A Simple Dataflow Pipeline (Python) 2.5

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.

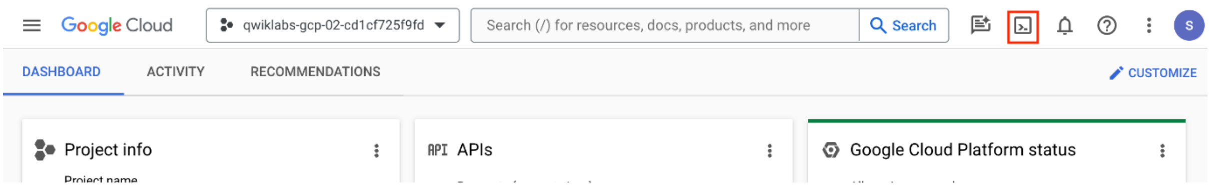
- Setup a Python Dataflow project using Apache Beam
- Write a simple pipeline in Python
- Execute the query on the local machine
- Execute the query on the cloud

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.

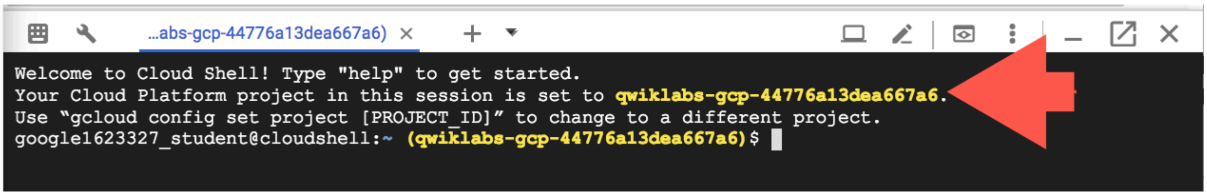
Google Cloud Shell provides command-line access to your Google Cloud resources.

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


Google Cloud
qwiklabs-gcp-02-cd1cf725f9fd
Search (/) for resources, docs, products, and more
Search
S
DASHBOARD
ACTIVITY
RECOMMENDATIONS
CUSTOMIZE
Project info
Project name
API APIs
Google Cloud Platform status

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:



```
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. 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)

</mydomain></myaccount>
```

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>

</project_id>
```

Example output:

```
[core]

project = qwiklabs-gcp-44776a13dea667a6
```

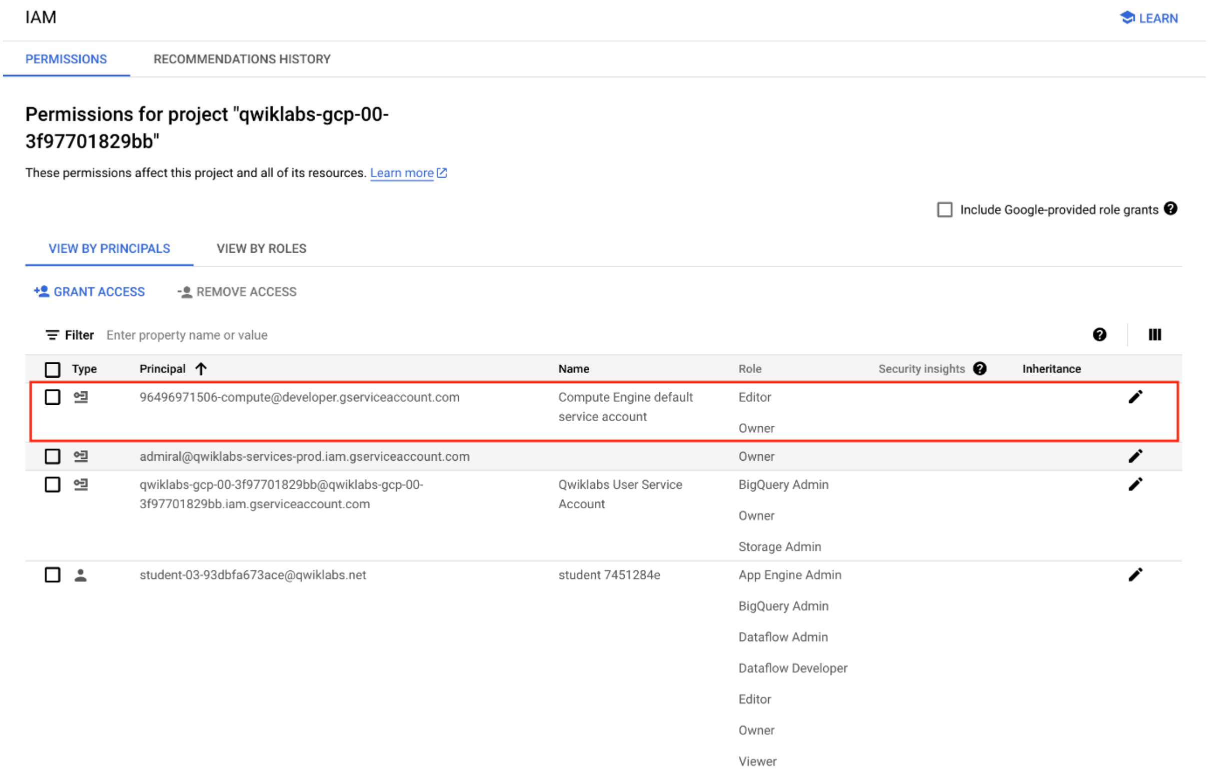
Note: Full documentation of **gcloud** is available in the gcloud CLI overview guide .

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** (☰), select **IAM & Admin** > **IAM**.

2. Confirm that the default compute Service Account {project-number}-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 > Cloud Overview > Dashboard**.



IAM

LEARN

PERMISSIONS | RECOMMENDATIONS HISTORY

Permissions for project "qwiklabs-gcp-00-3f97701829bb"

These permissions affect this project and all of its resources. Learn more

□ Include Google-provided role grants

VIEW BY PRINCIPALS | VIEW BY ROLES

GRANT ACCESS | REMOVE ACCESS

Filter Enter property name or value

□	Type	Principal ↑	Name	Role	Security insights	Inheritance
□		96496971506-compute@developer.gserviceaccount.com	Compute Engine default service account	Editor Owner		
□		admiral@qwiklabs-services-prod.iam.gserviceaccount.com		Owner		
□		qwiklabs-gcp-00-3f97701829bb@qwiklabs-gcp-00-3f97701829bb.iam.gserviceaccount.com	Qwiklabs User Service Account	BigQuery Admin Owner Storage Admin		
□		student-03-93dbfa673ace@qwiklabs.net	student 7451284e	App Engine Admin BigQuery Admin Dataflow Admin Dataflow Developer Editor Owner Viewer		

Task 1. Ensure that the Dataflow API is successfully enabled

- Execute the following block of code in the Cloud Shell:

```
gcloud services disable dataflow.googleapis.com --force

gcloud services enable dataflow.googleapis.com
```

Task 2. Preparation

Open the SSH terminal and connect to the training VM

You will be running all code from a curated training VM.

1. In the console, on the **Navigation menu** (☰), click **Compute Engine** > **VM instances**.
2. Locate the line with the instance called **training-vm**.
3. On the far right, under **Connect**, click on **SSH** to open a terminal window.
4. In this lab, you will enter CLI commands on the **training-vm**.

Download code repository

- Download a code repository to use in this lab. In the **training-vm** SSH terminal enter the following:

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

Create a Cloud Storage bucket

Follow these instructions to create a bucket.

1. In the Console, on the **Navigation menu**, click **Cloud Storage** > **Buckets**.
2. Click **+ Create**.
3. Specify the following, and leave the remaining settings as their defaults:

Property	**Value (type value or select option as specified)**
Name	qwiklabs-gcp-03-7cfc4afd6af5
Location type	Multi-region

4. Click **Create**.
5. If you get the Public access will be prevented prompt, select Enforce public access prevention on this bucket and click **Confirm**.

Record the name of your bucket. You will need it in subsequent tasks.

6. In the **training-vm** SSH terminal enter the following to create an environment variable named "BUCKET" and verify that it exists with the echo command:

```
BUCKET="qwiklabs-gcp-03-7cfc4afd6af5"

echo $BUCKET
```

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

Task 3. Pipeline filtering

The goal of this lab is to become familiar with the structure of a Dataflow project and learn how to execute a Dataflow pipeline.

1. Return to the **training-vm** SSH terminal and navigate to the directory /training-data-analyst/courses/data_analysis/lab2/python and view the file grep.py.
2. 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
```

3. Press CTRL+X to exit Nano.

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

- What files are being read? – java files in "../javahelp/src/main/java/com/google/cloud/training/dataanalyst/javahelp/"
- What is the search term? – "import"
- Where does the output go? "/tmp/"

There are three transforms in the pipeline:

- What does the transform do? – read the java files
- What does the second transform do? – run the grep command to find the search term
- Where does its input come from? – from the input location of the java files
- What does it do with this input? – it read the files line by line to find if the 'import' search term is present
- What does it write to its output? – to /tmp/
- Where does the output go? – to /tmp/
- What does the third transform do? – write the lines with the search term to the output location

Task 4. Execute the pipeline locally

1. In the **training-vm** SSH terminal, locally execute grep.py:

```
python3 grep.py
```

Note: Ignore the warning if any.

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.

2. Locate the correct file by examining the file's time:

```
ls -al /tmp
```

3. Examine the output file(s).
4. You can 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. In the **training-vm** SSH terminal, enter the following command:

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

2. Using Nano, edit the Dataflow pipeline in grepc.py:

```
nano grepc.py
```

3. Replace PROJECT, BUCKET, and REGION with the values listed below. Please retain the outside single quotes.

```
PROJECT='qwiklabs-gcp-03-7cfc4afd6af5'
```

```
BUCKET='qwiklabs-gcp-03-7cfc4afd6af5'
```

```
REGION='us-west1'
```

Save the file and close Nano by pressing the CTRL+X key, then type Y, and press Enter.

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 7-10 minutes).

5. Return to the browser tab for the console.
6. On the **Navigation menu**, click **VIEW ALL PRODUCTS**. In the **Analytics** section, click **Dataflow** and click on your job to monitor progress.
7. Wait for the **Job status** to be **Succeeded**.
8. Examine the output in the Cloud Storage bucket.
9. On the **Navigation menu**, click **Cloud Storage > Buckets** and click on your bucket.
10. Click the **javahelp** directory.

This job generates 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.

11. Click on the file to view it.

Alternatively, you can download the file via the **training-vm** SSH terminal and view it:

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

End your lab