

# ETL Processing on Google Cloud Using Dataflow and BigQuery

GSP290



# Overview

In this lab you build several Data Pipelines that ingest data from a publicly available dataset into BigQuery, using these Google Cloud services:

- **Cloud Storage**
- **Dataflow**
- **BigQuery**

You will create your own Data Pipeline, including the design considerations, as well as implementation details, to ensure that your prototype meets the requirements. Be sure to open the python files and read the comments when instructed to.

## Setup

### 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 Qwiklabs 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.

### What you need

To complete this lab, you need:

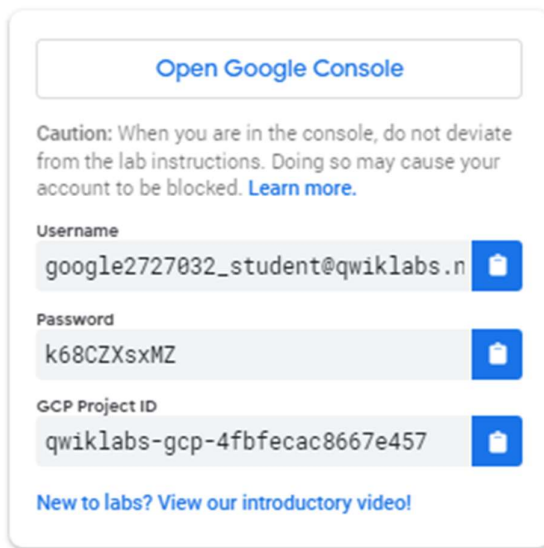
- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.

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

**Note:** If you are using a Pixelbook, open an Incognito window to run this lab.


### 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 a panel populated with the temporary credentials that you must use for this lab.




[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.](#)

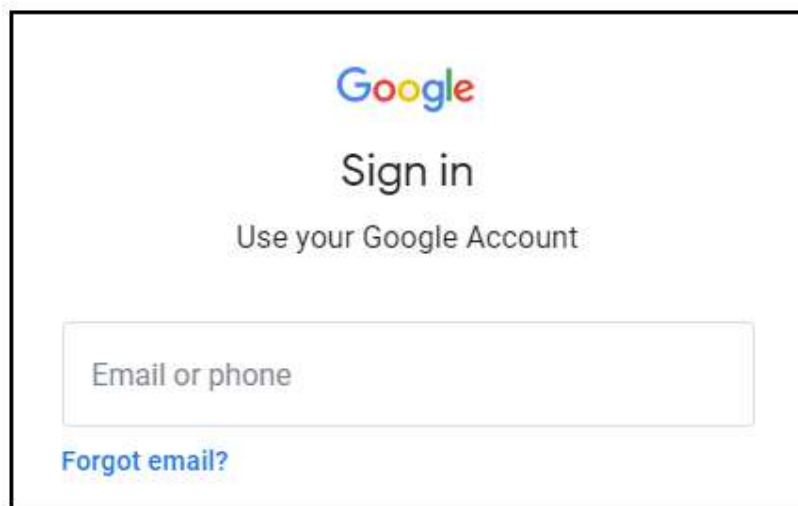
Username  
google2727032\_student@qwiklabs.n 

Password  
k68CZXsxMZ 

GCP Project ID  
qwiklabs-gcp-4fbfecac8667e457 

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

2. Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.



Google

Sign in

Use your Google Account

Email or phone

[Forgot email?](#)

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

If you see the **Choose an account** page, click **Use Another**



**Account.**

3. In the **Sign in** page, paste the username that you copied from the Connection Details panel. Then copy and paste the password.

**Important:** You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

4. 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.

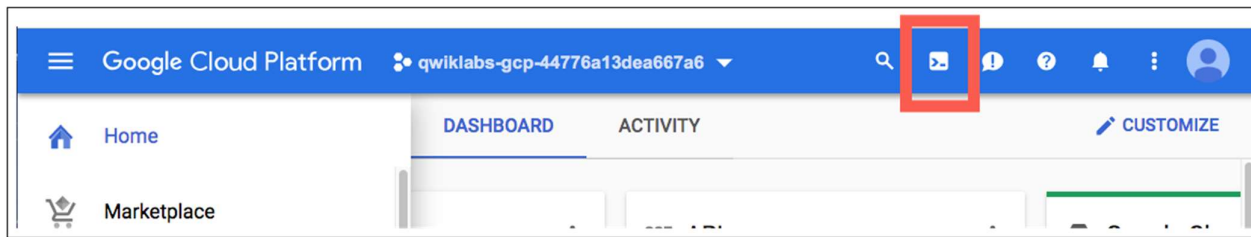
**Note:** You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-left.



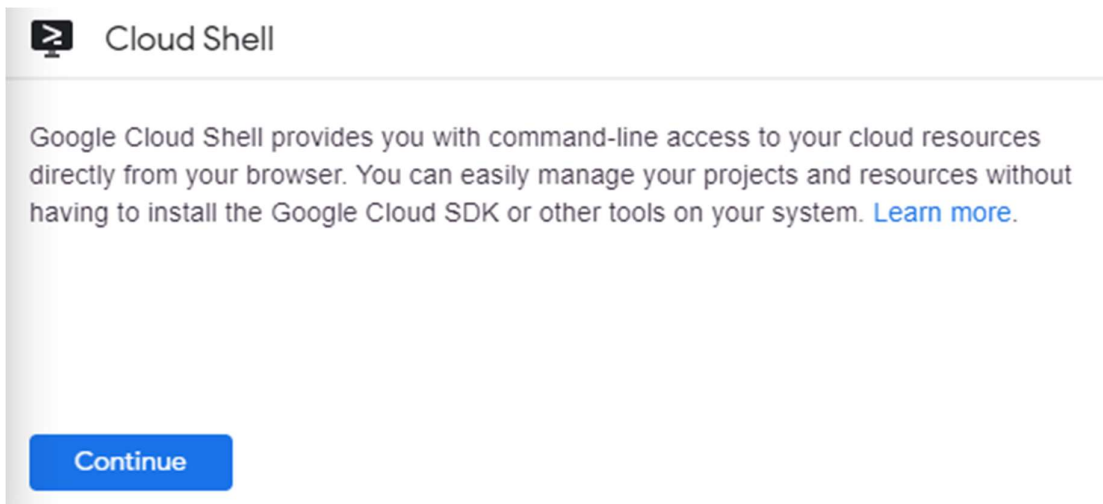
## Activate Cloud Shell

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.

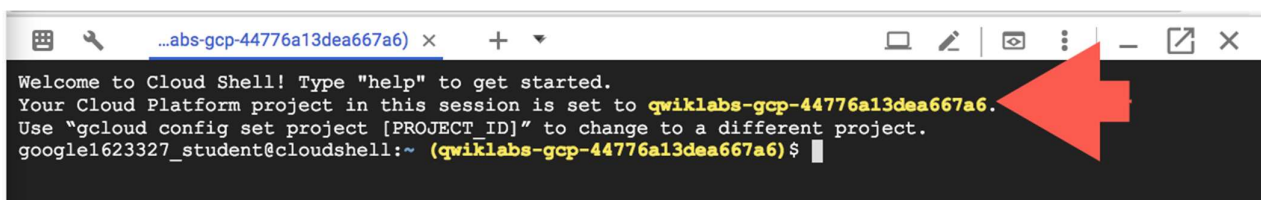
In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.



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:



`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@gwiklabs.net
```

You can list the project ID with this command:

```
gcloud config list project
```

(Output)

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


(Example output)

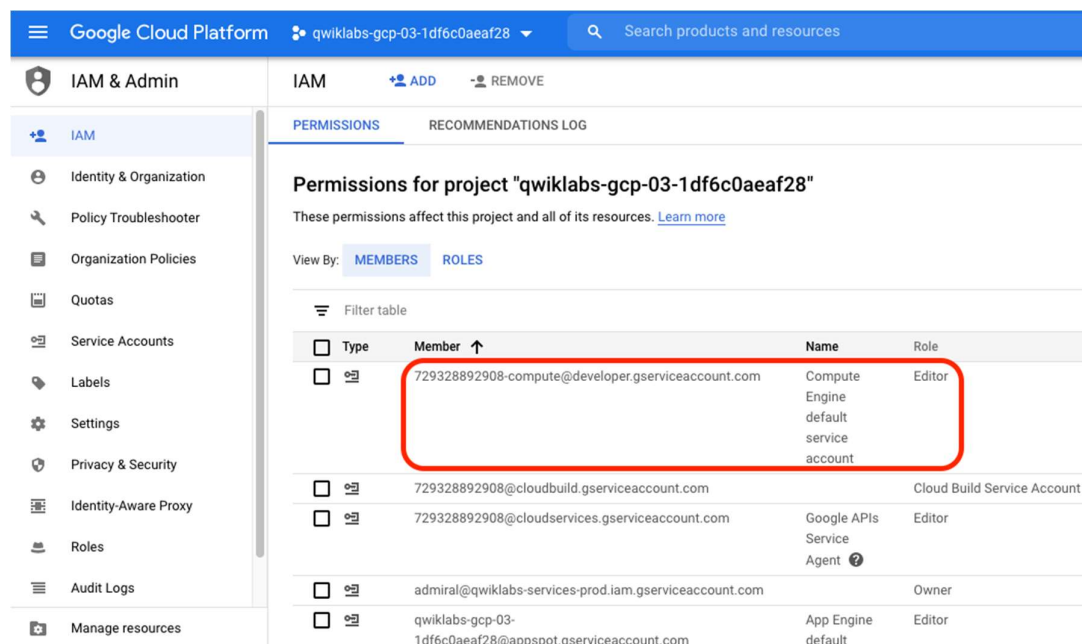
```
[core]  
project = qwiklabs-gcp-44776a13dea667a6
```

For full documentation of `gcloud` see the [gcloud command-line tool overview](#).

## 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 `{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 > Home**.



The screenshot shows the Google Cloud Platform console with the IAM & Admin section selected. The left sidebar lists various IAM-related options, and the main content area displays the 'Permissions for project "qwiklabs-gcp-03-1df6c0aeaf28"' page. The 'MEMBERS' tab is active, showing a table of service accounts and their roles. A red box highlights the first entry, which is the default compute service account with the 'Editor' role.

Type	Member	Name	Role
<input type="checkbox"/>	729328892908-compute@developer.gserviceaccount.com	Compute Engine default service account	Editor
<input type="checkbox"/>	729328892908@cloudbuild.gserviceaccount.com	Cloud Build Service Account	
<input type="checkbox"/>	729328892908@cloudservices.gserviceaccount.com	Google APIs Service Agent	Editor
<input type="checkbox"/>	admiral@qwiklabs-services-prod.iam.gserviceaccount.com		Owner
<input type="checkbox"/>	qwiklabs-gcp-03-1df6c0aeaf28@appspot.gserviceaccount.com	App Engine default	Editor

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 (or Basic) > Editor**. Click **Save**.

The screenshot shows the Google Cloud Platform console with the 'IAM & Admin' section selected. The 'IAM' page is open, displaying a list of permissions for the project 'qwiklabs-gcp-03-1df6c0aeaf28'. The 'New members' dialog is open, showing the email address '729328892908-compute@developer.gserviceaccount.com' entered. The 'Role' dropdown is set to 'Editor'. The 'Condition' dropdown is set to 'Add condition'. The 'Send notification email' checkbox is checked. The 'SAVE' button is highlighted.

Google Cloud Platform | qwiklabs-gcp-03-1df6c0aeaf28

IAM & Admin | IAM | ADD | REMOVE

PERMISSIONS | RECOMMENDATIONS

Permissions for project "qwiklabs-gcp-03-1df6c0aeaf28"

These permissions affect this project and its resources.

View By: MEMBERS | ROLES

Filter table

Type	Member
<input type="checkbox"/>	729328892908@clo
<input type="checkbox"/>	729328892908@clo
<input type="checkbox"/>	admiral@qwiklabs-s
<input type="checkbox"/>	qwiklabs-gcp-03-1df6c0aeaf28@app
<input type="checkbox"/>	qwiklabs-gcp-03-1df6c0aeaf28.iam.g

Add members to "qwiklabs-gcp-03-1df6c0aeaf28"

Add members, roles to "qwiklabs-gcp-03-1df6c0aeaf28" project

Enter one or more members below. Then select a role for these members to grant them access to your resources. Multiple roles allowed. [Learn more](#)

New members

729328892908-compute@developer.gserviceaccount.com

Role: Editor

Condition: Add condition

+ ADD ANOTHER ROLE

☐ Send notification email

This email will inform members that you've granted them access to this role for 'qwiklabs-gcp-03-1df6c0aeaf28'

SAVE | CANCEL

# Download the starter code

Open a session in Cloud Shell and run the following command to get Dataflow Python Examples from [Google Cloud's professional services GitHub](#):

```
gsutil -m cp -R gs://spls/gsp290/dataflow-python-examples .
```

Now set a variable equal to your project id, replacing `<YOUR-PROJECT-ID>` with your lab Project ID:

```
export PROJECT=<YOUR-PROJECT-ID>
gcloud config set project $PROJECT
```

## Create Cloud Storage Bucket

Use the `make bucket` command to create a new regional bucket in the `us-central1` region within your project:

```
gsutil mb -c regional -l us-central1 gs://$PROJECT
```

### Test Completed Task

Click **Check my progress** to verify your performed task.

Create a Cloud Storage Bucket

Check my progress

*Assessment Completed!*



# Copy Files to Your Bucket

Use the `gsutil` command to copy files to the Cloud Storage bucket you just created:

```
gsutil cp gs://splis/gsp290/data_files/usa_names.csv gs://$PROJECT/data_files/  
gsutil cp gs://splis/gsp290/data_files/head_usa_names.csv gs://$PROJECT/data_files/
```

## Test Completed Task

Click **Check my progress** to verify your performed task.

Copy Files to Your Bucket

Check my progress

*Assessment Completed!*

# Create the BigQuery Dataset

Create a dataset in BigQuery called `lake`. This is where all of your tables will be loaded in BigQuery:

```
bq mk lake
```

## Test Completed Task

Click **Check my progress** to verify your performed task.

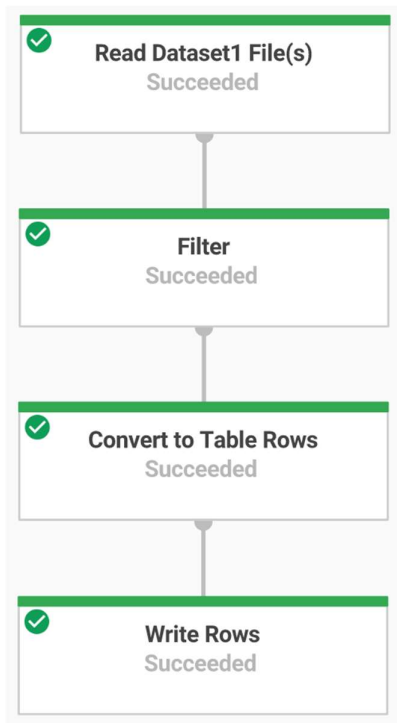
Create the BigQuery Dataset (name: lake)

Check my progress

*Assessment Completed! Dataset ID(s): ["lake"]*

# Build a Dataflow Pipeline

In this section you will create an append-only Dataflow which will ingest data into the BigQuery table. You can use the built-in code editor which will allow you to view and edit the code in the Google Cloud console.



## Open Code Editor

Navigate to the source code by clicking on the **Open Editor** icon in Cloud Shell:



If prompted click on **Open in New Window**. It will open the code editor in new window.

# Data Ingestion

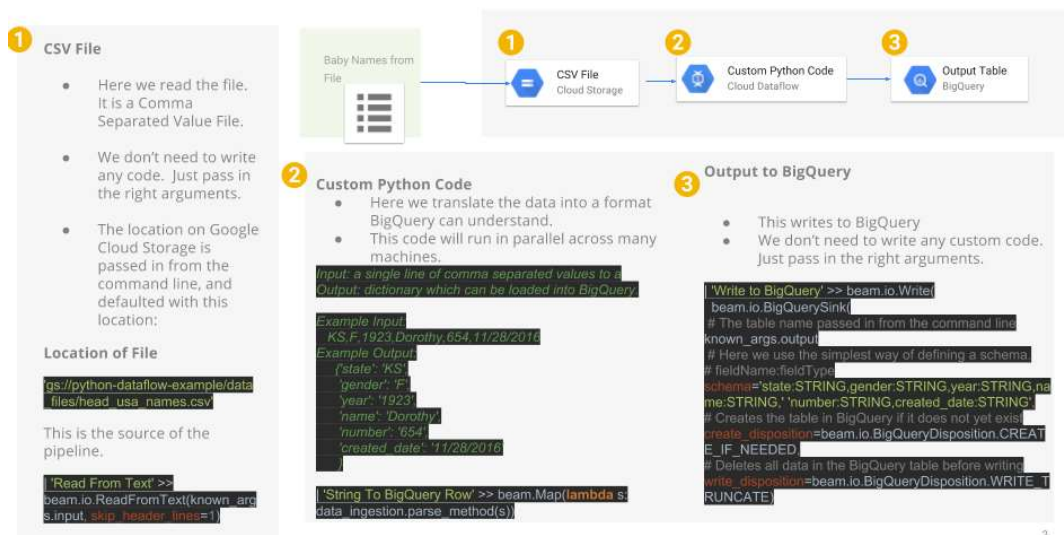
You will now build a Dataflow pipeline with a TextIO source and a BigQueryIO destination to ingest data into BigQuery. More specifically, it will:

- Ingest the files from Cloud Storage.
- Filter out the header row in the files.
- Convert the lines read to dictionary objects.
- Output the rows to BigQuery.

## Review pipeline python code

In the Code Editor navigate to `dataflow-python-examples > dataflow_python_examples` and open the `data_ingestion.py` file. Read through the comments in the file, which explain what the code is doing. This code will populate the data in BigQuery.

Ingest from File to BigQuery



# Run the Apache Beam Pipeline

Return to your Cloud Shell session for this step. You will now do a bit of set up for the required python libraries.

Run the following to set up the python environment:

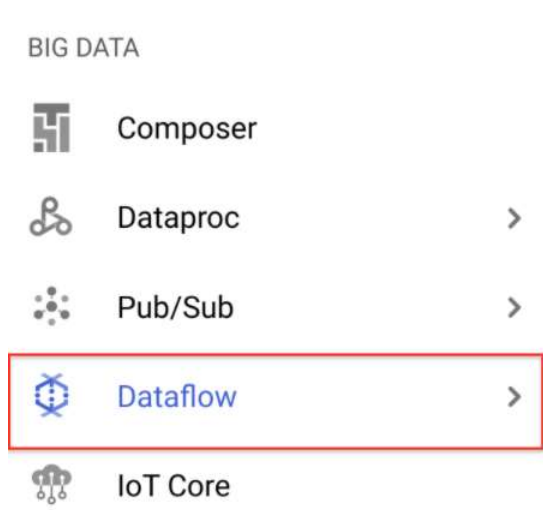
```
cd dataflow-python-examples/  
# Here we set up the python environment.  
# Pip is a tool, similar to maven in the java world  
sudo pip install virtualenv  
  
#Dataflow requires python 3.7  
virtualenv -p python3 venv  
  
source venv/bin/activate  
pip install apache-beam[gcp]==2.24.0
```

You will run the Dataflow pipeline in the cloud.

The following will spin up the workers required, and shut them down when complete:

```
python dataflow_python_examples/data_ingestion.py --project=$PROJECT --region=us-  
centrall --runner=DataflowRunner --staging_location=gs://$PROJECT/test --temp_location  
gs://$PROJECT/test --input gs://$PROJECT/data_files/head_usa_names.csv --  
save_main_session
```

Return to the Cloud Console and open the **Navigation menu > Dataflow** to view the status of your job.



Click on the name of your job to watch it's progress. Once your **Job Status** is **Succeeded**.

Navigate to BigQuery (**Navigation menu > BigQuery**) see that your data has been populated.

## BIG DATA



Composer



Dataproc



Pub/Sub



Dataflow

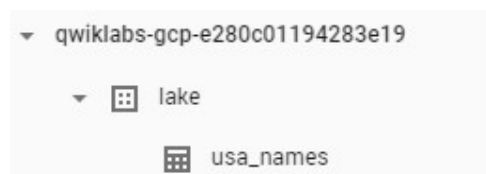


IoT Core



BigQuery

Click on your project name to see the **usa\_names** table under the `lake` dataset.



Click on the table then navigate to the **Preview** tab to see examples of the `usa_names` data.

**Note:** If you don't see the `usa_names` table, try refreshing the page or view the tables using the classic BigQuery UI.

## Test Completed Task

Click **Check my progress** to verify your performed task.

# Data Transformation

You will now build a Dataflow pipeline with a TextIO source and a BigQueryIO destination to ingest data into BigQuery. More specifically, you will:

- Ingest the files from Cloud Storage.
- Convert the lines read to dictionary objects.
- Transform the data which contains the year to a format BigQuery understands as a date.
- Output the rows to BigQuery.

## Review pipeline python code

Navigate to `data_transformation.py` and open it in Code Editor. Read through the comments in the file which explain what the code is doing.

## Run the Apache Beam Pipeline

You will run the Dataflow pipeline in the cloud. This will spin up the workers required, and shut them down when complete.

Run the following commands to do so:

```
python dataflow_python_examples/data_transformation.py --project=$PROJECT --region=us-central1 --runner=DataflowRunner --staging_location=gs://$PROJECT/test --temp_location gs://$PROJECT/test --input gs://$PROJECT/data_files/head_usa_names.csv --save_main_session
```

Navigate to **Navigation menu > Dataflow** and click on the name of this job view the status.

When your **Job Status** is **Succeeded** in the Dataflow Job Status screen, navigate to BigQuery to check to see that your data has been populated.

You should see the **usa\_names\_transformed** table under the lake dataset.

Click on the table and navigate to the **Preview** tab to see examples of the `usa_names_transformed` data.

**Note:** If you don't see the `usa_names_transformed` table, try refreshing the page or view the tables using the classic BigQuery UI.

### Test Completed Task

Click **Check my progress** to verify your performed task.

## Data Enrichment

You will now build a Dataflow pipeline with a TextIO source and a BigQueryIO destination to ingest data into BigQuery. More specifically, you will:

- Ingest the files from Cloud Storage.
- Filter out the header row in the files.
- Convert the lines read to dictionary objects.
- Output the rows to BigQuery.

# Review pipeline python code

Navigate to `data_enrichment.py` and open it in Code Editor. Check out the comments which explain what the code is doing. This code will populate the data in BigQuery.

Line 83 currently looks like:

```
values = [x.decode('utf8') for x in csv_row]
```

Edit it so it looks like the following:

```
values = [x for x in csv_row]
```

## Run the Apache Beam Pipeline

Here you'll run the Dataflow pipeline in the cloud. Run the following to spin up the workers required, and shut them down when complete:

```
python dataflow_python_examples/data_enrichment.py --project=$PROJECT --region=us-central1 --runner=DataflowRunner --staging location=gs://$PROJECT/test --temp_location gs://$PROJECT/test --input gs://$PROJECT/data_files/head_usa_names.csv --save_main_session
```

Navigate to **Navigation menu > Dataflow** to view the status of your job.

Once your **Job Status** is **Succeed** in the Dataflow Job Status screen, navigate to BigQuery to check to see that your data has been populated.

You should see the **usa\_names\_enriched** table under the lake dataset.

Click on the table and navigate to the Preview tab to see examples of data for the data.

**Note:** If you don't see the `usa_names_enriched` table, try refreshing the page or view the tables using the classic BigQuery UI.

### Test Completed Task

Click **Check my progress** to verify your performed task.



# Data lake to Mart

Now build a Dataflow pipeline that reads data from 2 BigQuery data sources, and then joins the data sources. Specifically, you:

- Ingest files from 2 BigQuery sources.
- Join the 2 data sources.
- Filter out the header row in the files.
- Convert the lines read to dictionary objects.
- Output the rows to BigQuery.

## Review pipeline python code

Navigate to `data_lake_to_mart.py` and open it in Code Editor. Read through the comments in the file which explain what the code is doing. This code will populate the data in BigQuery.

# Run the Apache Beam Pipeline

Now you'll run the Dataflow pipeline in the cloud. Run the following to spin up the workers required, and shut them down when complete:

```
python dataflow python examples/data_lake_to_mart.py --  
worker_disk_type="compute.googleapis.com/projects//zones//diskTypes/pd-ssd" --  
max_num_workers=4 --project=$PROJECT --runner=DataflowRunner --  
staging_location=gs://$PROJECT/test --temp_location gs://$PROJECT/test --  
save_main_session --region=us-central1
```

Navigate to **Navigation menu > Dataflow** and click on the name of this new job to view the status.

Once you've **Job Status** is **Succeeded** in the Dataflow Job Status screen, navigate to BigQuery to check to see that your data has been populated.

You should see the **orders\_denormalized\_sideinput** table under the lake dataset.

Click on the table and navigate to the **Preview** section to see examples of orders\_denormalized\_sideinput data.

**Note:** If you don't see the orders\_denormalized\_sideinput table, try refreshing the page or view the tables using the classic BigQuery UI.

## Test Completed Task

Click **Check my progress** to verify your performed task.

# 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.

ETL stands for \_\_\_\_\_.  
Extract, Transform and Load

# Congratulations!

You have used python files to ingest data into BigQuery using Dataflow.

## Finish your Quest



This self-paced lab is part of the Qwiklabs [Data Engineering Quest](#). 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 [Predict Visitor Purchases with a Classification Model in BQML](#), or check out these suggestions:

- [Predict Housing Prices with Tensorflow and AI Platform](#)
- [Cloud Composer: Copy BigQuery Tables Across Different Locations](#)

## Next steps / learn more

Looking for more? Check out official documentation on:

- [Google Dataflow](#)
- [BigQuery](#)
- Also review the [Apache Beam Programming Guide](#) for more advanced concepts.

## 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 October 26, 2020

Lab Last Tested October 26, 2020

Copyright 2021 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.