

# A simple Dataflow pipeline (Python)

2 hoursFree

[Rate Lab](#)

## Overview

In this lab, you will open a Dataflow project, use pipeline filtering, and execute the pipeline locally and on the cloud.

- Open Dataflow project
- Pipeline filtering
- Execute the pipeline locally and on the cloud

# Objective

In this lab, you learn how to write a simple Dataflow pipeline and run it both locally and on the cloud.

- Setup a Python Dataflow project using Apache Beam
- Write a simple pipeline in Python
- Execute the query on the local machine
- Execute the query on the cloud

# Setup

For each lab, you get a new Google Cloud project and set of resources for a fixed time at no cost.

1. Make sure you signed into Qwiklabs using an **incognito window**.

2. Note the lab's access time (for example, **02:00:00** and make sure you can finish in that time block.

There is no pause feature. You can restart if needed, but you have to start at the beginning.

START LAB

3. When ready, click .
4. Note your lab credentials. You will use them to sign in to the Google Cloud Console.

Open Google Console

**Caution:** When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. [Learn more.](#)

Username

google2876526\_student@qwiklabs.n



Password

TG959yrKDX



GCP Project ID

qwiklabs-gcp-0855e773352d3560



[New to labs? View our introductory video!](#)

5. Click **Open Google Console**.
6. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

If you use other credentials, you'll get errors or **incur charges**.

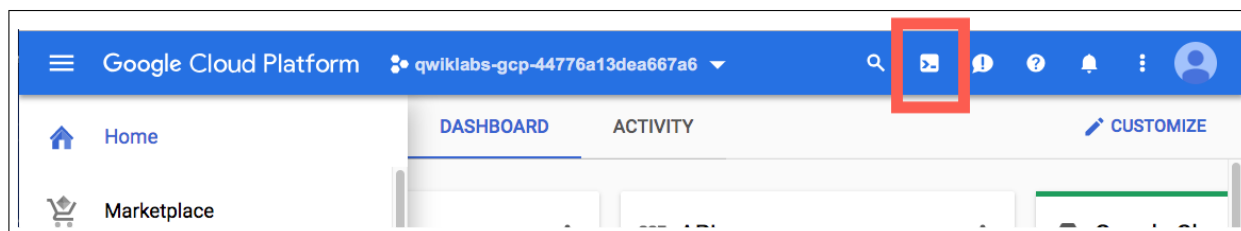
7. Accept the terms and skip the recovery resource page.

Do not click **End Lab** unless you are finished with the lab or want to restart it. This clears your work and removes the project.

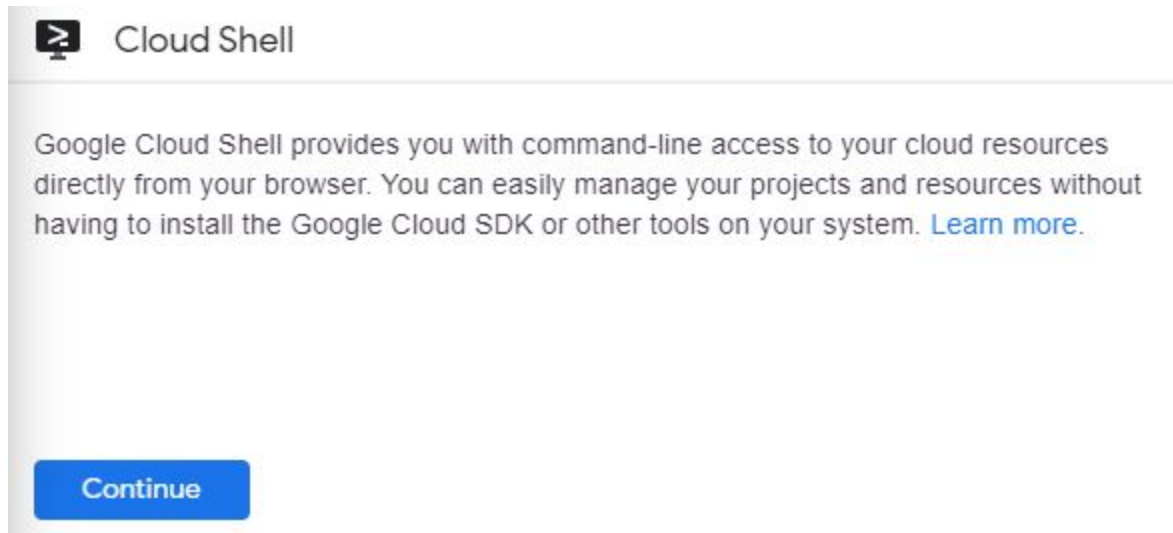
## Activate Cloud Shell

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.

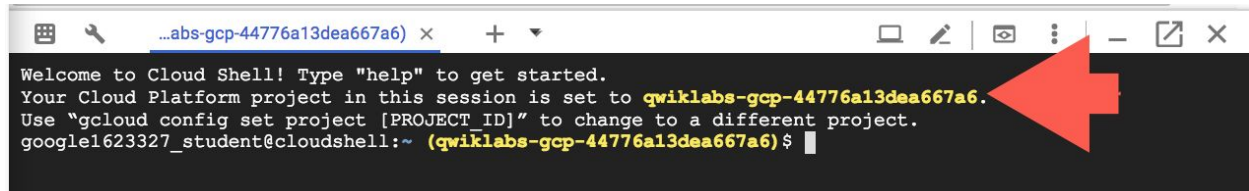
In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.



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*. For example:

A screenshot of a terminal window with a dark background. The terminal text reads: "Welcome to Cloud Shell! Type 'help' to get started. Your Cloud Platform project in this session is set to qwiklabs-gcp-44776a13dea667a6. Use 'gcloud config set project [PROJECT\_ID]' to change to a different project. google1623327\_student@cloudshell:~ (quiklabs-gcp-44776a13dea667a6) \$". A large red arrow points from the right side of the terminal towards the project ID "quiklabs-gcp-44776a13dea667a6". The browser tab at the top is labeled "...abs-gcp-44776a13dea667a6)".

`gcloud` is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

You can list the active account name with this command:

```
gcloud auth list
```

(Output)

```
Credentialed accounts:
- <myaccount>@<mydomain>.com (active)
```

(Example output)

```
Credentialed accounts:  
- google1623327 student@qwiklabs.net
```

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` see the [gcloud command-line tool overview](#).

## Launch Google Cloud Shell Code Editor

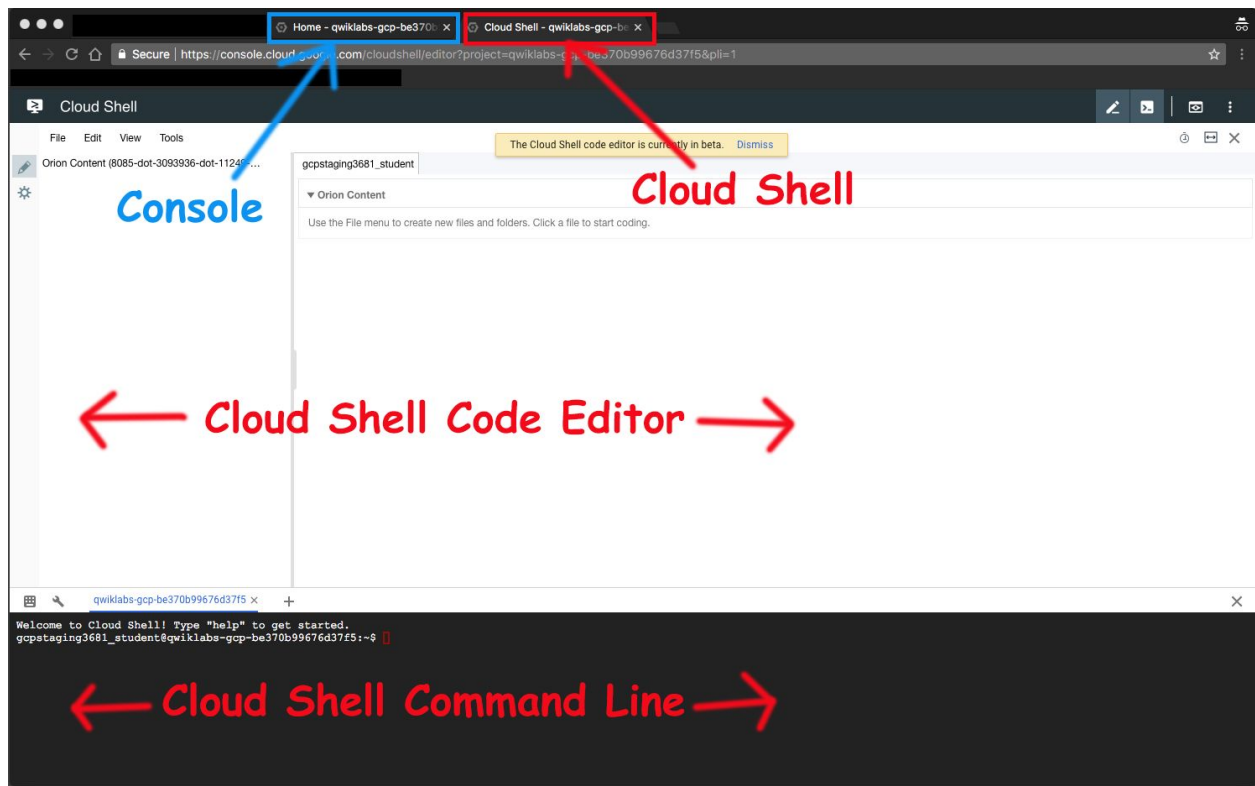
Use the Google Cloud Shell Code Editor to easily create and edit directories and files in the Cloud Shell instance.

Once you activate the Google Cloud Shell, click the **Open editor** button to open the Cloud Shell Code Editor.



You now have three interfaces available:

- The Cloud Shell Code Editor
- Console (By clicking on the tab). You can switch back and forth between the Console and Cloud Shell by clicking on the tab.
- The Cloud Shell Command Line (By clicking on **Open Terminal** in the Console)



# Task 1. Preparation

For this lab, you will need the training-data-analyst files and a Cloud Storage bucket.

## Verify that the repository files are in Cloud Shell Editor

1. Clone the repository from the Cloud Shell command line:

```
git clone https://github.com/GoogleCloudPlatform/training-data-analyst
```


2. Click on **File > Refresh** in the left navigator panel. You should see the **training-data-analyst** directory.

## Verify that you have a Cloud Storage bucket

If you don't have a bucket, you can follow these instructions to create a bucket.

3. In the Console, on the **Navigation menu** () , click **Home**.



4. **Select and copy** the Project ID. For simplicity you will use the Qwiklabs Project ID, which is already globally unique, as the bucket name.
5. In the Console, on the **Navigation menu** () , click **Storage > Browser**.
6. Click **Create Bucket**.
7. Specify the following, and leave the remaining settings as their defaults:

Property	Value (type value or select option as specified)
<b>Name</b>	<your unique bucket name (Project ID)>
<b>Default storage class</b>	Multi-Regional
<b>Location</b>	<Your location>

8.  
Click **Create**.
9. Record the name of your bucket. You will need it in subsequent tasks.
10. In Cloud Shell enter the following to create an environment variable named "BUCKET" and verify that it exists with the echo command.

```
BUCKET="<your unique bucket name (Project ID)>"  
echo $BUCKET
```

You can use `$BUCKET` in Cloud Shell commands. And if you need to enter the bucket name `<your-bucket>` in a text field in Console, you can quickly retrieve the name with `echo $BUCKET`.

## Verify that Dataflow API is enabled for this project

11. Return to the browser tab for Console. In the top search bar, enter **Dataflow API**. This will take you to the page, **Navigation menu > APIs & Services > Dashboard > Dataflow API**. It will either show a status information or it will give you the option to **Enable** the API.
12. If necessary, **Enable** the API.

## Task 2. Open Dataflow project

The goal of this lab is to become familiar with the structure of a Dataflow project and learn how to execute a Dataflow pipeline. You will need to update some files to install Apache Beam. Apache Beam is an open source platform for executing data processing workflows.

1. Return to the browser tab containing Cloud Shell. In Cloud Shell navigate to the directory for this lab:

```
cd ~/training-data-analyst/courses/data_analysis/lab2/python
```

2. Install the necessary dependencies for Python dataflow:

```
sudo ./install_packages.sh
```

3. Verify that you have the right version of pip. (It should be > 8.0):

```
pip3 -V
```

If not, open a new Cloud Shell tab and it should pick up the updated version of pip.

4. Use **File > Refresh** in Cloud Shell editor to view the local copy of the repository.

If at any time during the DataFlow labs you are logged out of Cloud Shell due to inactivity, when you login the in-memory elements of Apache Beam will be lost. So you will need to reissue these commands before proceeding:

```
cd ~/training-data-analyst/courses/data_analysis/lab2/python
```

```
sudo ./install_packages.sh
```

## Task 3. Pipeline filtering

1. In the Cloud Shell code editor navigate to the directory

`/training-data-analyst/courses/data_analysis/lab2/python` and view the file `grep.py`. **Do not make any changes to the code.**

Alternatively, you could view the file with nano. **Do not make any changes to the code.** If you use nano, press Ctrl + X to exit.

```
cd ~/training-data-analyst/courses/data_analysis/lab2/python
nano grep.py
```

Can you answer these questions about the file `grep.py`?

- What files are being read?
- What is the search term?
- Where does the output go?

There are three transforms in the pipeline:

- What does the transform do?
- What does the second transform do?
- Where does its input come from?
- What does it do with this input?
- What does it write to its output?
- Where does the output go to?
- What does the third transform do?

## Task 4. Execute the pipeline locally

1. In the Cloud Shell command line, locally execute `grep.py`.

```
cd ~/training-data-analyst/courses/data_analysis/lab2/python
python3 grep.py
```

Note: if you see an error that says "No handlers could be found for logger "oauth2client.contrib.multistore\_file", you may ignore it. The error is simply saying that logging from the oauth2 library will go to stderr.

2. The output file will be `output.txt`. If the output is large enough, it will be sharded into separate parts with names like: `output-00000-of-00001`. If necessary, you can locate the correct file by examining the file's time.

```
ls -al /tmp
```

3. Examine the output file. Replace `"-"` below with the appropriate suffix.

```
cat /tmp/output-*
```

Does the output seem logical?

## Task 5. Execute the pipeline on the cloud

1. Copy some Java files to the cloud.

```
gsutil cp
../javahelp/src/main/java/com/google/cloud/training/dataanalyst/javahelp/*.java
gs://$BUCKET/javahelp
```

2. Edit the Dataflow pipeline in `grepc.py`. In the Cloud Shell code editor  
navigate to the directory

```
/training-data-analyst/courses/data_analysis/lab2/python in and
edit the file grepc.py.
```

3. Replace PROJECT and BUCKET with your Project ID and Bucket name.

Here are easy ways to retrieve the values:

```
echo $DEVSHHELL PROJECT ID
echo $BUCKET
```

Example strings before:

```
PROJECT='cloud-training-demos'
BUCKET='cloud-training-demos'
```


Example strings after edit (use your values):

```
PROJECT='qwiklabs-gcp-your-value'
BUCKET='qwiklabs-gcp-your-value'
```

4. Submit the Dataflow job to the cloud:

```
python3 grepc.py
```


Because this is such a small job, running on the cloud will take significantly longer than running it locally (on the order of 2-3 minutes).

5. Return to the browser tab for Console. On the **Navigation menu** () , click **Dataflow** and click on your job to monitor progress.

Example:



#### Job summary

Job name	examplejob2
Job ID	2018-02-06_12_47_44-6148155460441137914
Region <sup>?</sup>	us-central1
Job status	 Succeeded
SDK version	Google Cloud Dataflow SDK for Python 2.2.0
Job type	Batch
Start time	Feb 6, 2018, 3:47:45 PM
Elapsed time	4 min 58 sec

#### Autoscaling


Workers	0
Current state	Stopping worker pool.

Feb 6, 2018 3:47 PM

6. Wait for the job status to turn to **Succeeded**. At this point, your Cloud Shell will display a command-line prompt.

7. Examine the output in the Cloud Storage bucket. On the **Navigation menu**



() , click **Storage > Browser** and click on your bucket. Click the **javahelp** directory. This job will generate the file `output.txt`. If the file is large enough it will be sharded into multiple parts with names like: `output-0000x-of-000y`. You can identify the most recent file by name or by the **Last modified** field. Click on the file to view it.

Alternatively, you could download the file in Cloud Shell and view it:

```
gsutil cp gs://$BUCKET/javahelp/output* .  
cat output*
```

## End your lab

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you've used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:



- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.

Manual Last Updated: May 16, 2020

Lab Last Tested: May 16, 2020

©2020 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.