

# Run a Big Data Text Processing Pipeline in Cloud Dataflow

 [cloudskillsboost.google/games/2854/labs/17211](https://cloudskillsboost.google/games/2854/labs/17211)

GSP047



## Overview

Dataflow is a unified programming model and a managed service for developing and executing a wide range of data processing patterns including ETL, batch computation, and continuous computation. Because Dataflow is a managed service, it can allocate resources on demand to minimize latency while maintaining high utilization efficiency.

The Dataflow model combines batch and stream processing so developers don't have to make tradeoffs between correctness, cost, and processing time. In this lab, you'll learn how to run a Dataflow pipeline that counts the occurrences of unique words in a text file.

## What you'll learn

- How to create a Maven project with the Cloud Dataflow SDK
- Run an example pipeline using the Cloud Console
- How to delete the associated Cloud Storage bucket and its contents

## Setup and Requirements

### Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

To complete this lab, you need:

Access to a standard internet browser (Chrome browser recommended).

**Note:** Use an Incognito or private browser window to run this lab. This prevents any conflicts between your personal account and the Student account, which may cause extra charges incurred to your personal account.

Time to complete the lab---remember, once you start, you cannot pause a lab.

**Note:** If you already have your own personal Google Cloud account or project, do not use it for this lab to avoid extra charges to your account.

## How to start your lab and sign in to the Google Cloud Console

---

1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is the **Lab Details** panel with the following:
  - The **Open Google Console** button
  - Time remaining
  - The temporary credentials that you must use for this lab
  - Other information, if needed, to step through this lab
2. Click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.

**Tip:** Arrange the tabs in separate windows, side-by-side.

**Note:** If you see the **Choose an account** dialog, click **Use Another Account**.

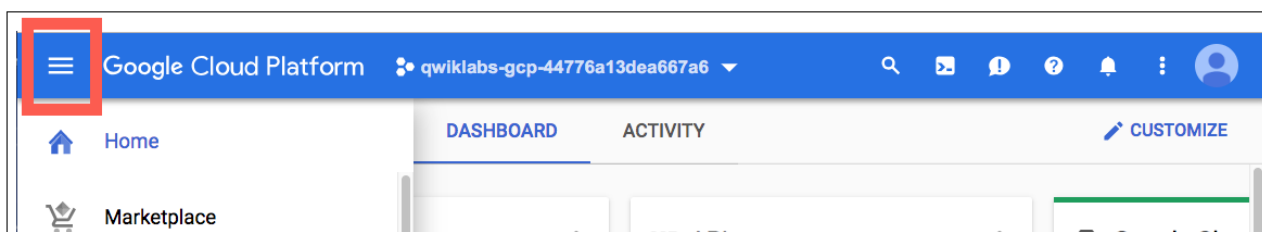
3. If necessary, copy the **Username** from the **Lab Details** panel and paste it into the **Sign in** dialog. Click **Next**.
4. Copy the **Password** from the **Lab Details** panel and paste it into the **Welcome** dialog. Click **Next**.

**Important:** You must use the credentials from the left panel. Do not use your Google Cloud Skills Boost credentials. **Note:** Using your own Google Cloud account for this lab may incur extra charges.

5. Click through the subsequent pages:
  - Accept the terms and conditions.
  - Do not add recovery options or two-factor authentication (because this is a temporary account).
  - Do not sign up for free trials.

After a few moments, the Cloud Console opens in this tab.

**Note:** You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-left.

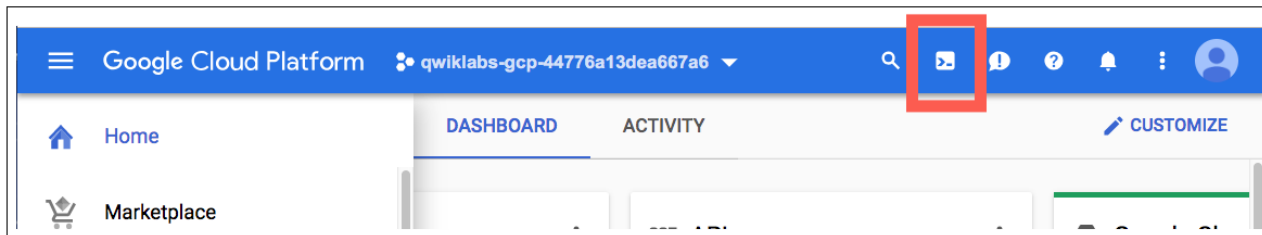


## Activate Cloud Shell

---

Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources.

1. In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.



2. Click **Continue**.

It takes a few moments to provision and connect to the environment. When you are connected, you are already authenticated, and the project is set to your **PROJECT\_ID**. The output contains a line that declares the **PROJECT\_ID** for this session:

Your Cloud Platform project in this session is set to YOUR\_PROJECT\_ID  
`gcloud` is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

3. (Optional) You can list the active account name with this command:

```
gcloud auth list
```

(Output)

ACTIVE: \* ACCOUNT: student-01-xxxxxxxxxxxx@qwiklabs.net To set the active account, run: `$ gcloud config set account 'ACCOUNT'`

4. (Optional) You can list the project ID with this command:

```
gcloud config list project
```

(Output)

```
[core] project = <project_ID>
```

(Example output)

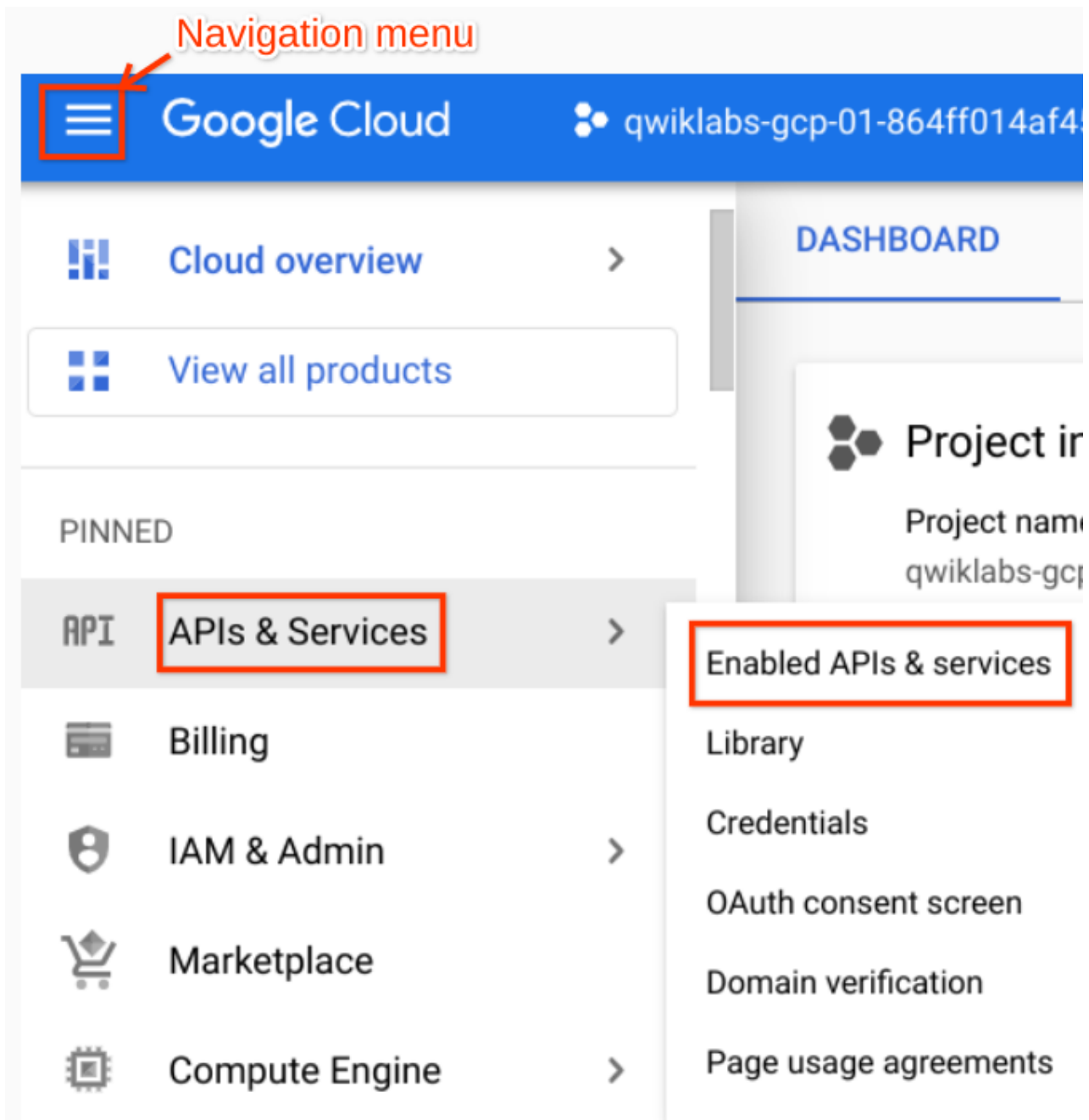
[core] project = qwiklabs-gcp-44776a13dea667a6 For full documentation of `gcloud`, in Google Cloud, Cloud SDK documentation, see the [gcloud command-line tool overview](#).

## Enable the APIs

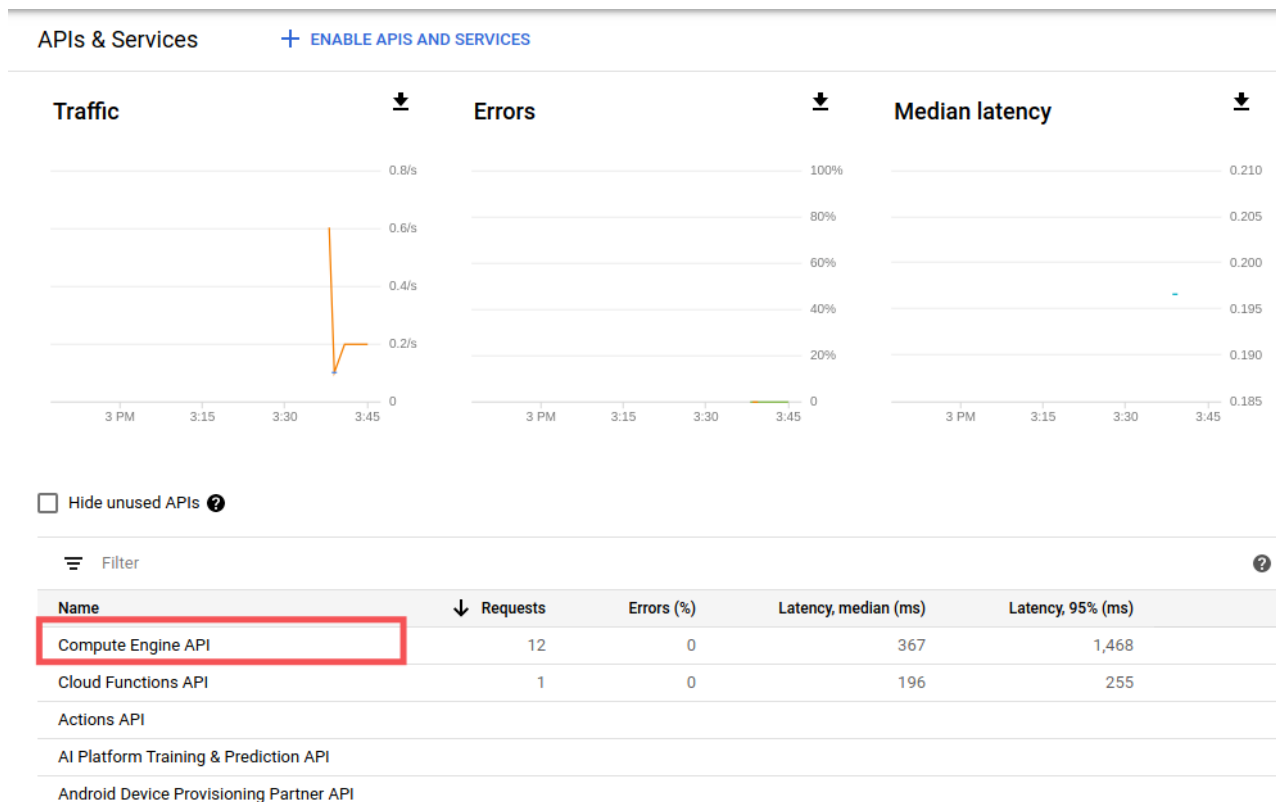
---

A number of APIs must be enabled for this lab to work. This section will show you how to check and if necessary, enable an API.

To check an API status, click **Navigation menu** > **APIs & Services** > **Enabled APIs & services**



Start by checking the status of Compute Engine API. In APIs & services, look down the API list and see if Compute Engine API is listed.



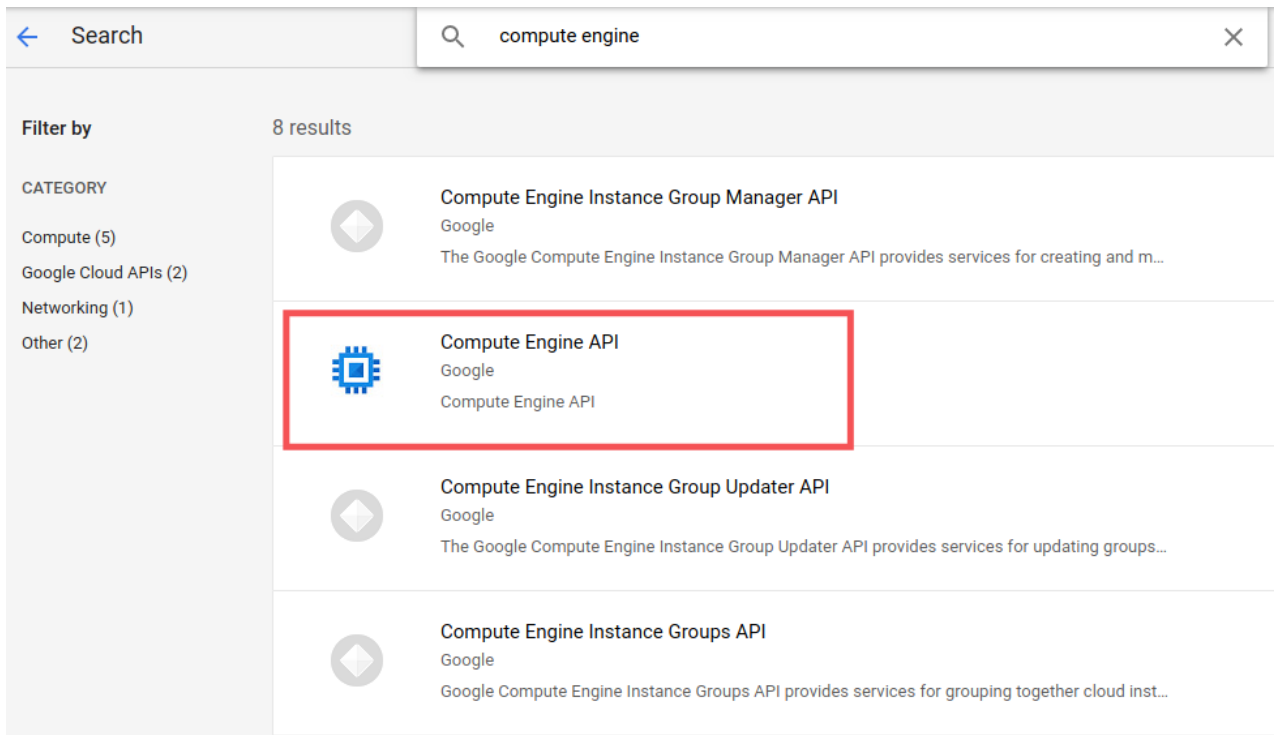
In the example above, Compute Engine API is listed, so it's enabled.

In most cases, all APIs required for the lab are enabled; if any are not, you must enable them.

One way to enable this API is, still in the **APIs & services**, click **ENABLE APIS AND SERVICES**.



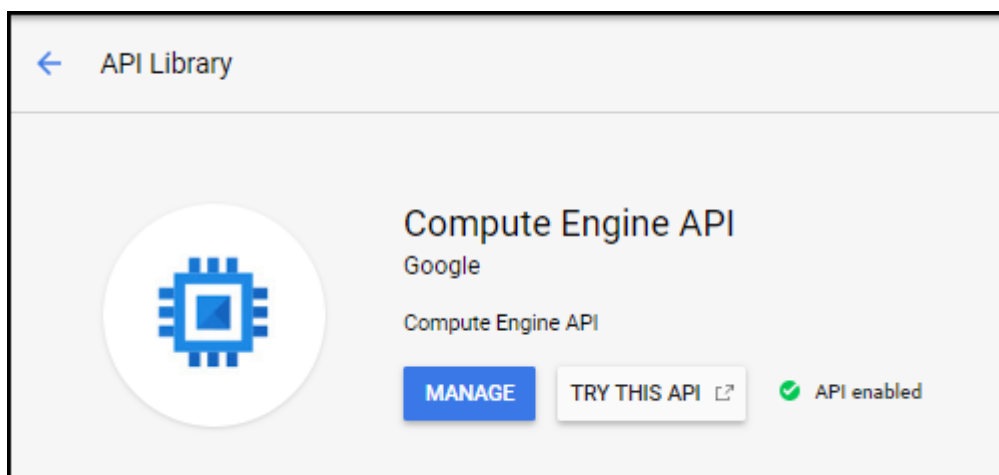
Still using Compute Engine API as an example, type "compute engine" in the search dialog, then click on **Compute Engine API**.



Click on **Compute Engine API** in search results. The API library will display details. The API is enabled. If you see a **Manage** button and an **API enabled** white check in a green circle.

If the API is not enabled, click the **Enable** button.

In the example below, the API is enabled.



Once it has enabled click the arrow to go back to the **API Library**.

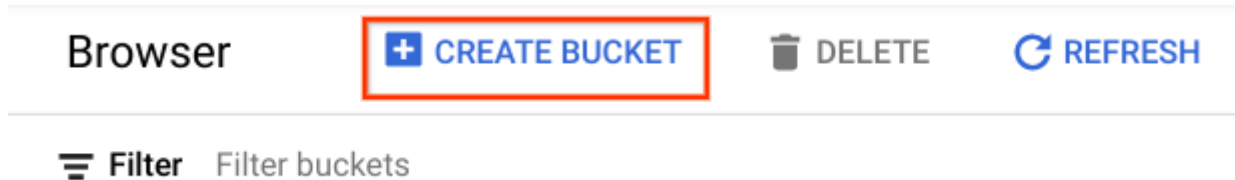
Search for the following APIs and enable them as needed:

- Dataflow API
- Cloud Logging API
- Cloud Storage
- Cloud Storage JSON API
- BigQuery API
- Cloud Pub/Sub API

- Google Cloud Datastore API

## Create a new Cloud Storage bucket

To create a new Cloud Storage bucket, In the [Cloud Console](#), click **Navigation menu** > **Cloud Storage**.



Click **Create bucket** to create a Cloud Storage Bucket.

Enter a unique name, and click **Create**.

After successfully creating a bucket, you are taken to your new, empty, bucket in **Browser**:

**qwiklabs-gcp-00-0f4420e42555**

Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Not public	None

**OBJECTS**   CONFIGURATION   PERMISSIONS   PROTECTION   LIFECYCLE

Buckets > **qwiklabs-gcp-00-0f4420e42555**

[UPLOAD FILES](#)   [UPLOAD FOLDER](#)   [CREATE FOLDER](#)   [MANAGE HOLDS](#)   [DOWNLOAD](#)   [DELETE](#)

Filter by name prefix only ▾   **Filter** Filter objects and folders   ☐ Show deleted data  

<input type="checkbox"/>	Name	Size	Type	Created ?	Storage class	Last modified	Public access ?	Version history ?	Encryption ?	Retention expiration date ?	Holds ?
No rows to display											

## Test Completed Task

Click **Check my progress** to verify your performed task. If you have successfully created a new Cloud Storage bucket, you will see an assessment score.

Create a new Cloud Storage bucket.

## Create a Maven project

In this section, you'll create a Maven project that contains the Cloud Dataflow SDK for Java.

Before you begin, in the Cloud Shell command line, enter the `ls` command to see what's what.

`ls`

You'll see one directory, `README-cloudshell.txt`.

Create the Maven project, enter the `mvn archetype:generate` command:

```
mvn archetype:generate \
  -DarchetypeGroupId=org.apache.beam \
  -DarchetypeArtifactId=beam-sdks-java-maven-archetypes-examples \
  -DarchetypeVersion=2.20.0 \
  -DgroupId=org.example \
  -DartifactId=first-dataflow \
  -Dversion="0.1" \
  -Dpackage=org.apache.beam.examples \
  -DinteractiveMode=false
```

Now you should see a new directory called `first-dataflow` under your current directory. `first-dataflow` contains a Maven project that includes the Cloud Dataflow SDK for Java and example pipelines. You'll see the following output when the build is finished:



```
[INFO] Generating project in Batch mode
[INFO] Archetype repository not defined. Using the one from [org.apache.beam:beam-sdks-java-maven-archetypes-examples:2.8.0] found in catalog remote
Downloaded: https://repo.maven.apache.org/maven2/org/apache/beam/beam-sdks-java-maven-archetypes-examples/2.8.0/beam-sdks-java-maven-archetypes-examples-2.8.0.pom (4 KB at 11.8 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/org/apache/beam/beam-sdks-java-maven-archetypes-examples/2.8.0/beam-sdks-java-maven-archetypes-examples-2.8.0.jar (2966 KB at 3787.6 KB/sec)
[INFO] -----
[INFO] Using following parameters for creating project from Archetype: beam-sdks-java-maven-archetypes-examples:2.8.0
[INFO] -----
[INFO] Parameter: groupId, Value: org.example
[INFO] Parameter: artifactId, Value: first-dataflow
[INFO] Parameter: version, Value: 0.1
[INFO] Parameter: package, Value: org.apache.beam.examples
[INFO] Parameter: packageInPathFormat, Value: org/apache/beam/examples
[INFO] Parameter: package, Value: org.apache.beam.examples
[INFO] Parameter: version, Value: 0.1
[INFO] Parameter: groupId, Value: org.example
[INFO] Parameter: targetPlatform, Value: 1.8
[INFO] Parameter: artifactId, Value: first-dataflow
[INFO] Project created from Archetype in dir: /home/gcpstaging27601_student/first-dataflow
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 57.485 s
[INFO] Finished at: 2018-12-10T14:06:31+05:30
[INFO] Final Memory: 15M/48M
[INFO] -----
gcpstaging27601_student@cloudshell:~ (qwiklabs-gcp-fa5e5305eff572c4) $ ls
first-dataflow  README-cloudshell.txt
gcpstaging27601_student@cloudshell:~ (qwiklabs-gcp-fa5e5305eff572c4) $
```

Check out your directory again:

ls

You'll see a new directory, `first-dataflow`.

## Run a text processing pipeline on Cloud Dataflow

Save your project ID and Cloud Storage bucket names as environment variables. Be sure to replace `<your_project_id>` with your **Project ID**, found in the **Connection Details** section of the lab.

`export PROJECT_ID=<your_project_id>` **Note:** You can retrieve the Project ID with the `gcloud` command and with using ENV variable:

- `gcloud` command: `gcloud config get-value project`
- **ENV Variable:** `echo $DEVSHHELL_PROJECT_ID`

If you use mentioned ways then your final command will be like:

- `export PROJECT_ID=$DEVSHHELL_PROJECT_ID`
- `export PROJECT_ID=$(gcloud config get-value project)`

Now do the same for the Cloud Storage bucket. Remember to replace `<your_bucket_name>` with the bucket name you created earlier.

`export BUCKET_NAME=<your_bucket_name>`

Change to the `first-dataflow/` directory.

`cd first-dataflow`

We're going to run a pipeline called WordCount, which reads text, tokenizes the text lines into individual words, and performs a frequency count on each of those words. While the pipeline is running, we'll take a look at what's happening in each step.

Start the pipeline by running `mvn -Pdataflow-runner compile exec:java`. For the `-project`, `--stagingLocation`, and `--output` arguments, the command below references the environment variables you just set up.

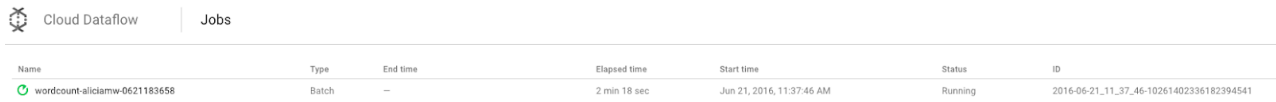
`mvn -Pdataflow-runner compile exec:java \ - Dexec.mainClass=org.apache.beam.examples.WordCount \ -Dexec.args="--`


```
project=${PROJECT_ID} \ --stagingLocation=gs://${BUCKET_NAME}/staging/ \ --  
output=gs://${BUCKET_NAME}/output \ --runner=DataflowRunner"
```

While the job is running, let's find the job in the job list in the Dataflow page in the console.

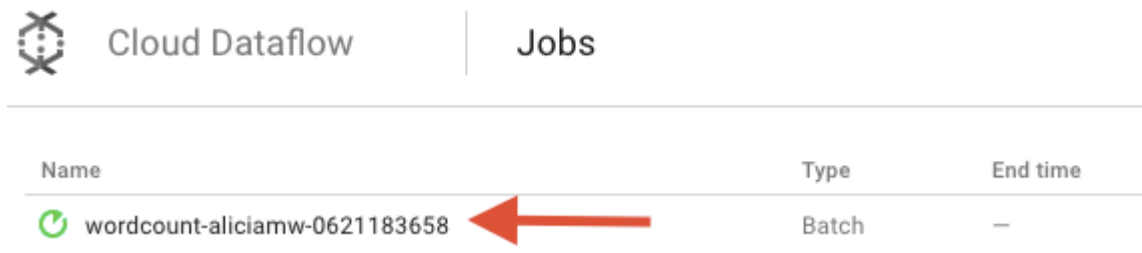
Click the **Navigation menu** > **Dataflow** to open the **Dataflow** page.


You should see your wordcount job with a status of **Running**:



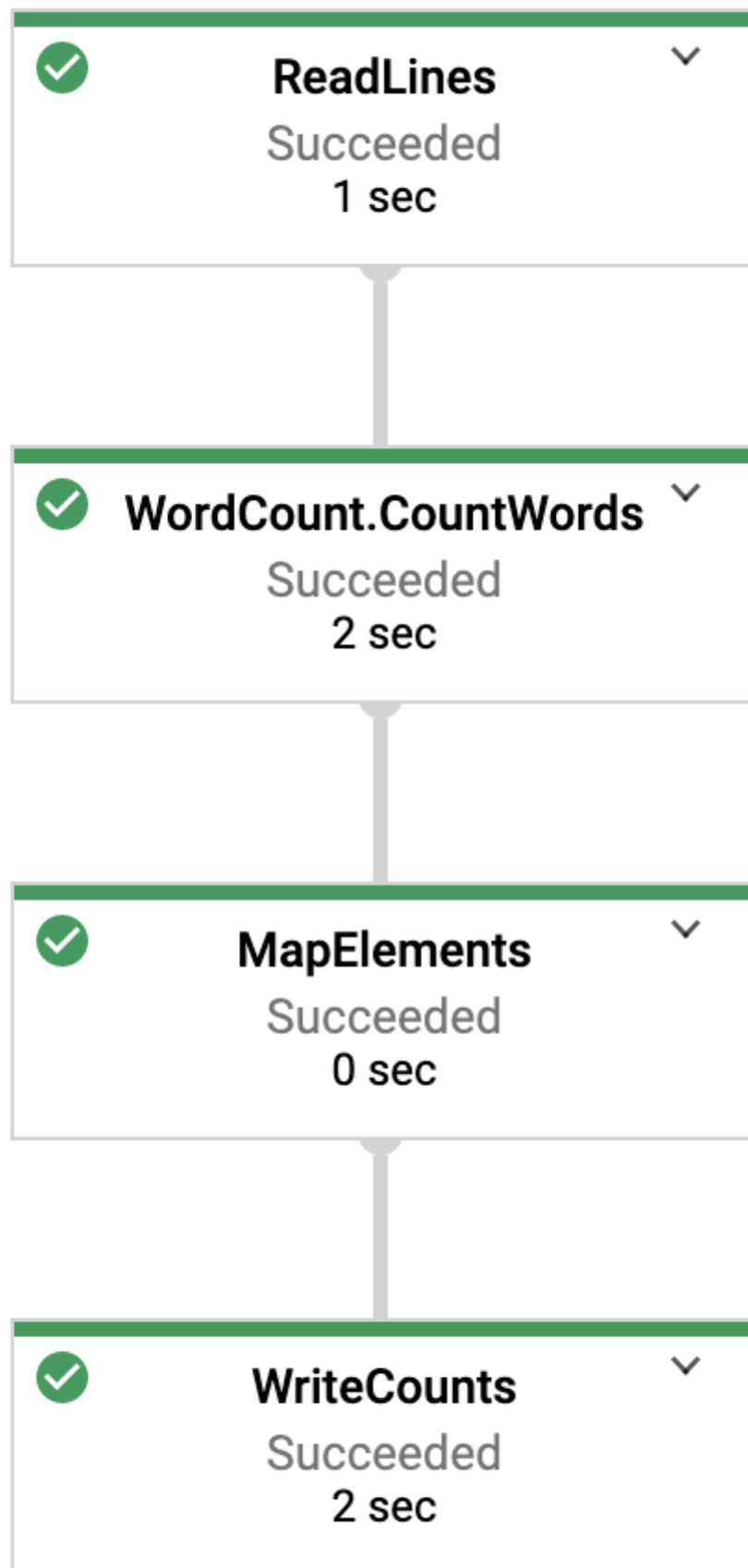
Cloud Dataflow		Jobs				
Name	Type	End time	Elapsed time	Start time	Status	ID
 wordcount-aliciamw-0621183658	Batch	—	2 min 18 sec	Jun 21, 2016, 11:37:46 AM	Running	2016-06-21_11_37_46-10261402336182394541

Let's look at the pipeline parameters. Start by clicking on the name of your job:



Cloud Dataflow		Jobs	
Name	Type	End time	
 wordcount-aliciamw-0621183658	Batch	—	

When you select a job, you'll see the **execution graph**. A pipeline's execution graph represents each transform in the pipeline as a box that contains the transform name and some status information. You can click on the carat in the top right corner of each step to see more details:



Let's see how the pipeline transforms the data at each step:

- **Read:** The pipeline reads from an input source. In this case, it's a text file from Cloud Storage with the entire text of the Shakespeare play *King Lear*. Our pipeline reads the file line by line and outputs each a `PCollection`, where each line in the text file is an element in the collection.
- **CountWords:** The `CountWords` step has two parts. First, it uses a parallel do function (ParDo) named `ExtractWords` to tokenize each line into individual words. The output of `ExtractWords` is a new `PCollection` where each element is a word. The next step, `count`, utilizes a transform provided by the Dataflow SDK which returns key, value pairs where the key is a unique word and the value is the number of times it occurs. Here's the method implementing `CountWords`. You can check out the full `WordCount.java` file on [GitHub](#):

```
/** * A PTransform that converts a PCollection containing lines of text * into a  
PCollection of formatted word counts. */ public static class CountWords extends  
PTransform<PCollection<String>, PCollection<KV<String, Long>>> { @Override public  
PCollection<KV<String, Long>> apply(PCollection<String> lines) { // Convert lines of  
text into individual words. PCollection<String> words = lines.apply( ParDo.of(new  
ExtractWordsFn())); // Count the number of times each word occurs.  
PCollection<KV<String, Long>> wordCounts = words.apply(Count.  
<String>perElement()); return wordCounts; } }
```

**FormatAsText:** This is a function that formats each key, value pair into a printable string. Here's the `FormatAsText` transform to implement this:

```
/** A SimpleFunction that converts a Word and Count into a printable string. */ public  
static class FormatAsTextFn extends SimpleFunction<KV<String, Long>, String> {  
@Override public String apply(KV<String, Long> input) { return input.getKey() + ": " +  
input.getValue(); } }
```

**WriteCounts:** The printable strings are written into multiple sharded text files.

We'll take a look at the resulting output from the pipeline in a few minutes.

Now take a look at the **Job info** section to the right of the execution graph, which includes **Pipeline options** that were included in the `mvn -Pdataflow-runner compile exec:java` command:

## Job info



Job name	wordcount-student0040657f9effb492-1217101228-f9c81289
Job ID	2020-12-17_02_12_52-2995612318743217849
Job type	Batch
Job status	Succeeded
SDK version	Apache Beam SDK for Java 2.20.0 A newer version of the SDK family exists and updating is recommended. <a href="#">Learn more</a>
Job region	us-central1
Worker location	us-central1-b
Current workers	0
Latest worker status	Worker pool stopped.
Start time	December 17, 2020 at 3:42:54 PM GMT+5
Elapsed time	6 min 4 sec
Encryption type	Google-managed key

## Resource metrics



Current vCPUs	1
Total vCPU time	0.075 vCPU hr
Current memory	3.75 GB
Total memory time	0.283 GB hr
Current HDD PD	250 GB
Total HDD PD time	18.841 GB hr
Current SSD PD	0 B
Total SSD PD time	0 GB hr

## Custom counters



Filter by counter name, value or step



Counter name	Value	Step
emptyLines	1,663	WordCount.CountWords/ParDo(ExtractWords)
lineLenDistro_COUNT	5,525	WordCount.CountWords/ParDo(ExtractWords)
lineLenDistro_MAX	69	WordCount.CountWords/ParDo(ExtractWords)
lineLenDistro_MEAN	27	WordCount.CountWords/ParDo(ExtractWords)
lineLenDistro_MIN	0	WordCount.CountWords/ParDo(ExtractWords)

## Pipeline options



stagingLocation	gs://qwiklabs-gcp-04-32dbe4d496e9/staging/
appName	WordCount
pipelineUrl	gs://qwiklabs-gcp-04-32dbe4d496e9/staging/pipeline-Boh1hyaJbSh2f0EIN_pkFQ.pb
jobName	wordcount-student0040657f9effb492-1217101228-f9c81289
tempLocation	gs://dataflow-staging-us-central1-602339540504/temp/
project	qwiklabs-gcp-04-32dbe4d496e9
userAgent	Apache_Beam_SDK_for_Java/2.20.0(JDK_11_environment)
filesToStage	[/home/student_04_657f0effb492/first-dataflow/target/classes /home/student_04_657f0effb492/m2/r

You can also see **Custom counters** for the pipeline, which in this case shows how many empty lines have been encountered so far during execution. Add new counters to your pipeline to track application-specific metrics.


Click the **LOGS > JOB LOGS** tab to see specific error messages.

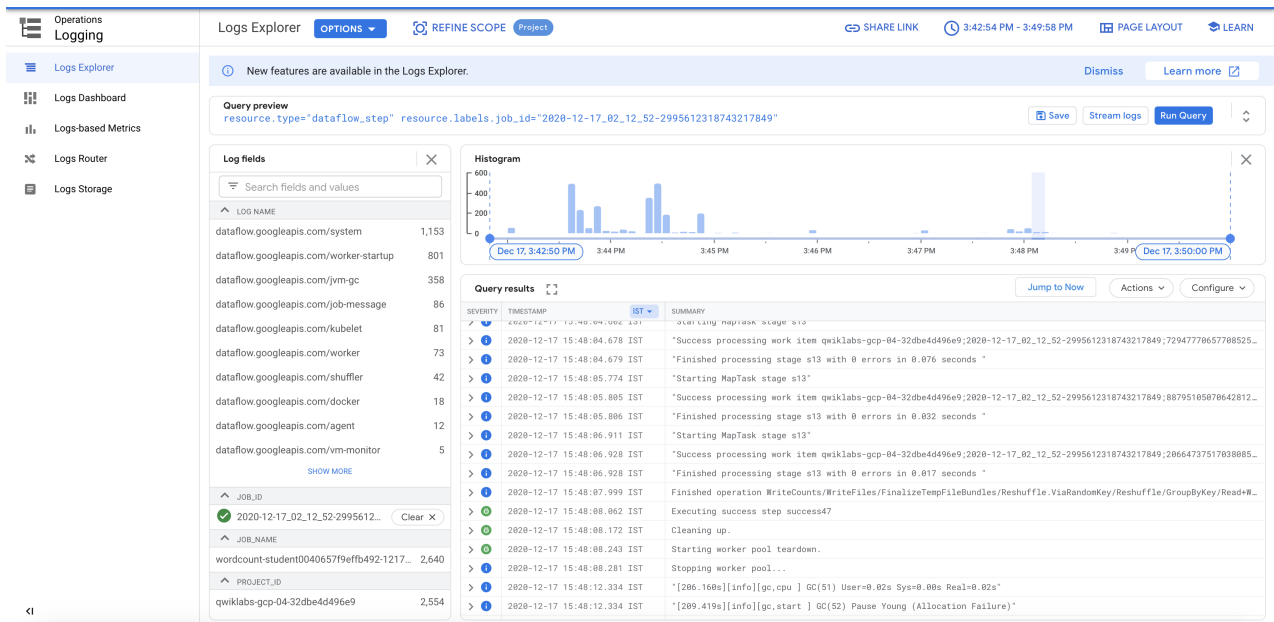
Filter the messages that appear in the Job Log tab by using the Minimum Severity drop-down menu.

The screenshot displays the Databricks Logs Explorer with the **JOB LOGS** tab selected. The interface shows a search bar with the text "Showing 30 messages" and a filter menu set to "Info". Below the search bar, there is a message: "No older entries found matching current filter." The log messages are listed in a table with columns for timestamp, log level, and message content. The messages show the execution of a WordCount pipeline, including worker configuration, data shuffling, and word counting operations.

Timestamp	Log Level	Message Content
2020-12-17T10:13:00.929462874Z	Info	Worker configuration: n1-standard-1 in us-central1-b.
2020-12-17T10:13:03.632615354Z	Info	Executing operation WriteCounts/WriteFiles/FinalizeTempFileBundles/Reshuffle.ViaRandomKey/Reshuffle/GroupByKey/C...
2020-12-17T10:13:03.664655017Z	Info	Executing operation WordCount.CountWords/Count.PerElement/Combine.perKey(Count)/GroupByKey/Create
2020-12-17T10:13:03.680445638Z	Info	Executing operation WriteCounts/WriteFiles/WriteUnshardedBundlesToTempFiles/GroupUnwritten/Create
2020-12-17T10:13:03.712885683Z	Info	Starting 1 workers in us-central1-b...
2020-12-17T10:13:03.767198383Z	Info	Finished operation WriteCounts/WriteFiles/FinalizeTempFileBundles/Reshuffle.ViaRandomKey/Reshuffle/GroupByKey/Cr...
2020-12-17T10:13:03.781226390Z	Info	Finished operation WordCount.CountWords/Count.PerElement/Combine.perKey(Count)/GroupByKey/Create
2020-12-17T10:13:03.781226455Z	Info	Finished operation WriteCounts/WriteFiles/WriteUnshardedBundlesToTempFiles/GroupUnwritten/Create
2020-12-17T10:13:03.997305144Z	Info	Executing operation ReadLines/Read+WordCount.CountWords/ParDo(ExtractWords)+WordCount.CountWords/Count.PerElemen...
2020-12-17T10:14:43.894665629Z	Info	Finished operation ReadLines/Read+WordCount.CountWords/ParDo(ExtractWords)+WordCount.CountWords/Count.PerElemen...
2020-12-17T10:14:43.978815533Z	Info	Executing operation WordCount.CountWords/Count.PerElement/Combine.perKey(Count)/GroupByKey/Close
2020-12-17T10:14:44.127353196Z	Info	Finished operation WordCount.CountWords/Count.PerElement/Combine.perKey(Count)/GroupByKey/Close
2020-12-17T10:14:44.232514318Z	Info	Executing operation WordCount.CountWords/Count.PerElement/Combine.perKey(Count)/GroupByKey/Read+WordCount.CountWo...
2020-12-17T10:17:54.331524451Z	Info	Finished operation WordCount.CountWords/Count.PerElement/Combine.perKey(Count)/GroupByKey/Read+WordCount.CountWo...

Click the **WORKER LOGS** tab to see the worker logs for the Compute Engine instances that run your pipeline. Worker Logs consist of log lines generated by your code and the Dataflow generated code running it.

If you are trying to debug a failure in the pipeline, Click on **View in Logs Explorer** icon(  ). Oftentimes there will be additional logging in the Worker Logs to help solve the problem. These logs are aggregated across all workers and can be filtered and searched.



Now we'll check that your job succeeded.

## Check that your job succeeded

In the console, click **Navigation menu** > **Dataflow** to open the **Dataflow** page and monitor your job.

You'll see your wordcount job with a status of **Running** at first, and then **Succeeded**:

Cloud Dataflow		Jobs				
Name	Type	Start time	Elapsed time	Status	ID	
✔ wordcount	Batch	Mar 10, 2016, 1:55:47 PM	12 min 41 sec	Succeeded	2016-03-10_13_55_47-8540823045578098352	

The job will take approximately 3-4 minutes to run.

## Test Completed Task

Click **Check my progress** to verify your performed task. If you have successfully run a text processing pipeline on Cloud Dataflow, you will see an assessment score.

Run a text processing pipeline on Cloud Dataflow

Remember when you ran the pipeline and specified an output bucket? Let's take a look at the result (because don't you want to see how many times each word in *King Lear* occurred?!).

In the Cloud Console, click **Navigation menu** > **Cloud Storage**, then **Browser**. In your bucket you should see the output text files and staging folder that your job created:

Storage

Browser

Monitoring

Settings

←

Bucket details

REFRESH

LEARN

qwiklabs-gcp-04-32dbe4d496e9

OBJECTS

CONFIGURATION

PERMISSIONS

RETENTION

LIFECYCLE

Buckets

>

qwiklabs-gcp-04-32dbe4d496e9

UPLOAD FILES

UPLOAD FOLDER

CREATE FOLDER

MANAGE HOLDS

DOWNLOAD

DELETE

Filter

Filter by object or folder name prefix

<input type="checkbox"/>	Name	Size	Type	Created time ⓘ	Storage class	Last modified	Public access ⓘ	Encryption ⓘ	Retention expiration date ⓘ	Holds ⓘ	
<input type="checkbox"/>	output-00000-of-00003	15 KB	text/plain	Dec 17, 2020, 3:48:01 PM	Standard	Dec 17, 2020, 3:48:01 PM	Not public	Google-managed key	—	None	<div><div>↓</div><div>↑</div><div>⌵</div></div>
<input type="checkbox"/>	output-00001-of-00003	15.3 KB	text/plain	Dec 17, 2020, 3:48:01 PM	Standard	Dec 17, 2020, 3:48:01 PM	Not public	Google-managed key	—	None	<div><div>↓</div><div>↑</div><div>⌵</div></div>
<input type="checkbox"/>	output-00002-of-00003	14.9 KB	text/plain	Dec 17, 2020, 3:48:01 PM	Standard	Dec 17, 2020, 3:48:01 PM	Not public	Google-managed key	—	None	<div><div>↓</div><div>↑</div><div>⌵</div></div>
<input type="checkbox"/>	staging/	—	Folder	—	—	—	—	—	—	—	<div><div>↓</div><div>↑</div><div>⌵</div></div>

Open a text file to see some results.

## Test your Understanding

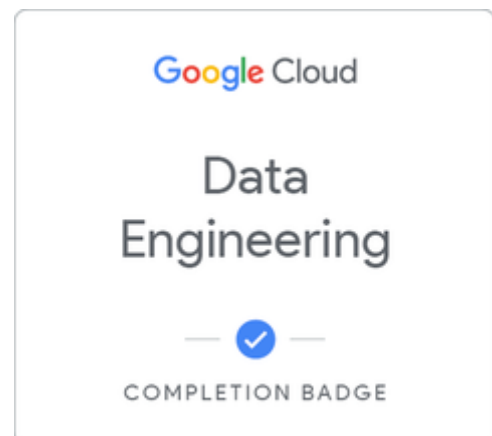
Below are a multiple choice questions to reinforce your understanding of this lab's concepts. Answer them to the best of your abilities.

## Congratulations!

You learned how to create a Maven project with the Cloud Dataflow SDK, run an example pipeline using the Cloud Console, and delete the associated Cloud Storage bucket and its contents.

## Finish Your Quest

Continue your Quest with [Data Engineering](#). A Quest is a series of related labs that form a learning path. Completing this Quest earns you the badge above, to recognize your achievement. You can make your badge (or badges) public and link to them in your online resume or social media account. [Enroll in this Quest](#) and get immediate completion credit if you've taken this lab. [See other available Qwiklabs Quests](#).



## Take Your Next Lab

Continue your Quest with [Dataproc: Qwik Start - Console](#) or try one of these:

- [Building an IoT Analytics Pipeline on Google Cloud](#)
- [Working with Cloud Dataprep on Google Cloud](#)

## Learn More

Dataflow Documentation: <https://cloud.google.com/dataflow/docs/>



## Google Cloud Training & Certification

---

...helps you make the most of Google Cloud technologies. Our classes include technical skills and best practices to help you get up to speed quickly and continue your learning journey. We offer fundamental to advanced level training, with on-demand, live, and virtual options to suit your busy schedule. Certifications help you validate and prove your skill and expertise in Google Cloud technologies.

Manual Last Updated June 22, 2022

Lab Last Tested June 22, 2022

Copyright 2022 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.