CS356 -lab10:1g - Branden-Codd, 940428984

Note: had a lot of issues with stuff running past step 12, assuming this will be my gremlin lab as mentioned in slack.

1. Dataproc Lab #1 (π)

- No screenshots or observations

2. Calculating π

- No screenshots or observations

3. Code

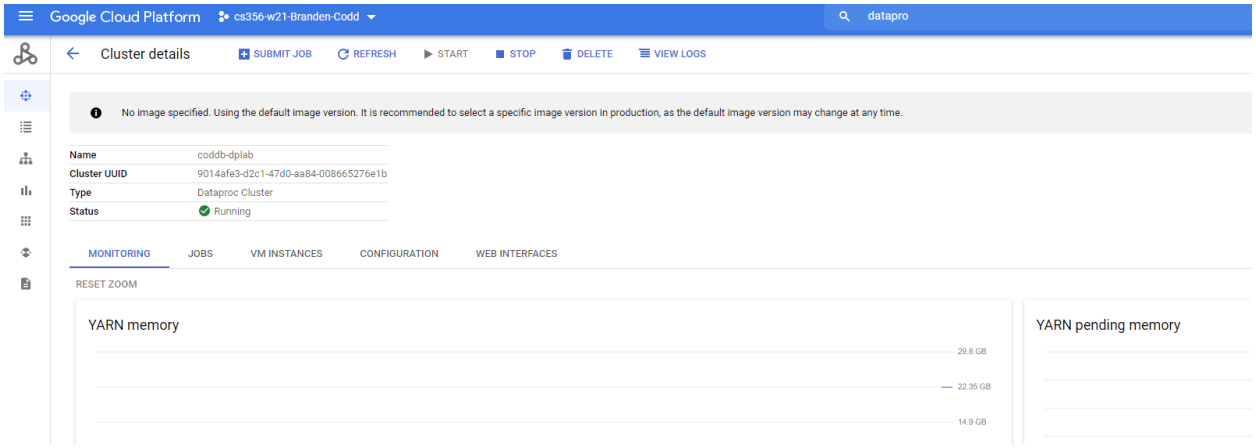
- No screenshots or observations

4. Dataproc setup

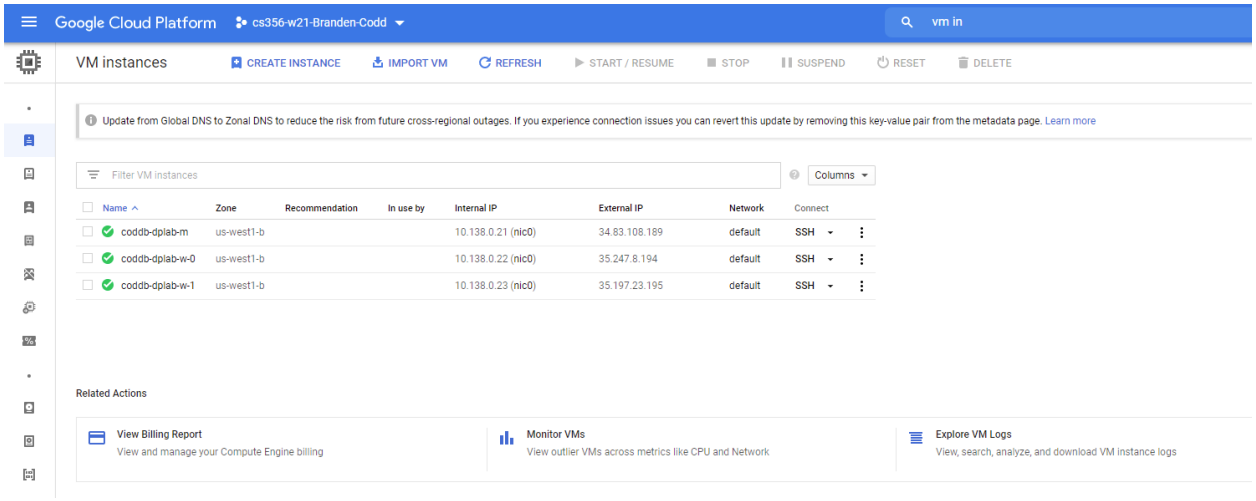
- No screenshots or observations

5. Create Compute Engine cluster

- View the cluster in the web console of Dataproc and take a screenshot.



- View the nodes of the cluster in the web console of Compute Engine and take a screenshot:



6. Run computation

```
coddb@cloudshell:~ (cs356-w21-branden-codd)$ date
Thu 18 Mar 2021 06:54:37 AM UTC
coddb@cloudshell:~ (cs356-w21-branden-codd)$
coddb@cloudshell:~ (cs356-w21-branden-codd)$ gcloud dataproc jobs submit spark --cluster ${CLUSTERNAME} \
>   --class org.apache.spark.examples.SparkPi \
>   --jars file:///usr/lib/spark/examples/jars/spark-examples.jar -- 1000 \
>   >& output.txt &
[1] 1138
```

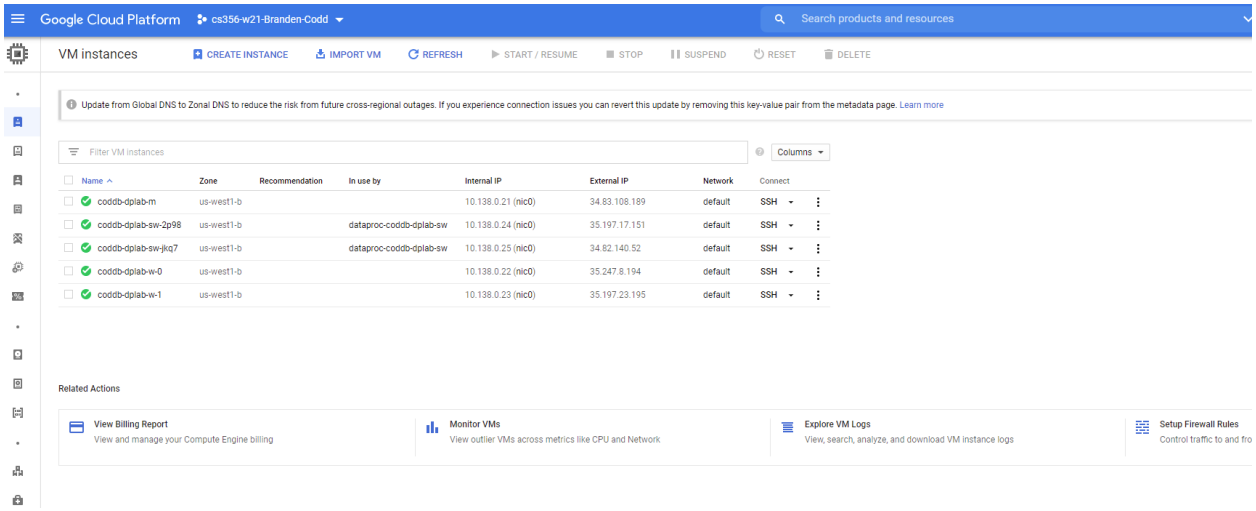
```
coddb@cloudshell:~ (cs356-w21-branden-codd)$ gcloud dataproc jobs list --cluster ${CLUSTERNAME}

JOB_ID                            TYPE   STATUS
741b07c84e3d46298ca2c2a3ef59f2c7  spark  DONE
220436e1732c4f11bae7bdc34dc00f6d  spark  DONE
02036a1e2ae644d8875ac706170618b  spark  DONE
[1]+  Done                    gcloud dataproc jobs submit spark --cluster ${CLUSTERNAME} --class org.apache.spark.examples.SparkPi --jars file:///usr/lib/spark/examples/jars/spark-examples.jar -- 1000 &> output.txt
coddb@cloudshell:~ (cs356-w21-branden-codd)$
coddb@cloudshell:~ (cs356-w21-branden-codd)$ date
Thu 18 Mar 2021 06:55:04 AM UTC
```

For your lab notebook:

- How long did the job take to execute?
    - 67 seconds
- Examine output.txt (or the console) and show the estimate of π calculated.
    - Pi is roughly 3.14168799141688

7. Scale cluster

- Take a screenshot to show the new nodes



8. Run computation again

```
Thu 18 Mar 2021 07:00:57 AM UTC
coddb@cloudshell:~ (cs356-w21-branden-codd)$
coddb@cloudshell:~ (cs356-w21-branden-codd)$ gcloud dataproc jobs submit spark --cluster ${CLUSTERNAME} \
>   --class org.apache.spark.examples.SparkPi \
>   --jars file:///usr/lib/spark/examples/jars/spark-examples.jar -- 1000 \
>   >& output2.txt &
[1] 1296
coddb@cloudshell:~ (cs356-w21-branden-codd)$ gcloud dataproc jobs list --cluster ${CLUSTERNAME}
JOB_ID                            TYPE   STATUS
6793b3117bfa4314928f31dd8317396c  spark  RUNNING
741b07c84e3d46298ca2c2a3ef59f2c7  spark  DONE
220436e1732c4f11bae7bdc34dc00f6d  spark  DONE
02036a1e2ae644d8875ac7061706818b  spark  DONE
coddb@cloudshell:~ (cs356-w21-branden-codd)$
coddb@cloudshell:~ (cs356-w21-branden-codd)$ dategcloud dataproc jobs list --cluster ${CLUSTERNAME}
-bash: dategcloud: command not found
coddb@cloudshell:~ (cs356-w21-branden-codd)$
coddb@cloudshell:~ (cs356-w21-branden-codd)$ date
Thu 18 Mar 2021 07:01:14 AM UTC
coddb@cloudshell:~ (cs356-w21-branden-codd)$ gcloud dataproc jobs list --cluster ${CLUSTERNAME}

JOB_ID                            TYPE   STATUS
6793b3117bfa4314928f31dd8317396c  spark  RUNNING
741b07c84e3d46298ca2c2a3ef59f2c7  spark  DONE
220436e1732c4f11bae7bdc34dc00f6d  spark  DONE
02036a1e2ae644d8875ac7061706818b  spark  DONE
coddb@cloudshell:~ (cs356-w21-branden-codd)$
coddb@cloudshell:~ (cs356-w21-branden-codd)$ date
Thu 18 Mar 2021 07:01:16 AM UTC
coddb@cloudshell:~ (cs356-w21-branden-codd)$ gcloud dataproc jobs list --cluster ${CLUSTERNAME}

JOB_ID                            TYPE   STATUS
6793b3117bfa4314928f31dd8317396c  spark  RUNNING
741b07c84e3d46298ca2c2a3ef59f2c7  spark  DONE
220436e1732c4f11bae7bdc34dc00f6d  spark  DONE
02036a1e2ae644d8875ac7061706818b  spark  DONE
coddb@cloudshell:~ (cs356-w21-branden-codd)$
coddb@cloudshell:~ (cs356-w21-branden-codd)$ date
Thu 18 Mar 2021 07:01:23 AM UTC
coddb@cloudshell:~ (cs356-w21-branden-codd)$ gcloud dataproc jobs list --cluster ${CLUSTERNAME}

JOB_ID                            TYPE   STATUS
6793b3117bfa4314928f31dd8317396c  spark  DONE
741b07c84e3d46298ca2c2a3ef59f2c7  spark  DONE
220436e1732c4f11bae7bdc34dc00f6d  spark  DONE
02036a1e2ae644d8875ac7061706818b  spark  DONE
[1]+  Done                    gcloud dataproc jobs submit spark --cluster ${CLUSTERNAME} --class org.apache.spark.examples.SparkPi --jars file:///usr/lib/spark/examples/jars/spark-examples.jar -- 1000 &> output2.txt
coddb@cloudshell:~ (cs356-w21-branden-codd)$
coddb@cloudshell:~ (cs356-w21-branden-codd)$ date
Thu 18 Mar 2021 07:01:33 AM UTC
coddb@cloudshell:~ (cs356-w21-branden-codd)$
```

For your lab notebook:

- How long did the job take to execute? How much faster did it take?
    - 36 seconds
- Examine output2.txt and show the estimate of π calculated.
    - Pi is roughly 3.1414578714145787

9. Clean up

- No screenshots or observations

10. Dataflow Lab #1 (Java package popularity)

- No screenshots or observations

11. Setup

- No screenshots or observations

12. Beam code

Answer the following questions for your lab notebook.

- Where is the input taken from by default?

```
63      # find most used packages
64      (p
65          | 'GetJava' >> beam.io.ReadFromText(input)
```

- Where does the output go by default?

```
| 'write' >> beam.io.WriteToText(output_prefix)
```

- Examine both the getPackages() function and the splitPackageName() function. What operation does the PackageUse() transform implement?
    - Yield?
        - Im not sure what this question is asking
- Look up Beam's CombinePerKey. What operation does the TotalUse operation implement?
    - Bitwise right shift

The operations in the pipeline mimic a Map-Reduce pattern, demonstrating Beam's ability to support it.

Answer the following question for your lab notebook.

- Which operations correspond to a "Map"?

```
| 'GetImports' >> beam.FlatMap(lambda line: startsWith(line, keyword))
| 'PackageUse' >> beam.FlatMap(lambda line: packageUse(line, keyword))
```

- Which operation corresponds to a "Shuffle-Reduce"?

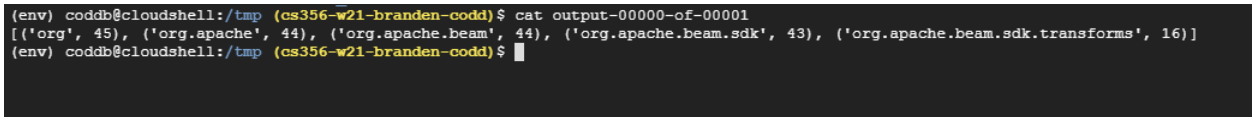
```
| 'TotalUse' >> beam.CombinePerKey(sum)
```

- Which operation corresponds to a "Reduce"?

```
| 'Top_5' >> beam.transforms.combiners.Top.Of(5, key=lambda kv: kv[1])
```

13. Run pipeline locally

- Take a screenshot of its contents

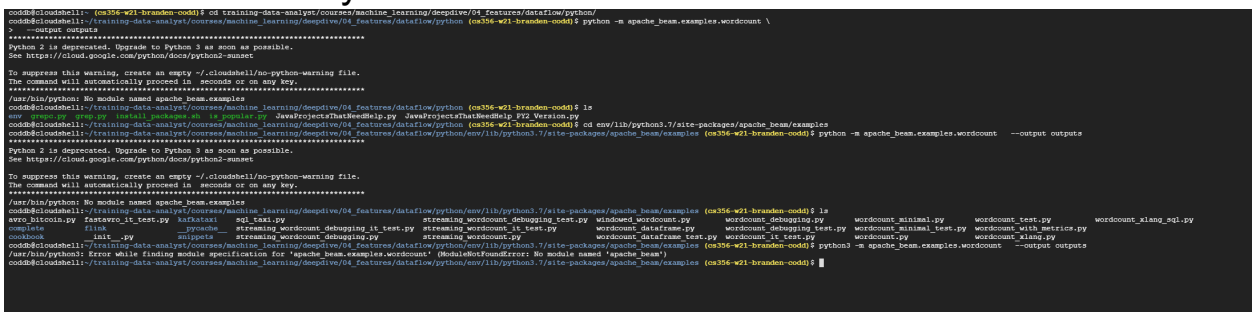


- Explain what the data in this output file corresponds to based on your understanding of the program.
    - Output most used packages

14. Dataflow Lab #2 (Word count)

- What are the names of the stages in the pipeline?
    - Split, pairwithone, groupandsum, format, write
- Describe what each stage does.
    - Split: splits using the Pardo function and word extracitng
    - Pairwithone: maps
    - Groupsandsum: matches up the key
    - Format: using tuple formats result
    - Write: writes output

15. Run code locally



- Had issues with running the script

16. Setup for Cloud Dataflow

- No screenshots or observations

17. Service account setup

- No screenshots or observations

18. Run code using Dataflow runner

- Again having issues running the script and code. Do not have the time to trouble shoot with other finals