Project Milestone 4 - Data Processing: Dataflow- apache beam

Esam Uddin - 100711116

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.

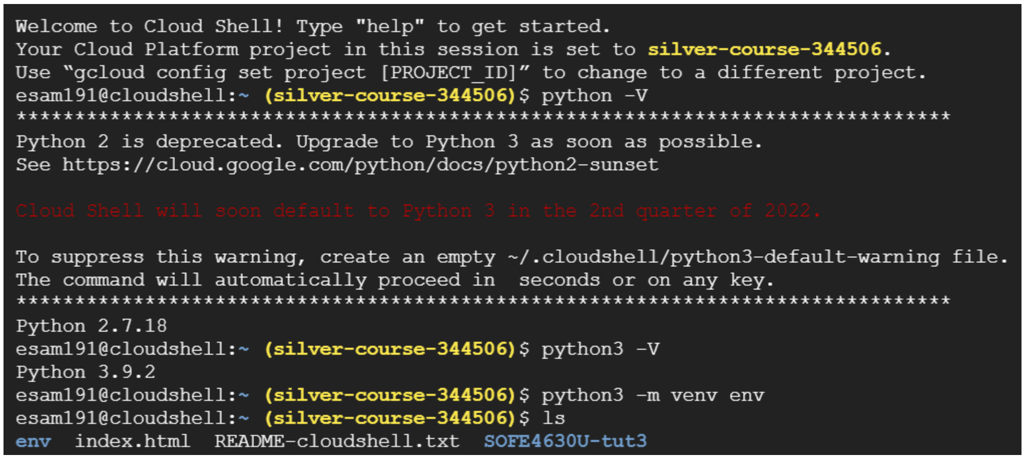
Procedure:

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

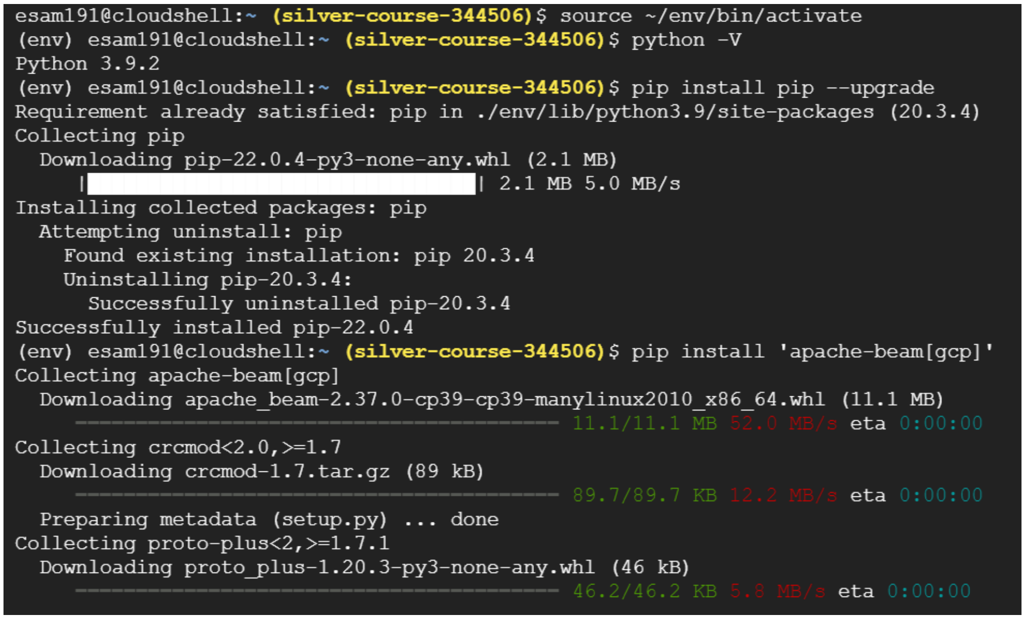
2. **Watch the following video Describing how to apply MapReduce to count the words within a certain document.**
3. **Follow the following video to set up the GCP environment for Dataflow and run wordcount examples.**

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

   Part 1: creating python environment

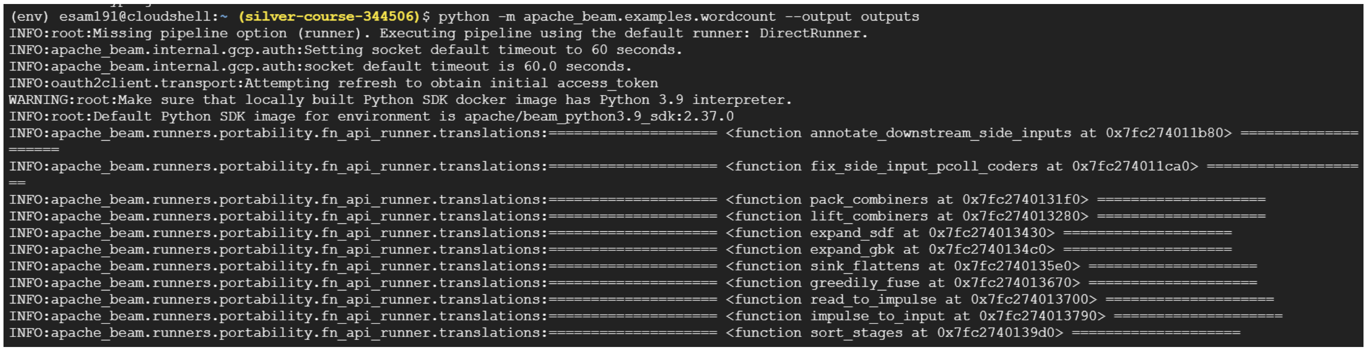


```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to silver-course-344506.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
esam191@cloudshell:~ (silver-course-344506)$ python -V
********************************************************************************
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)$ python3 -m venv env
esam191@cloudshell:~ (silver-course-344506)$ ls
env  index.html  README-cloudshell.txt  SOFE4630U-tut3
```

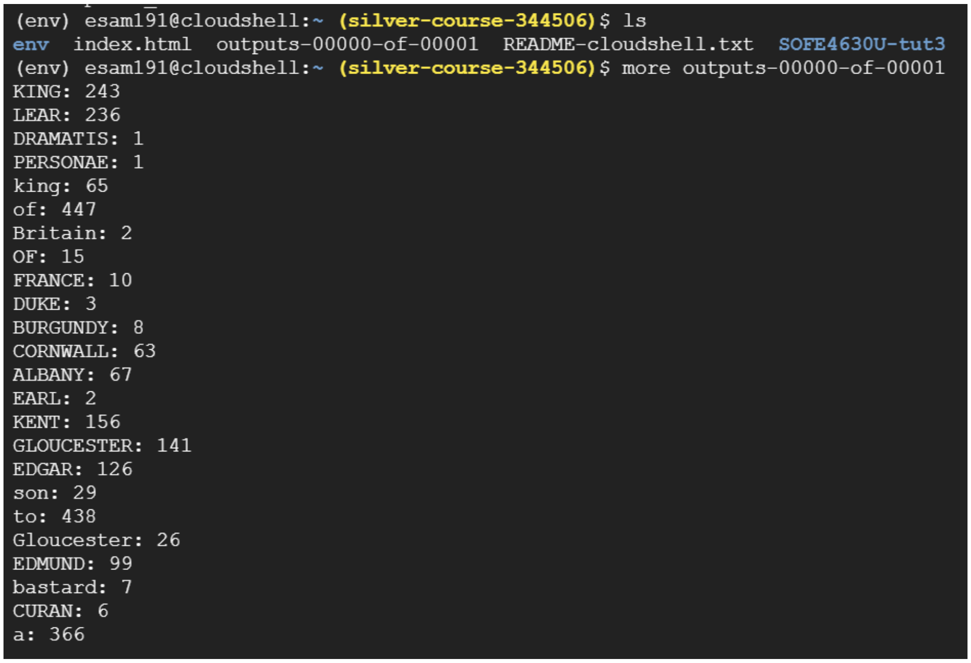


```
esam191@cloudshell:~ (silver-course-344506)$ source ~/env/bin/activate
(env) esam191@cloudshell:~ (silver-course-344506)$ python -V
Python 3.9.2
(env) esam191@cloudshell:~ (silver-course-344506)$ pip install pip --upgrade
Requirement already satisfied: pip in ./env/lib/python3.9/site-packages (20.3.4)
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
```

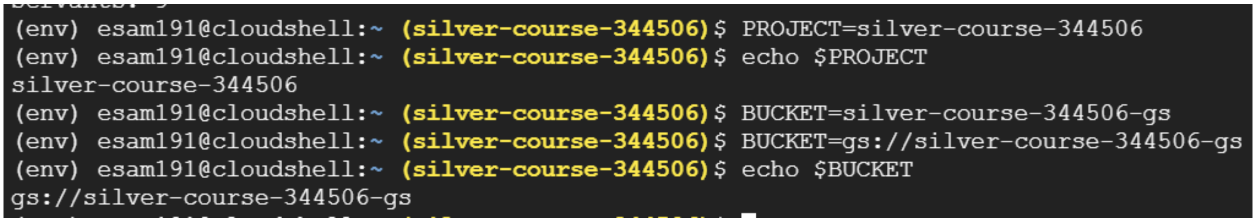
Part 2:testing wordcount example + creating cloud storage



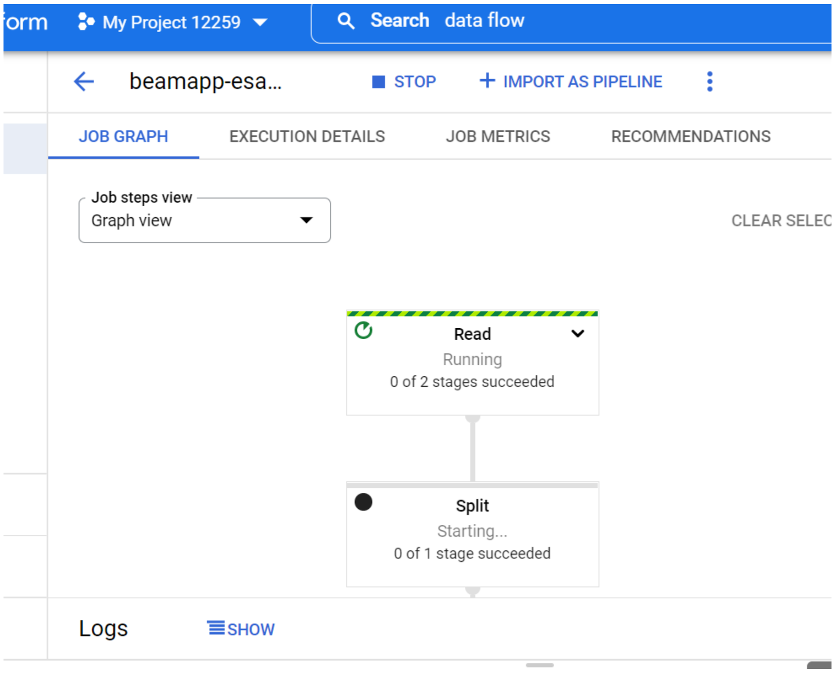
```
(env) esam191@cloudshell:~ (silver-course-344506)$ python -m apache_beam.examples.wordcount --output outputs
INFO:root:Missing pipeline option (runner). Executing pipeline using the default runner: DirectRunner.
INFO:apache_beam.internal.gcp.auth:Setting socket default timeout to 60 seconds.
INFO:apache_beam.internal.gcp.auth:socket 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> ====================
```

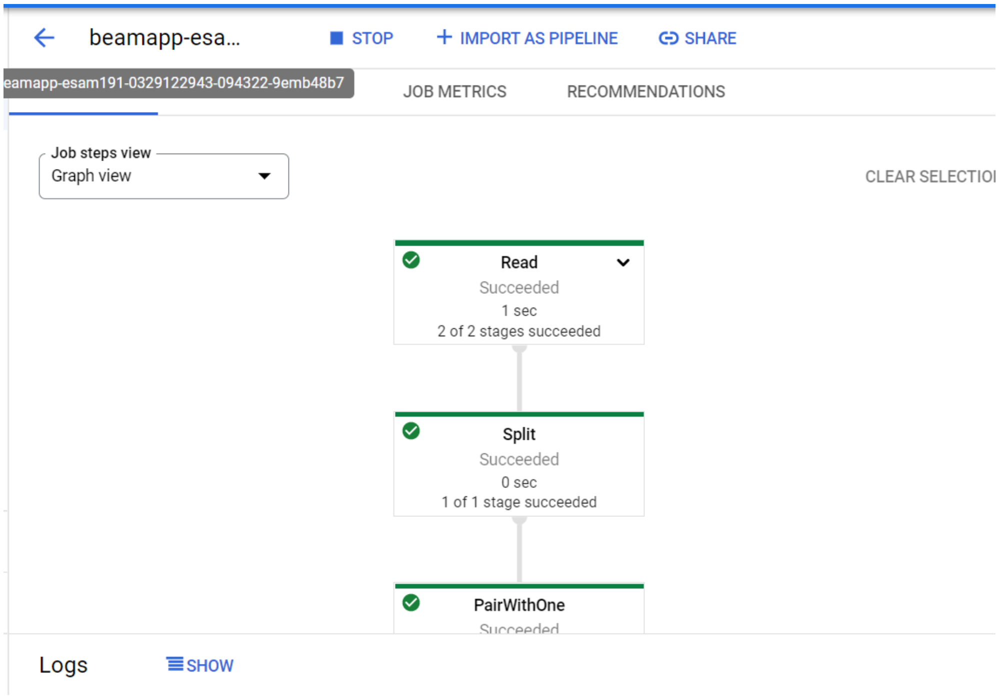


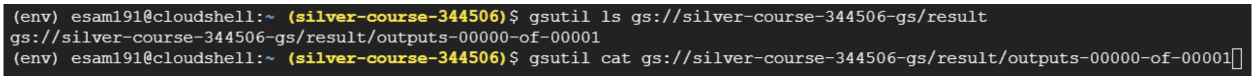
```
(env) esam191@cloudshell:~ (silver-course-344506)$ ls
env  index.html  outputs-00000-of-00001  README-cloudshell.txt  SOFE4630U-tut3
(env) esam191@cloudshell:~ (silver-course-344506)$ more outputs-00000-of-00001
KING: 243
LEAR: 236
DRAMATIS: 1
PERSONAE: 1
king: 65
of: 447
Britain: 2
OF: 15
FRANCE: 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
```



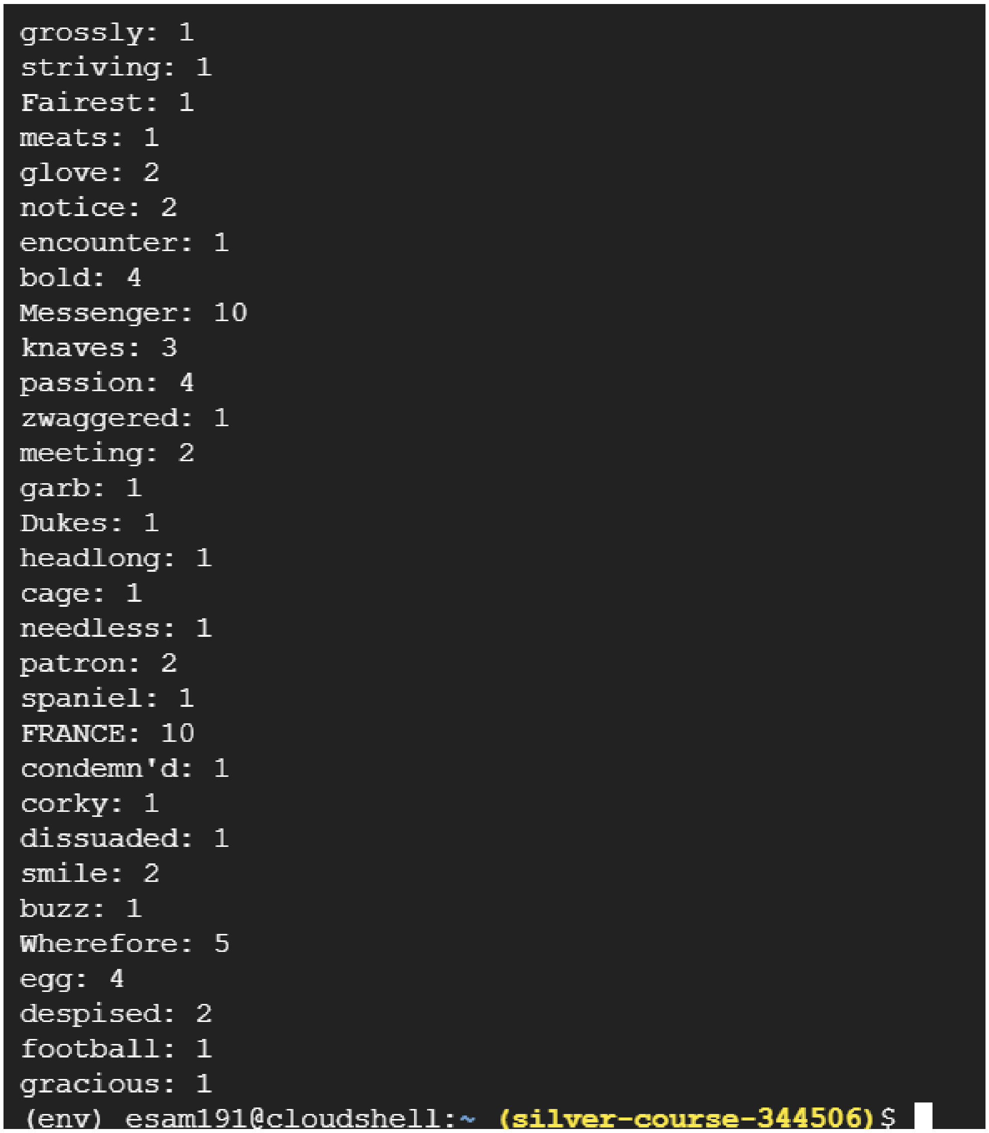
```
(env) esam191@cloudshell:~ (silver-course-344506)$ PROJECT=silver-course-344506
(env) esam191@cloudshell:~ (silver-course-344506)$ echo $PROJECT
silver-course-344506
(env) esam191@cloudshell:~ (silver-course-344506)$ BUCKET=silver-course-344506-gs
(env) esam191@cloudshell:~ (silver-course-344506)$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



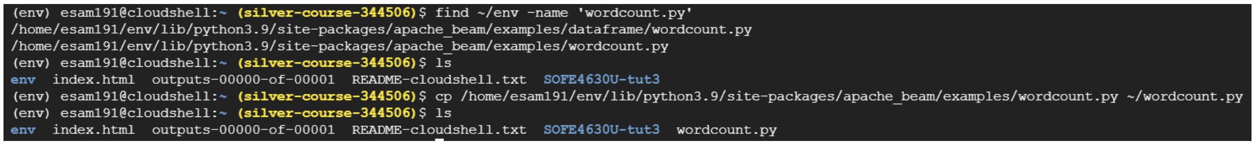




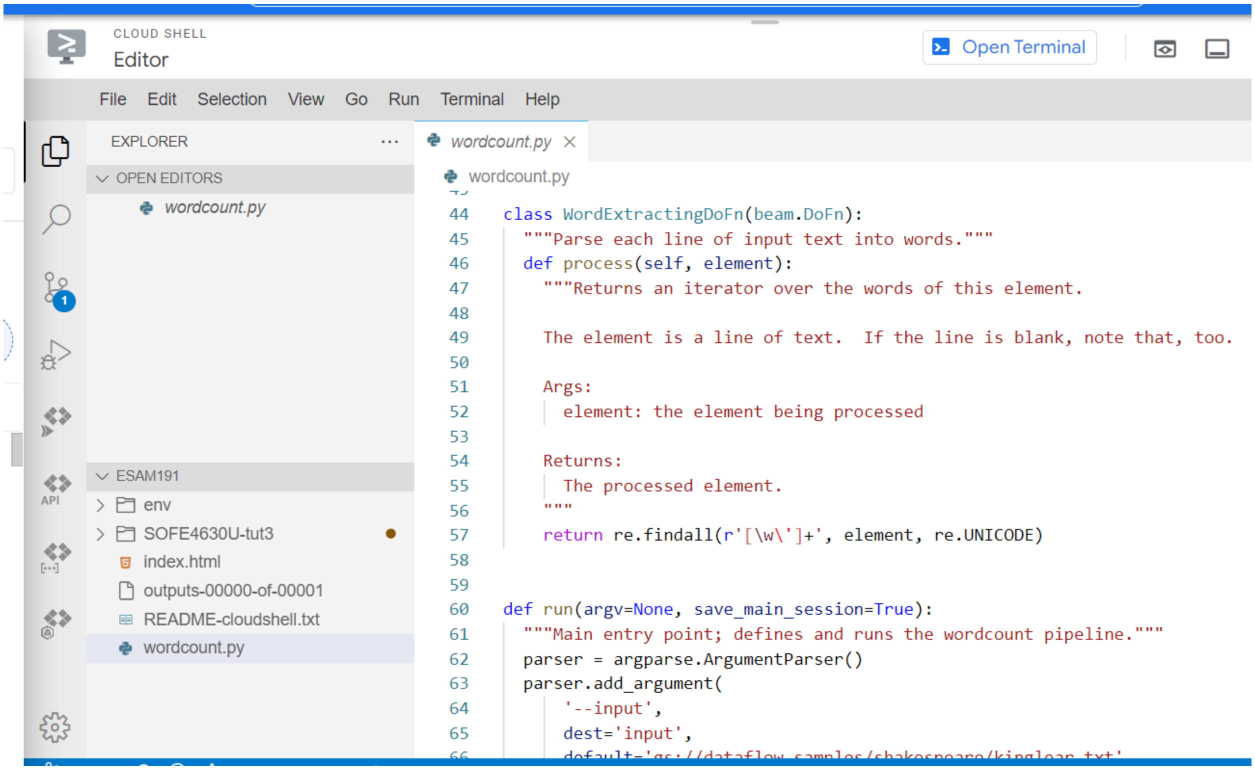
```
(env) esam191@cloudshell:~ (silver-course-344506)$ gsutil ls gs://silver-course-344506-gs/result
gs://silver-course-344506-gs/result/outputs-00000-of-00001
(env) esam191@cloudshell:~ (silver-course-344506)$ gsutil cat gs://silver-course-344506-gs/result/outputs-00000-of-00001
```



```
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',
      default='gs://dataflow-samples/shakespeare/kinglear.txt',
```

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

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.**

   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

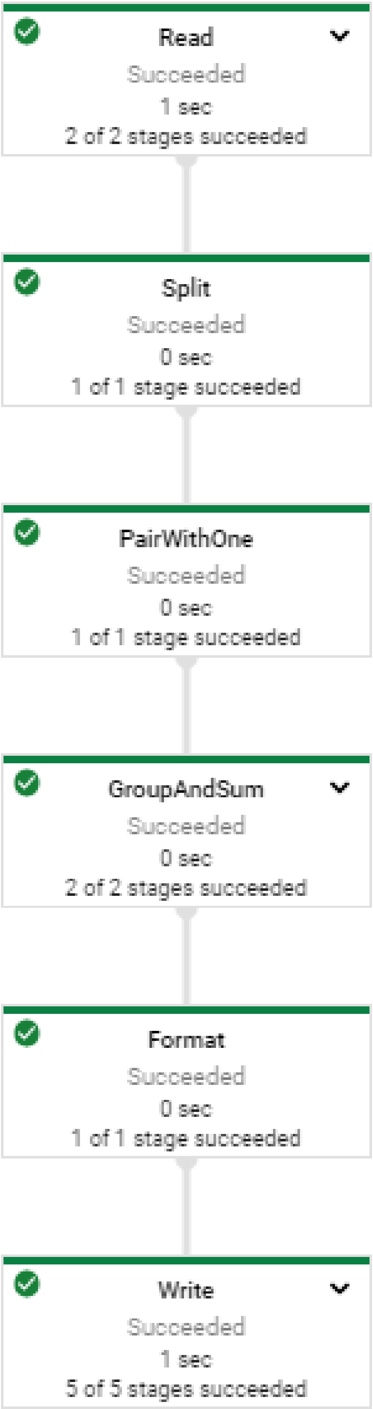
7. **Suggest a practical application using both stream and batch processing that can be applied to a given dataset. It's expected to use the dataset uploaded in the third milestone but you can use any other dataset. If you decide to use another dataset, It should maintain both variety and huge volume.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

    - **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