup.py
Listening for messages on projects/fresh-office-375623/subscriptions/mnist_predict-sub..

Received {'ID': 284, 'P0': 4.4273262139737923e-11, 'P1': 9.557197257713517e-15, 'P2': 2.5975296913394175e-
e-09, 'P6': 4.043088563596142e-13, 'P7': 2.33916057368333e-06, 'P8': 6.731252444325264e-09, 'P9': 0.999997
Received {'ID': 309, 'P0': 7.003395259552736e-13, 'P1': 7.545810944975528e-07, 'P2': 1.1774632824312903e-0
, 'P6': 7.8154791666174e-12, 'P7': 2.8597963840776286e-12, 'P8': 2.880220684176038e-08, 'P9': 2.1886239487
Received {'ID': 293, 'P0': 1.2079634167938558e-12, 'P1': 5.175615935826272e-09, 'P2': 1.0, 'P3': 8.5534184
6010126165e-13, 'P7': 6.241540617679675e-10, 'P8': 8.184519728615669e-09, 'P9': 1.4052079812254537e-11}.
Received {'ID': 299, 'P0': 4.564661892914046e-08, 'P1': 6.7714229778914614e-09, 'P2': 4.13885771877176e-07
7, 'P6': 1.1715331815764785e-08, 'P7': 4.045850460210332e-11, 'P8': 0.9999884366989136, 'P9': 2.0098701369
Received {'ID': 301, 'P0': 3.321098279197854e-13, 'P1': 3.317468175167981e-11, 'P2': 2.120997123711277e-06
0, 'P6': 3.4759249011155375e-16, 'P7': 0.9999886751174927, 'P8': 3.750333932295291e-10, 'P9': 2.4495629968
Received {'ID': 296, 'P0': 1.0, 'P1': 1.1352777509851704e-14, 'P2': 4.077361559495785e-09, 'P3': 1.3571917
892971020497e-09, 'P7': 1.277837461766216e-14, 'P8': 5.006513324572193e-13, 'P9': 4.3658531651002974e-12}.
Received {'ID': 295, 'P0': 3.6372511271218365e-17, 'P1': 1.1554213835696103e-10, 'P2': 3.1726357652139825e
5049086947705e-17, 'P7': 3.004395665584525e-08, 'P8': 3.693765779355651e-11, 'P9': 3.7671274100148366e-08}
Received {'ID': 302, 'P0': 1.7673625063441278e-10, 'P1': 0.9999915361404419, 'P2': 1.396312825363566e-07,
 'P6': 9.519320087747474e-08, 'P7': 6.50131767088169e-07, 'P8': 6.83299674619775e-08, 'P9': 1.275232031083
Received {'ID': 285, 'P0': 5.3230822527616795e-11, 'P1': 3.2565319885158317e-10, 'P2': 1.0, 'P3': 9.276965
56273962559e-12, 'P7': 9.15977405036017e-10, 'P8': 5.640442779508703e-09, 'P9': 5.995058219792444e-15}.
Received {'ID': 289, 'P0': 1.3740395540112749e-12, 'P1': 1.8407176227697164e-08, 'P2': 8.616727775745403e-
, 'P6': 6.03780654273578e-06, 'P7': 6.102321981060754e-10, 'P8': 0.0005841287784278393, 'P9': 0.0002703250
```

Design:

In the previous milestone, you have sent the smart meter readings to Google Pub/Sub. It's needed to add a Dataflow job to preprocess the smart meter measurements. The job consists of the following stages
Read from PubSub: read the measurement reading .
Filter: Eliminate records with missing measurements (containing None).

Convert: convert the pressure from kPa to psi and the temperature from Celsius to Fahrenheit using the following equations

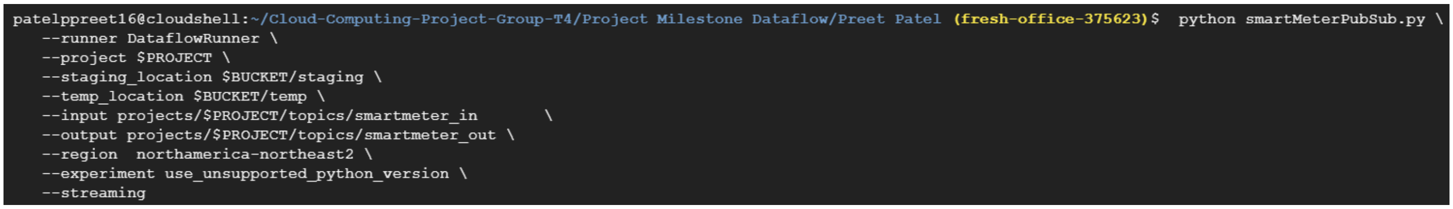
$P(psi) = P(kPa)/\mathbf{6.895}$

$T(F) = T(C)*\mathbf{1.8+32}$

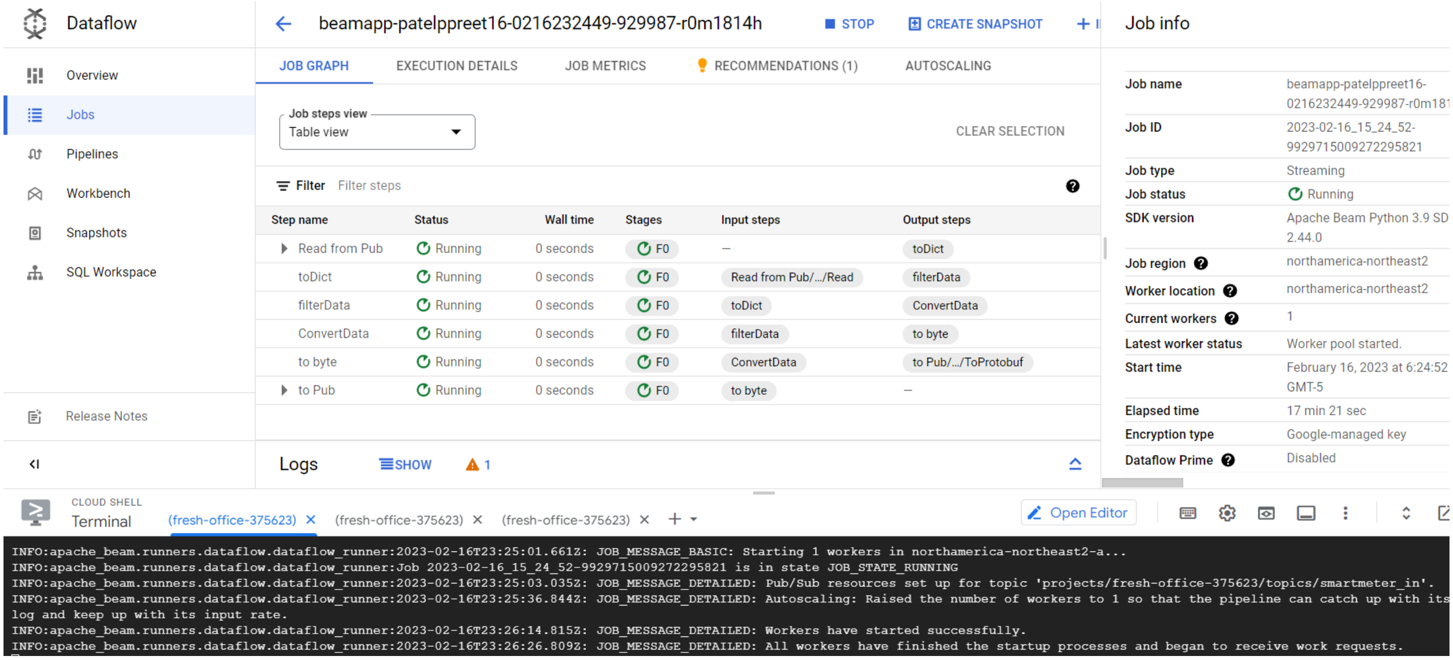
Write to PubSub: send the measurement back to another topic

- **Video Demonstration:** https://drive.google.com/file/d/1CnePrT04BtB4e2lXuJROsx1u9Q6rG0TW/view?usp=sharing
- **Python Code:** https://github.com/preetpatel87/Cloud-Computing-Project-Group-T4/tree/main/Project%20Milestone%20Dataflow/Preet%20Patel

Created a dataflow job using the created smartMeterPubSub.py script by executing the following command. The commands provide the required parameters such as the input which is a topic that the job will subscribe to/consume that will provide the input data in messages sent to the topic and the output which is also a topic that the job will publish the output data from the job to.



```
patelppreet16@cloudshell:~/Cloud-Computing-Project-Group-T4/Project Milestone Dataflow/Preet Patel (fresh-office-375623)$  python smartMeterPubSub.py \
   --runner DataflowRunner \
   --project $PROJECT \
   --staging_location $BUCKET/staging \
   --temp_location $BUCKET/temp \
   --input projects/$PROJECT/topics/smartmeter_in       \
   --output projects/$PROJECT/topics/smartmeter_out \
   --region  northamerica-northeast2 \
   --experiment use_unsupported_python_version \
   --streaming
```



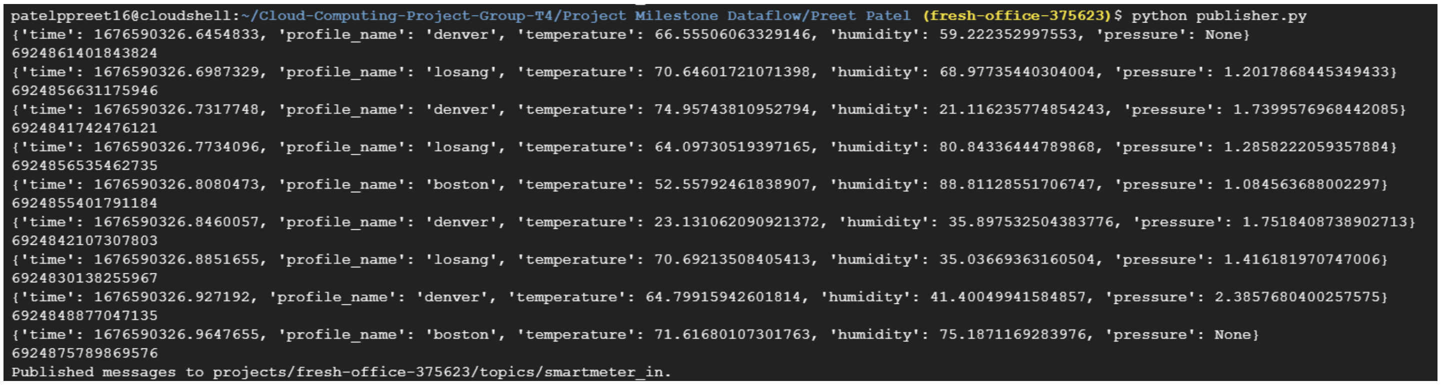
Step name	Status	Wall time	Stages	Input steps	Output steps
Read from Pub	Running	0 seconds	F0	–	toDict
toDict	Running	0 seconds	F0	Read from Pub/.../Read	filterData
filterData	Running	0 seconds	F0	toDict	ConvertData
ConvertData	Running	0 seconds	F0	filterData	to byte
to byte	Running	0 seconds	F0	ConvertData	to Pub/.../ToProtobuf
to Pub	Running	0 seconds	F0	to byte	–

Job info

Job name	beamapp-patelppreet16-0216232449-929987-r0m181
Job ID	2023-02-16_15_24_52-9929715009272295821
Job type	Streaming
Job status	Running
SDK version	Apache Beam Python 3.9 SD 2.44.0
Job region	northamerica-northeast2
Worker location	northamerica-northeast2
Current workers	1
Latest worker status	Worker pool started.
Start time	February 16, 2023 at 6:24:52 GMT-5
Elapsed time	17 min 21 sec
Encryption type	Google-managed key
Dataflow Prime	Disabled

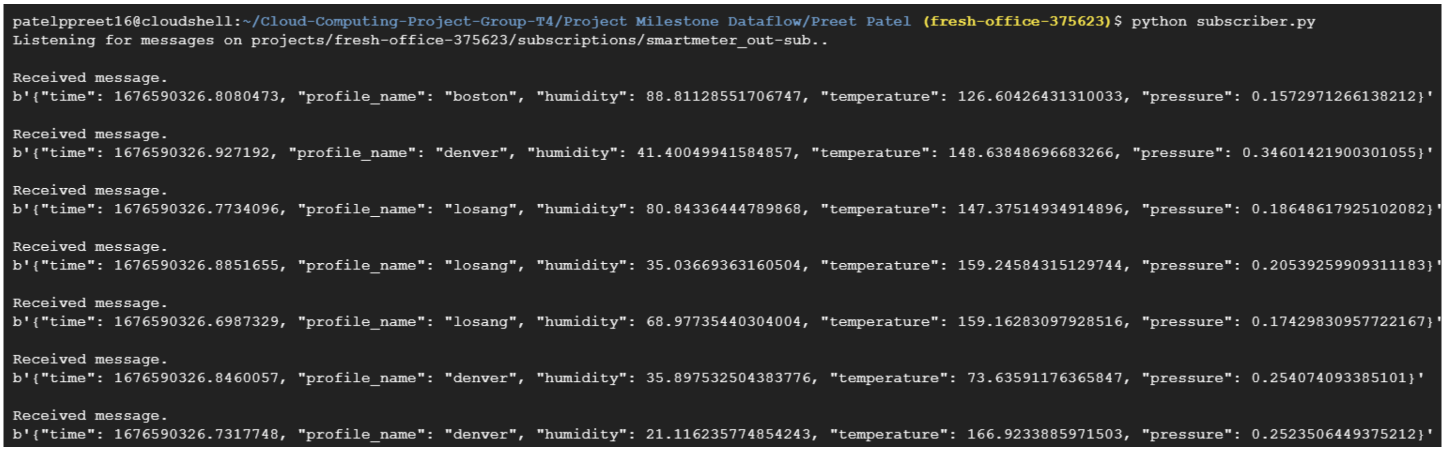
```
INFO:apache_beam.runners.dataflow.dataflow_runner:2023-02-16T23:25:01.661Z: JOB_MESSAGE_BASIC: Starting 1 workers in northamerica-northeast2-a...
INFO:apache_beam.runners.dataflow.dataflow_runner:Job 2023-02-16_15_24_52-9929715009272295821 is in state JOB_STATE_RUNNING
INFO:apache_beam.runners.dataflow.dataflow_runner:2023-02-16T23:25:03.035Z: JOB_MESSAGE_DETAILED: Pub/Sub resources set up for topic 'projects/fresh-office-375623/topics/smartmeter_in'.
INFO:apache_beam.runners.dataflow.dataflow_runner:2023-02-16T23:25:36.844Z: JOB_MESSAGE_DETAILED: Autoscaling: Raised the number of workers to 1 so that the pipeline can catch up with its
log and keep up with its input rate.
INFO:apache_beam.runners.dataflow.dataflow_runner:2023-02-16T23:26:14.815Z: JOB_MESSAGE_DETAILED: Workers have started successfully.
INFO:apache_beam.runners.dataflow.dataflow_runner:2023-02-16T23:26:26.809Z: JOB_MESSAGE_DETAILED: All workers have finished the startup processes and began to receive work requests.
```

The following image shows the publisher script that will publish generated smart meter measurements input data required for the job to the input topic. This script generates and publishes 9 different messages containing generated data.



```
patelppreet16@cloudshell:~/Cloud-Computing-Project-Group-T4/Project Milestone Dataflow/Preet Patel (fresh-office-375623)$ python publisher.py
{'time': 1676590326.6454833, 'profile_name': 'denver', 'temperature': 66.55506063329146, 'humidity': 59.222352997553, 'pressure': None}
6924861401843824
{'time': 1676590326.6987329, 'profile_name': 'losang', 'temperature': 70.64601721071398, 'humidity': 68.97735440304004, 'pressure': 1.2017868445349433}
6924856631175946
{'time': 1676590326.7317748, 'profile_name': 'denver', 'temperature': 74.95743810952794, 'humidity': 21.116235774854243, 'pressure': 1.7399576968442085}
6924841742476121
{'time': 1676590326.7734096, 'profile_name': 'losang', 'temperature': 64.09730519397165, 'humidity': 80.84336444789868, 'pressure': 1.2858222059357884}
6924856535462735
{'time': 1676590326.8080473, 'profile_name': 'boston', 'temperature': 52.55792461838907, 'humidity': 88.81128551706747, 'pressure': 1.084563688002297}
6924855401791184
{'time': 1676590326.8460057, 'profile_name': 'denver', 'temperature': 23.131062090921372, 'humidity': 35.897532504383776, 'pressure': 1.7518408738902713}
6924842107307803
{'time': 1676590326.8851655, 'profile_name': 'losang', 'temperature': 70.69213508405413, 'humidity': 35.03669363160504, 'pressure': 1.416181970747006}
6924830138255967
{'time': 1676590326.927192, 'profile_name': 'denver', 'temperature': 64.79915942601814, 'humidity': 41.40049941584857, 'pressure': 2.3857680400257575}
6924848877047135
{'time': 1676590326.9647655, 'profile_name': 'boston', 'temperature': 71.61680107301763, 'humidity': 75.1871169283976, 'pressure': None}
6924875789869576
Published messages to projects/fresh-office-375623/topics/smartmeter_in.
```

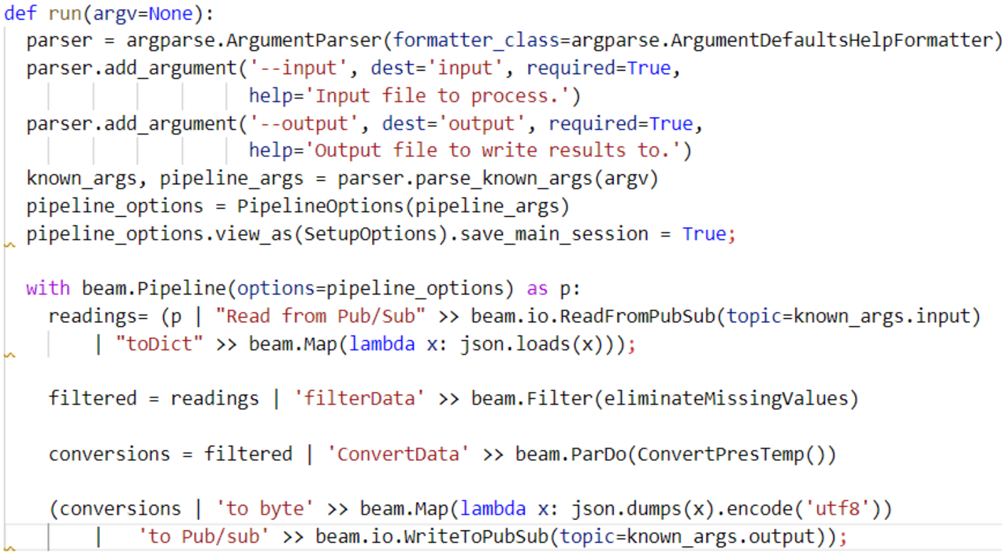
The following image shows the subscriber script that will consume the output topic to print the output of the job to the terminal. We can observe that out of the 9 inputs the input data that had missing data was filtered out and the rest of the data is published to the output topic which is consumed and printed in the terminal by the script. We also observed the change in temperature and pressure measurements which occurs due to the conversion of the temperature (to Fahrenheit) and pressure data (to psi).



```
patelppreet16@cloudshell:~/Cloud-Computing-Project-Group-T4/Project Milestone Dataflow/Preet Patel (fresh-office-375623)$ python subscriber.py
Listening for messages on projects/fresh-office-375623/subscriptions/smartmeter_out-sub..

Received message.
b'{"time": 1676590326.8080473, "profile_name": "boston", "humidity": 88.81128551706747, "temperature": 126.60426431310033, "pressure": 0.1572971266138212}'

Received message.
b'{"time": 1676590326.927192, "profile_name": "denver", "humidity": 41.40049941584857, "temperature": 148.63848696683266, "pressure": 0.34601421900301055}'

Received message.
b'{"time": 1676590326.7734096, "profile_name": "losang", "humidity": 80.84336444789868, "temperature": 147.37514934914896, "pressure": 0.18648617925102082}'

Received message.
b'{"time": 1676590326.8851655, "profile_name": "losang", "humidity": 35.03669363160504, "temperature": 159.24584315129744, "pressure": 0.20539259909311183}'

Received message.
b'{"time": 1676590326.6987329, "profile_name": "losang", "humidity": 68.97735440304004, "temperature": 159.16283097928516, "pressure": 0.17429830957722167}'

Received message.
b'{"time": 1676590326.8460057, "profile_name": "denver", "humidity": 35.897532504383776, "temperature": 73.63591176365847, "pressure": 0.254074093385101}'

Received message.
b'{"time": 1676590326.7317748, "profile_name": "denver", "humidity": 21.116235774854243, "temperature": 166.9233885971503, "pressure": 0.2523506449375212}'
```

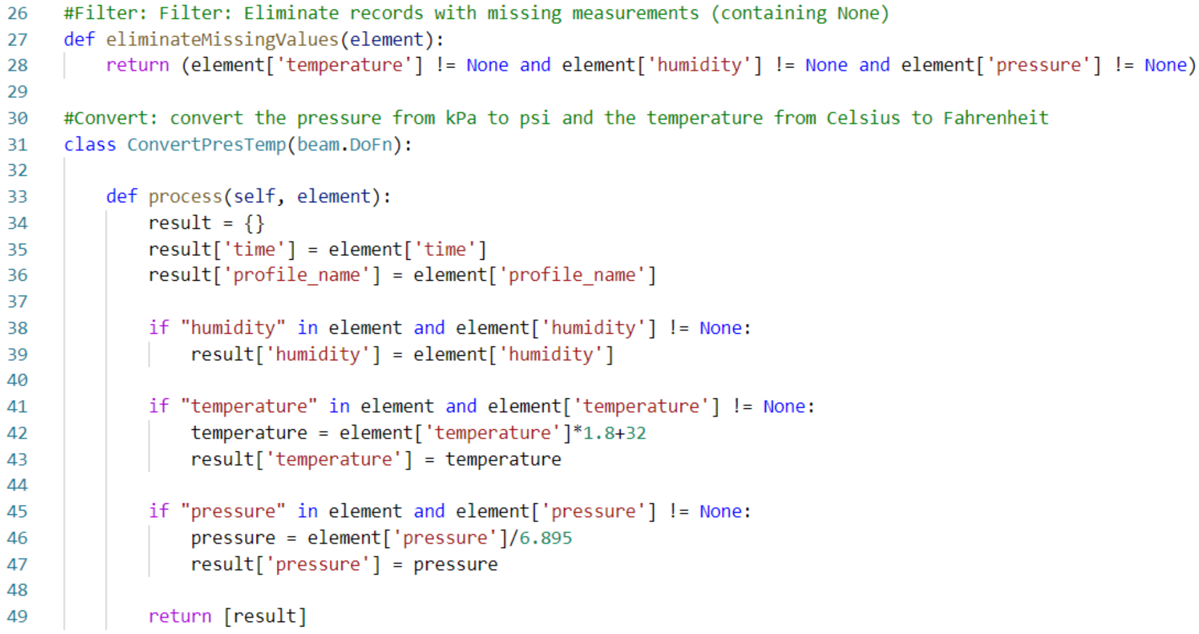
Pipeline Description:

The following images of the code can be used to describe the pipeline that was created to execute this design task. The first step of the pipeline 'Read from Pub/Sub', consumes the topic that was provided as the source of input to receive the generated smart meter measurements data. The second step is 'toDict', which will convert the data which is received in the form of JSON (from the messages that are consumed from the input topic) to a map (dictionary). The third step of the pipeline is the 'filterData' step, in which the current record would be filtered out of the job if it has any missing measurement values (temperature, humidity, and pressure). For this step, we used the beam.Filter command which is provided to filter data out according to any provided condition. We created a function called eliminateMissingValues that returns true or false depending on if the provided data contains all three measurements (temperature, humidity, and pressure). The next step in the pipeline is the ‘ConvertData’ step which will execute the ConvertPresTemp stage function that converts the temperature measurement from Celsius to Fahrenheit and the pressure measurement from kPa to psi using the provided equations. Once the data has been filtered and converted as per the job requirements of the design task, in the second last stage (called ‘to byte’) the result map will be converted to a JSON again and encoded so that it can be published to the ‘output’ topic. In the last step of the pipeline, ‘to Pub/Sub’, will publish the converted output data to the output topic so that the output can then get consumed by any consumers of the topic.



```
def run(argv=None):
    parser = argparse.ArgumentParser(formatter_class=argparse.ArgumentDefaultsHelpFormatter)
    parser.add_argument('--input', dest='input', required=True,
                        help='Input file to process.')
    parser.add_argument('--output', dest='output', required=True,
                        help='Output file to write results to.')
    known_args, pipeline_args = parser.parse_known_args(argv)
    pipeline_options = PipelineOptions(pipeline_args)
    pipeline_options.view_as(SetupOptions).save_main_session = True;

    with beam.Pipeline(options=pipeline_options) as p:
        readings= (p | "Read from Pub/Sub" >> beam.io.ReadFromPubSub(topic=known_args.input)
            | "toDict" >> beam.Map(lambda x: json.loads(x)));

        filtered = readings | 'filterData' >> beam.Filter(eliminateMissingValues)

        conversions = filtered | 'ConvertData' >> beam.ParDo(ConvertPresTemp())

        (conversions | 'to byte' >> beam.Map(lambda x: json.dumps(x).encode('utf8'))
            |   'to Pub/sub' >> beam.io.WriteToPubSub(topic=known_args.output));
```

eliminateMissingValues and ConvertPresTemp functions:



```
#Filter: Filter: Eliminate records with missing measurements (containing None)
def eliminateMissingValues(element):
    return (element['temperature'] != None and element['humidity'] != None and element['pressure'] != None)

#Convert: convert the pressure from kPa to psi and the temperature from Celsius to Fahrenheit
class ConvertPresTemp(beam.DoFn):

    def process(self, element):
        result = {}
        result['time'] = element['time']
        result['profile_name'] = element['profile_name']

        if "humidity" in element and element['humidity'] != None:
            result['humidity'] = element['humidity']

        if "temperature" in element and element['temperature'] != None:
            temperature = element['temperature']*1.8+32
            result['temperature'] = temperature

        if "pressure" in element and element['pressure'] != None:
            pressure = element['pressure']/6.895
            result['pressure'] = pressure

        return [result]
```