Beam/Dataflow setup:

https://github.com/cs327e-spring2020/snippets/wiki/Beam--%26-Dataflow-Setup

```
python -m apache_beam.examples.wordcount \
--project $PROJECT_ID \
--runner DataflowRunner \
--staging_location gs://$BUCKET/staging \
--temp_location gs://$BUCKET/temp \
--output gs://$BUCKET/output
```

*Replace $PROJECT_ID and $BUCKET with your project id and bucket name.

*Don't include the dollar sign.

Dataflow

allows processing batch data and streaming data using the same code, which is unique feature of the Dataflow.

Apache Beam

Pipeline:

- DAG, nodes = Transforms, edges = PCollections
- Executed as a single unit.
- A Pipeline encapsulates your entire data processing task, from start to finish. This includes reading input data, transforming that data, and writing output data. All Beam driver programs must create a Pipeline. When you create the Pipeline, you must also specify the execution options that tell the Pipeline where and how to run.

PCollection:

- A collection of bounded or unbounded elements
- Immutable
- Everytime we run a transform, we create a new PCollection.
- Our pipeline typically creates an initial PCollection by reading data from an external data source, but you can also create a PCollection from in-memory data within your driver program. From there, PCollections are the inputs and outputs for each step in your pipeline.

* Bounded Data: Bound data is finite and unchanging data, where everything is known about the set of data.
* Unbounded Data: Unbound data is unpredictable, infinite, and not always sequential.

Transform:

- Data processing operations
- Serializable: Converted to byte stream to transfer over the network
- Parallelizable: Many instances will be running it as subsets of the data
  will be using it
- Idempotent: safe to apply multiple times leading to similar results.
- [Output PCollection] = [Input PCollection] | [Transform]

Pardo

- ParDo is a Beam transform for generic parallel processing.

Helpful link to understand Apache Beam python syntax

https://stackoverflow.com/questions/43796046/explain-apache-beam-python-syntax