

Serverless Data Analysis with Dataflow: A Simple Dataflow Pipeline (Python) | Qwiklabs

Tuesday, November 17, 2020 12:24 PM

Clipped from:

https://googlecourses.qwiklabs.com/course_sessions/71782/labs/11644

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 GCP 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.

3. When ready, click

START LAB

4. Note your lab credentials. You will use them to sign in to Cloud Platform Console.

[Open Google Console](#)

Caution: When you are in the console, do not deviate

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

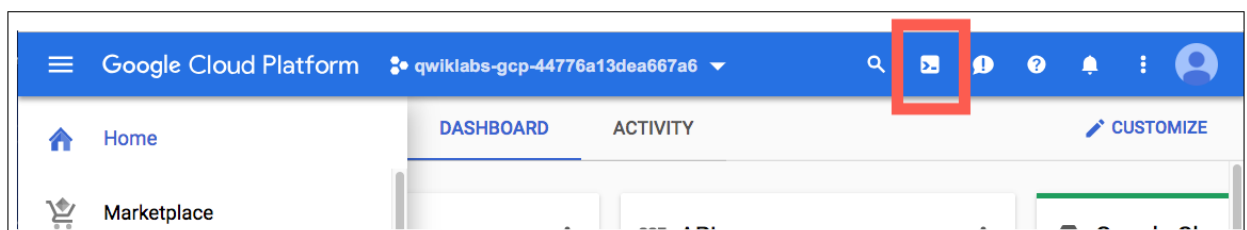
[New to labs? View our introductory video!](#)

5. Click **Open Google Console.**
6. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.
1. Accept the terms and skip the recovery resource page.

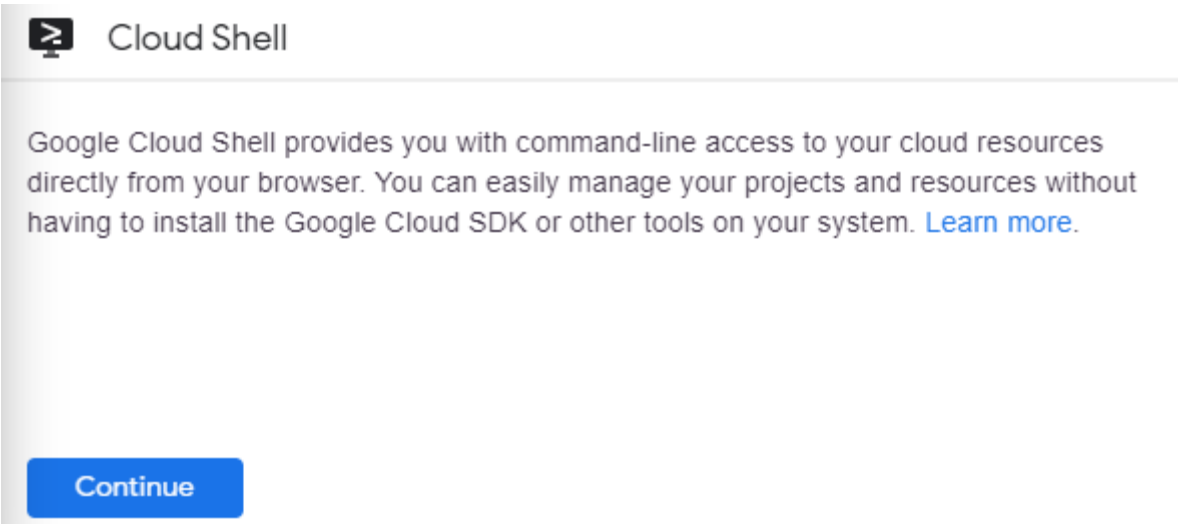
[Activate Google Cloud Shell](#)

Google 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. Google Cloud Shell provides command-line access to your GCP resources.

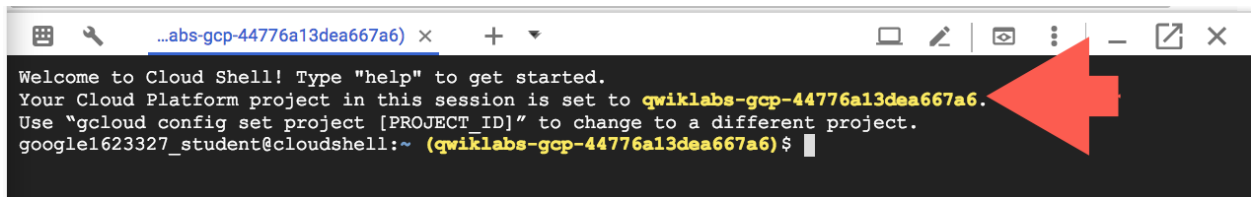
1. In GCP console, on the top right toolbar, click the Open Cloud Shell button.



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*. For example:



```
...abs-gcp-44776a13dea667a6) x + v
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:~ (qwiklabs-gcp-44776a13dea667a6) $
```

gcloud is the command-line tool for Google Cloud Platform. 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 Full
documentation of gcloud is available on Google Cloud gcloud 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.

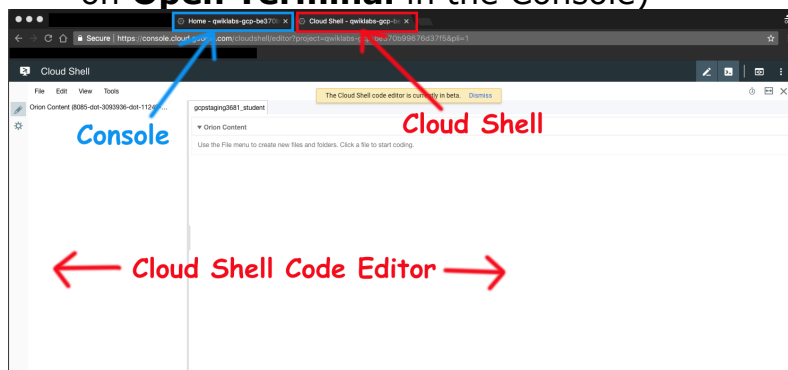
Once you activate the Google Cloud Shell, click the **Open editor** button to open the Cloud Shell

Code Editor.




You now have three interfaces available:

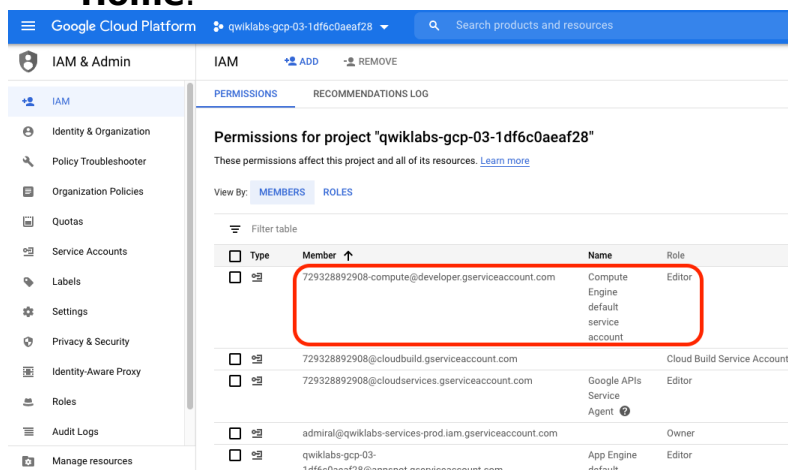
- The Cloud Shell Code Editor
- Console (By clicking on the tab). You can switch back and forth between the Console and Cloud Shell by clicking on the tab.
- The Cloud Shell Command Line (By clicking on **Open Terminal** in the Console)



Check project permissions

Before you begin your work on Google Cloud, you need to ensure that your project has the correct permissions within Identity and Access Management (IAM).

1. In the Google Cloud console, on the **Navigation menu** (), click **IAM & Admin** > **IAM**.
2. Confirm that the default compute Service Account 729328892908-compute@developer.gserviceaccount.com is present and has the editor role assigned. The account prefix is the project number, which you can find on **Navigation menu** > **Home**.



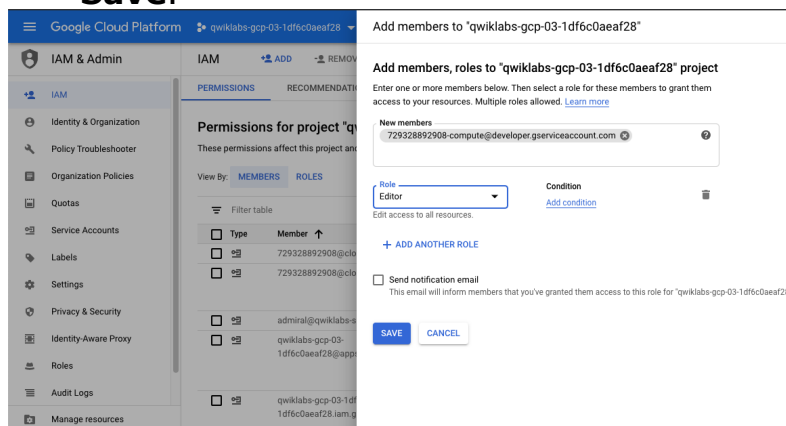
If the account is not present in IAM or does not have the editor role, follow the steps below to assign the required role.

- In the Google Cloud console, on the **Navigation menu**, click **Home**.
- Copy the project number (e.g. 729328892908).
- On the **Navigation menu**, click **IAM & Admin > IAM**.
- At the top of the **IAM** page, click **Add**.
- For **New members**, type:

[{project-number}-compute@developer.gserviceaccount.com](#)

Replace {project-number} with your project number.

- For **Role**, select **Project > Editor**. Click **Save**.



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>


1. Click on the **Refresh** icon 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.

1. In the Console, on the **Navigation menu** (), click **Home**.
2. **Select and copy** the Project ID. For simplicity you will use the Qwiklabs Project

ID, which is already globally unique, as the bucket name.

3. In the Console, on the **Navigation menu** () click **Storage > Browser**.
4. Click **Create Bucket**.
5. Specify the following, and leave the remaining settings as their defaults:

Property Value (type value or select option as specified)

Name	<your unique bucket name (Project ID)>
Default storage class	Multi-Regional
Location	<Your location>

1. Click **Create**.
2. Record the name of your bucket. You will need it in subsequent tasks.
3. 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](#)

1. 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.
2. 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
```

1. Install the necessary dependencies for Python dataflow:

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

1. 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.

1. Use **Refresh** icon in Cloud Shell editor to view the local copy of the repository.

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.**

```
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
```

1. 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
```

1. 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/t
raining/dataanalyst/javahelp/*.java
gs://$BUCKET/javahelp
```

Click *Check my progress* to verify the objective.
Copy Java files to the Cloud

1. 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`.
2. 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'
```

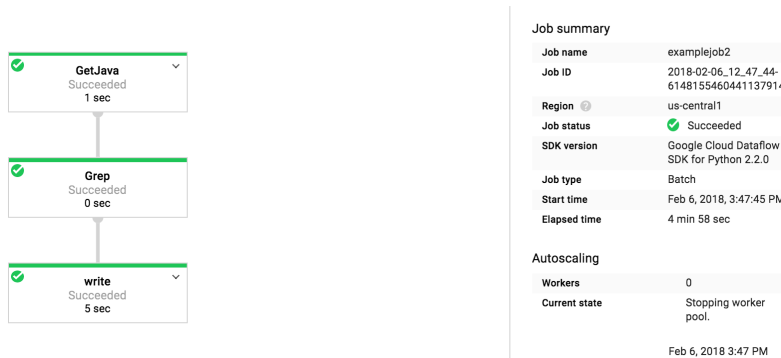
1. Submit the Dataflow job to the cloud:

```
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).

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

Example:



The screenshot shows the Google Cloud Dataflow console. On the left, a DAG (Directed Acyclic Graph) is displayed with three steps: 'GetJava' (Succeeded, 1 sec), 'Grep' (Succeeded, 0 sec), and 'write' (Succeeded, 5 sec). On the right, the 'Job summary' section is visible, showing details for job 'examplejob2'.

Job summary	
Job name	examplejob2
Job ID	2018-02-06_12_47_44-6148155460441137914
Region	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

Autoscaling	
Workers	0
Current state	Stopping worker pool.

Feb 6, 2018 3:47 PM

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

Click *Check my progress* to verify the objective.
Submit the Dataflow job to the Cloud

1. 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.