Serving ML Predictions in batch and real-time

2 hoursFree

Rate Lab

Overview

Duration is 1 min

In this lab, you run Dataflow pipelines to serve predictions for batch requests as well as streaming in real-time.

What you learn

In this lab, you write code to:

- Create a prediction service that calls your trained model deployed in Cloud to serve predictions
- Run a Dataflow job to have the prediction service read in batches from a CSV file and serve predictions

- Run a streaming Dataflow pipeline to read requests real-time from Cloud Pub/Sub and write predictions into a BigQuery table

Setup

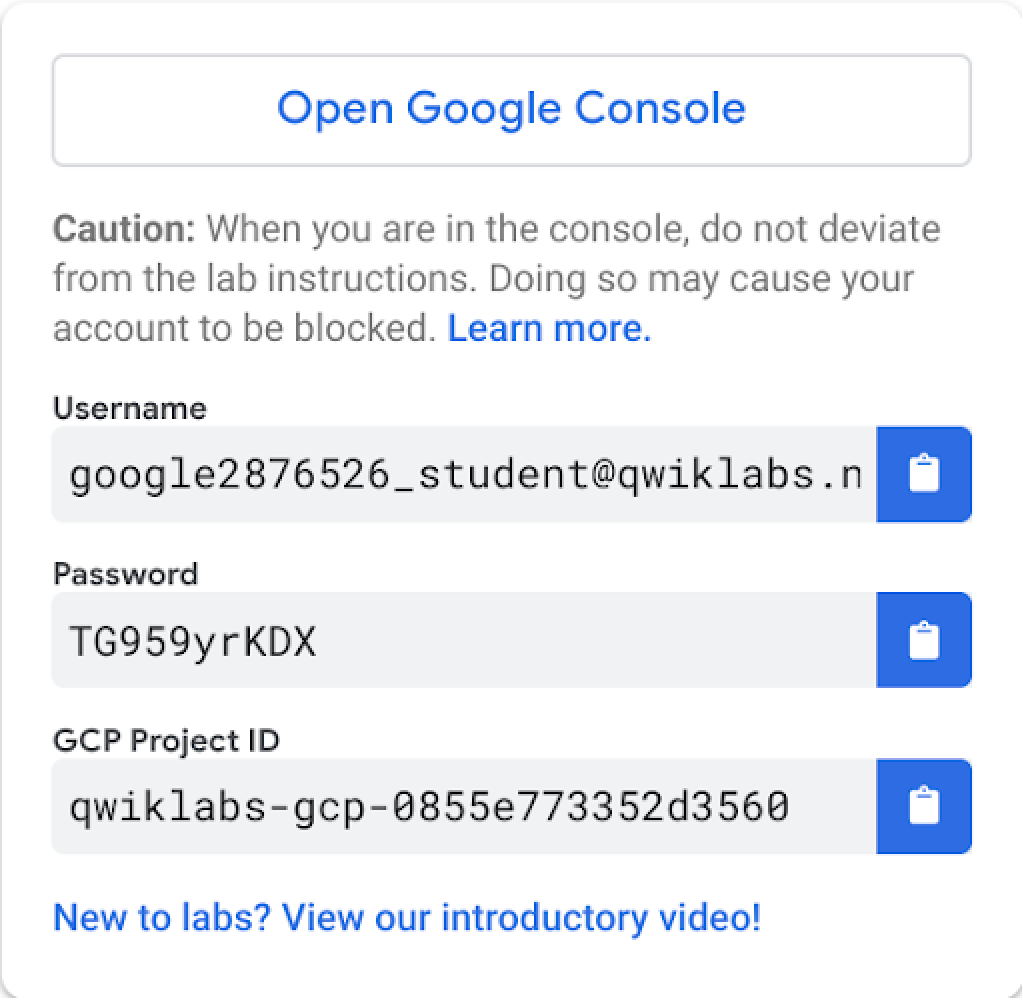
For each lab, you get a new Google Cloud project and set of resources for a fixed time at no cost.

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.

3. When ready, click START LAB.
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**.

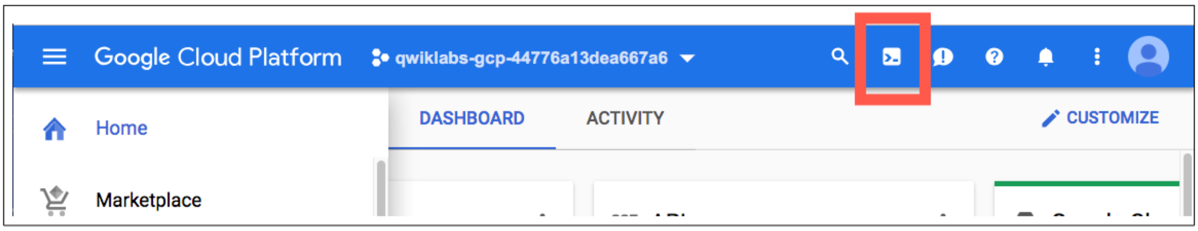
7. Accept the terms and skip the recovery resource page.

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.

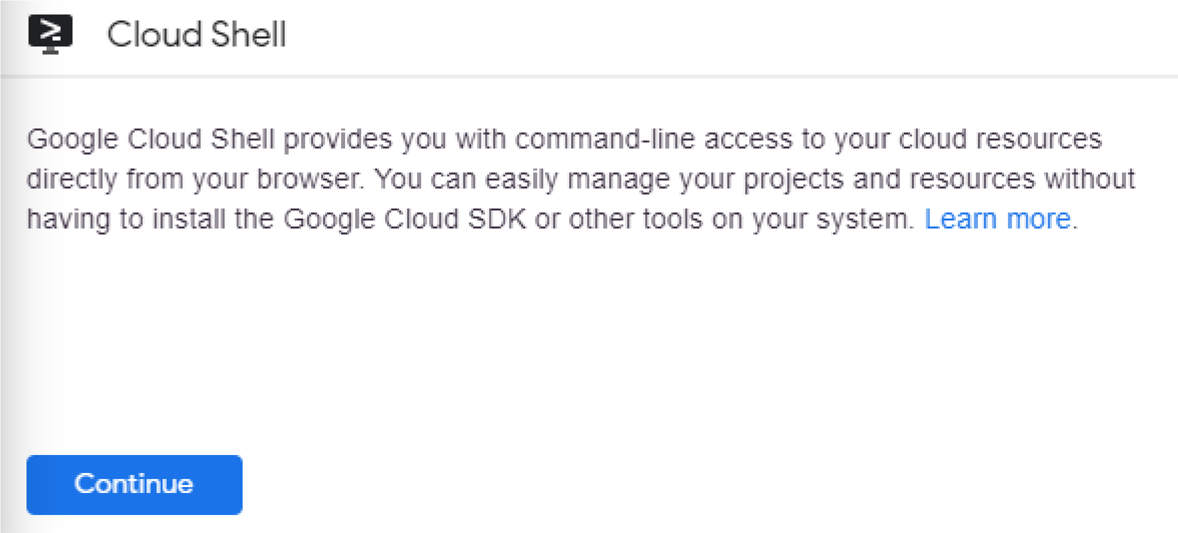
Activate Cloud Shell

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.

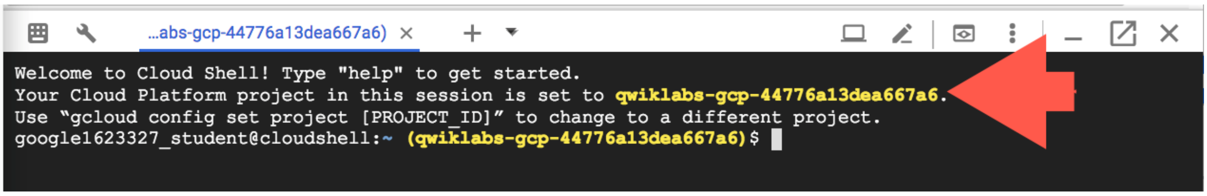


Click **Continue**.


Cloud Shell

Google Cloud Shell provides you with command-line access to your cloud resources directly from your browser. You can easily manage your projects and resources without having to install the Google Cloud SDK or other tools on your system. Learn more.

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
content_copy
```

(Output)

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

(Example output)

```
Credentialed accounts:
 - google1623327_student@qwiklabs.netcontent_copy
```

You can list the project ID with this command:

```
gcloud config list project
content_copy
```

(Output)

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

(Example output)

```
[core]
project = qwiklabs-gcp-44776a13dea667a6content_copy
```

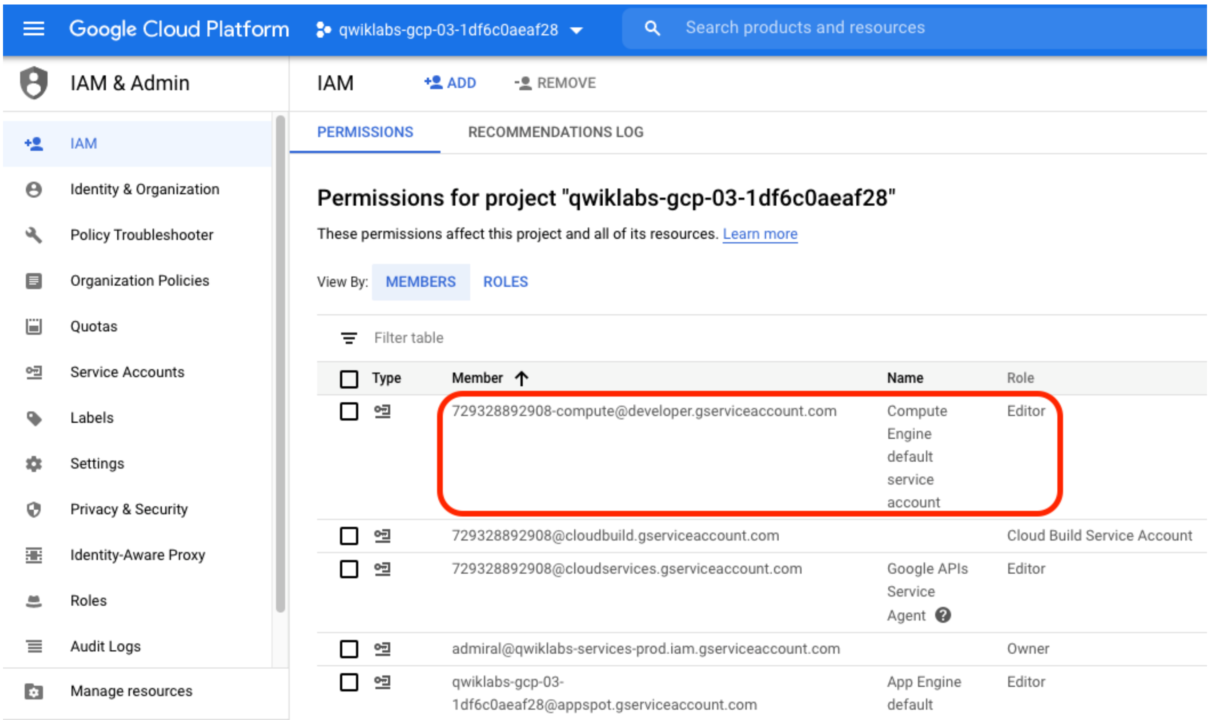
For full documentation of `gcloud` see the gcloud command-line tool overview.

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**.



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

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

□ Type	Member	Name	Role
□	729328892908-compute@developer.gserviceaccount.com	Compute Engine default service account	Editor
□	729328892908@cloudbuild.gserviceaccount.com		Cloud Build Service Account
□	729328892908@cloudservices.gserviceaccount.com	Google APIs Service Agent	Editor
□	admiral@qwiklabs-services-prod.iam.gserviceaccount.com		Owner
□	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**.

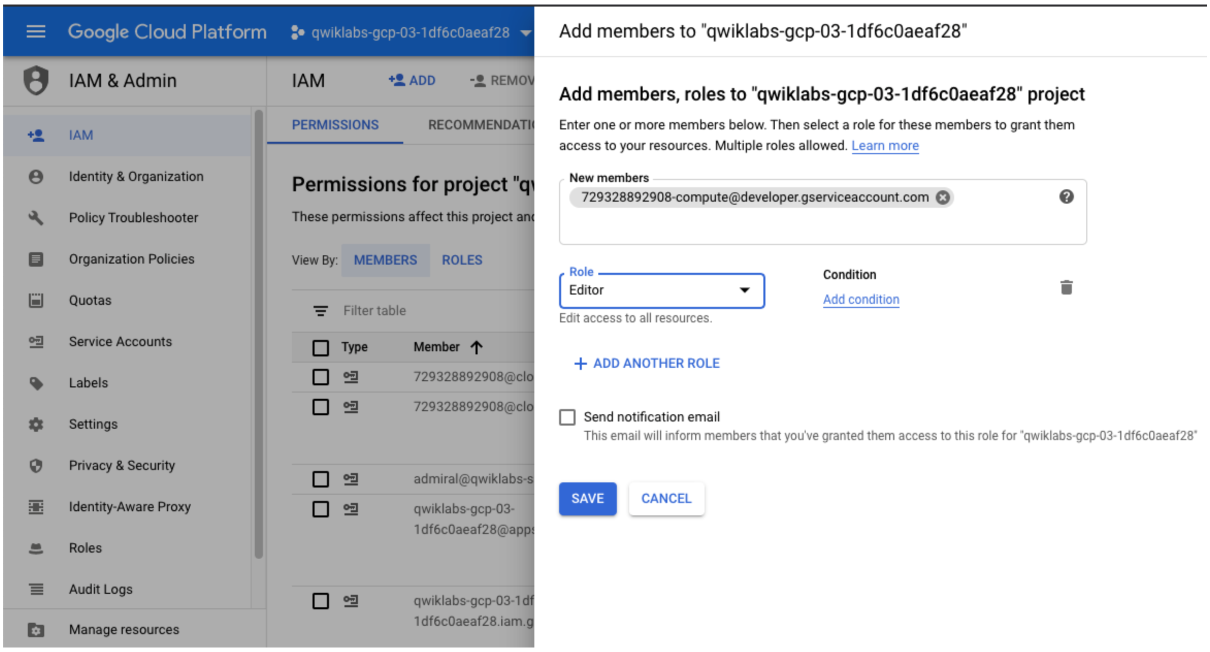
- At the top of the **IAM** page, click **Add**.

- For **New members**, type:

```
{project-number}-compute@developer.gserviceaccount.com
content_copy
```

Replace `{project-number}` with your project number.

- For **Role**, select **Project** (or Basic) > **Editor**. Click **Save**.



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

Edit access to all resources.

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

Creating the virtual environment

Execute the following command to download and update the packages list.

```
sudo apt-get update
content_copy
```

Python virtual environments are used to isolate package installation from the system.

```
sudo apt-get install virtualenv
content_copy
```

If prompted [Y/n], press `Y` and then `Enter`.

```
virtualenv -p python3 venv
content_copy
```

Activate the virtual environment.

```
source venv/bin/activate
content_copy
```

Copy trained model

Step 1

Set necessary variables and create a bucket:

```
REGION=us-central1
BUCKET=$(gcloud config get-value project)
TFVERSION=2.1
gsutil mb -l ${REGION} gs://${BUCKET}content_copy
```

Step 2

Copy trained model into your bucket:

```
gsutil -m cp -R gs://cloud-training-demos/babyweight/trained_model
gs://${BUCKET}/babyweightcontent_copy
```

Deploy trained model

Step 1

Set necessary variables:

```
MODEL_NAME=babyweight
MODEL_VERSION=ml_on_gcp
MODEL_LOCATION=$(gsutil ls gs://${BUCKET}/babyweight/export/exporter/ |
tail -1)content_copy
```

Step 2

Deploy trained model:

```
gcloud ai-platform models create ${MODEL_NAME} --regions $REGION
gcloud ai-platform versions create ${MODEL_VERSION} --model ${MODEL_NAME} --origin ${MODEL_LOCATION} --runtime-version $TFVERSIONcontent_copy
```

Browse lab files

Duration is 5 min

Step 1

Clone the course repository:

```
cd ~
git clone https://github.com/GoogleCloudPlatform/training-data-analystcontent_copy
```

Step 2

In Cloud Shell, navigate to the folder containing the code for this lab:

```
cd ~/training-data-analyst/courses/machine_learning/deepdive/06_structured/labs/servingcontent_copy
```

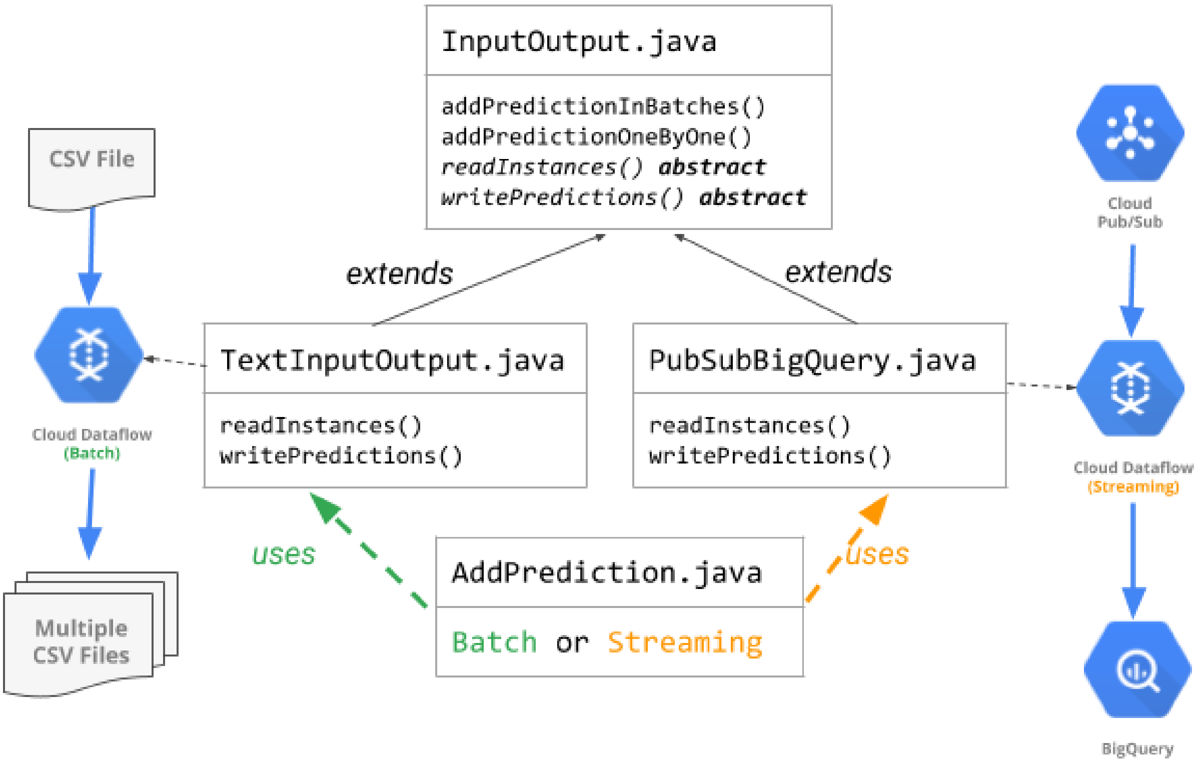
Step 3

Run the `what_to_fix.sh` script to see a list of items you need to add/modify to existing code to run your app:

```
./what_to_fix.shcontent_copy
```

As a result of this, you will see a list of filenames and lines within those files marked with **TODO**. These are the lines where you have to add/modify code. For this lab, you will focus on #TODO items for **.java files only**, namely `BabyweightMLService.java` : which is your prediction service.

How the code is organized



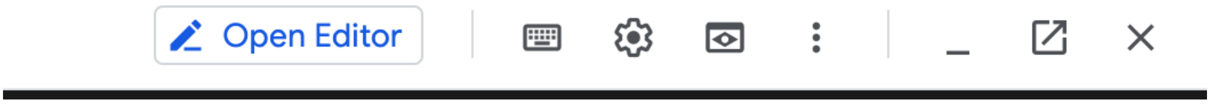
Prediction service

In this section, you fix the code in **BabyweightMLService.java** and test it with the **run_once.sh** script that is provided. If you need help with the code, look at the next section that provides hints on how to fix code in BabyweightMLService.java.

Step 1

You may use the Cloud Shell code editor to view and edit the contents of these files.

In Cloud Shell, click the **Open Editor** icon on the top right.



Step 2

After it is launched, navigate to the following directory:

```
training-data-analyst/courses/machine_learning/deepdive/06_structured/labs/serving/pipeline/src/main/java/com/google/cloud/training/mlongcpcontent_copy
```

Step 3

Open the `BabyweightMLService.java` files and replace *#TODOs* in the code.

Step 4

Once completed, go into your Cloud Shell and run the `run_once.sh` script to test your ML service.

```
cd ~/training-data-analyst/courses/machine_learning/deepdive/06_structured/labs/serving
./run_once.shcontent_copy
```

Serve predictions for batch requests

This section of the lab calls AddPrediction.java that takes a batch input (one big CSV), calls the prediction service to generate baby weight predictions and writes them into local files (multiple CSVs).

Step 1

In your Cloud Shell code editor, open the `AddPrediction.java` file available in the following directory:

```
training-data-analyst/courses/machine_learning/deepdive/06_structured/labs/serving/pipeline/src/main/java/com/google/cloud/training/mlongcpcontent_copy
```

Step 2

Look through the code and notice how, based on input argument, it decides to set up a batch or streaming pipeline, and creates the appropriate TextInputOutput or PubSubBigQuery io object respectively to handle the reading and writing.

Note: Look back at the diagram in "how code is organized" section to make sense of it all.

Step 3

Test batch mode by running the `run_ontext.sh` script provided in the lab directory:

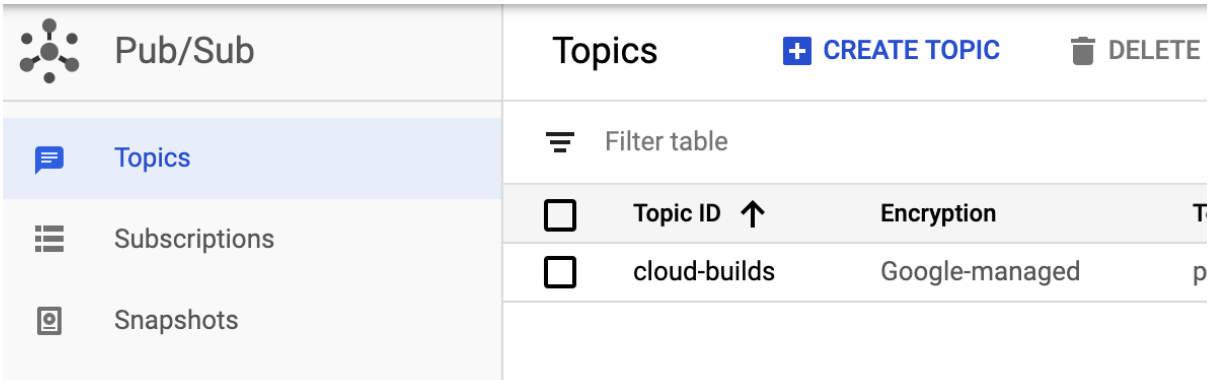
```
cd ~/training-data-
analyst/courses/machine_learning/deepdive/06_structured/labs/servingcontent_c
opy
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64content_copy
./run_ontext.shcontent_copy
```

Serve predictions real-time with a streaming pipeline

In this section of the lab, you will launch a streaming pipeline with Dataflow, which will accept incoming information from Cloud Pub/Sub, use the info to call the prediction service to get baby weight predictions, and finally write that info into a BigQuery table.

Step 1

On your GCP Console's left-side menu, go into **Pub/Sub** and click the **CREATE TOPIC** button on top. Create a topic called **babies**.



□	Topic ID ↑	Encryption	T
□	cloud-builds	Google-managed	p

Step 2

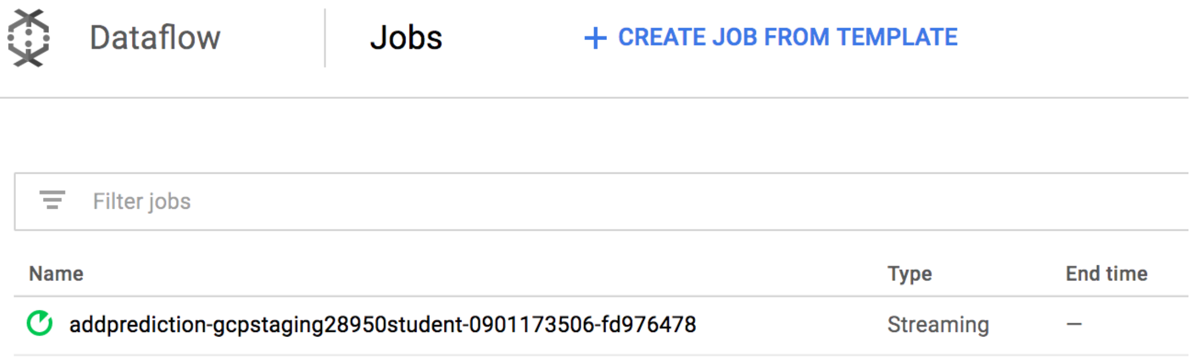
Back in your Cloud Shell, modify the script `run_dataflow.sh` to get Project ID (using --*project*) from command line arguments, and then run as follows:

```
cd ~/training-data-
analyst/courses/machine_learning/deepdive/06_structured/labs/serving
./run_dataflow.shcontent_copy
```

This will create a streaming Dataflow pipeline.

Step 3

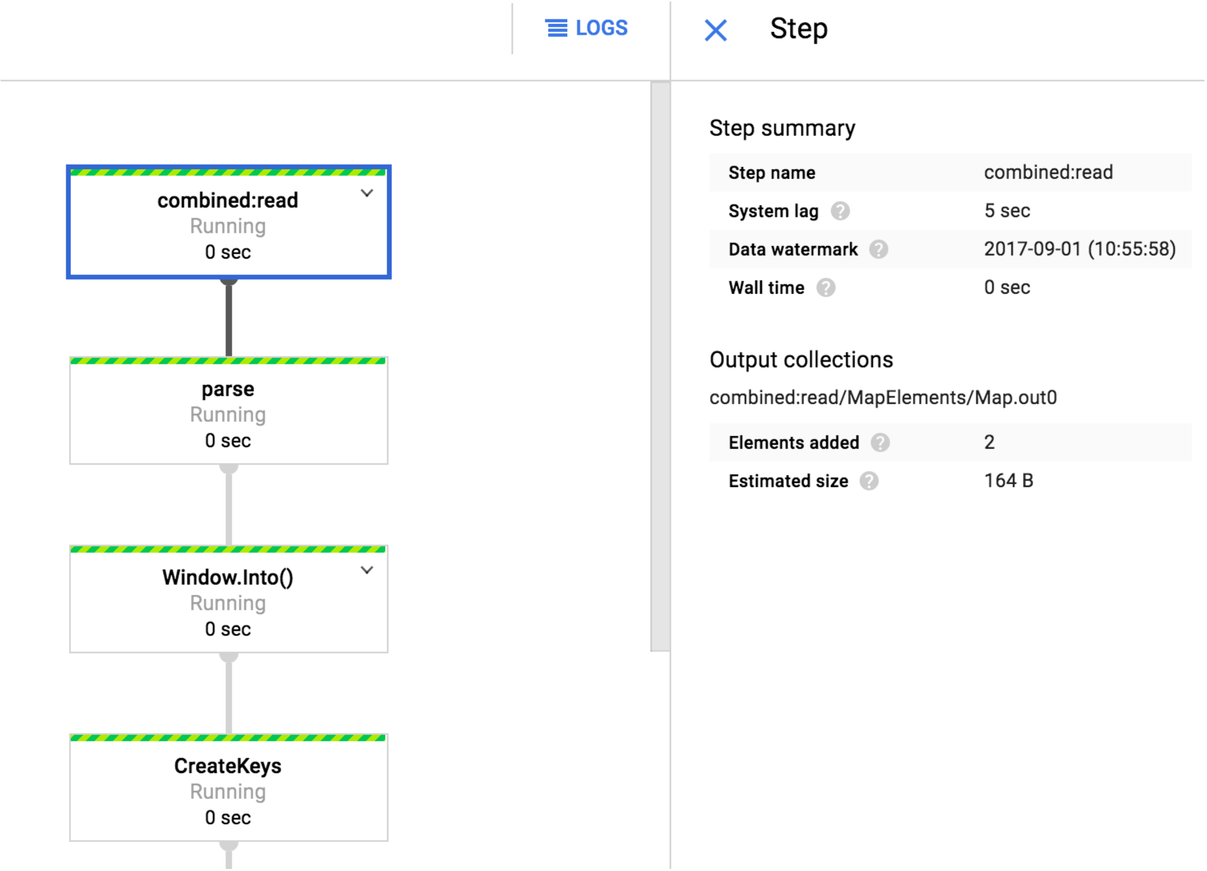
Back in your GCP Console, use the left-side menu to go into **Dataflow** and verify that the streaming job is created.



Name	Type	End time
addprediction-gcpstaging28950student-0901173506-fd976478	Streaming	–

Step 4

Next, click on the job name to view the pipeline graph. Click on the pipeline steps (boxes) and look at the run details (like system lag, elements added, etc.) of that step on the right side.



Step summary

Step name	combined:read
System lag	5 sec
Data watermark	2017-09-01 (10:55:58)
Wall time	0 sec

Output collections

combined:read/MapElements/Map.out0

Elements added	2
Estimated size	164 B

This means that your pipeline is running and waiting for input. Let's provide input through the Pub/Sub topic.

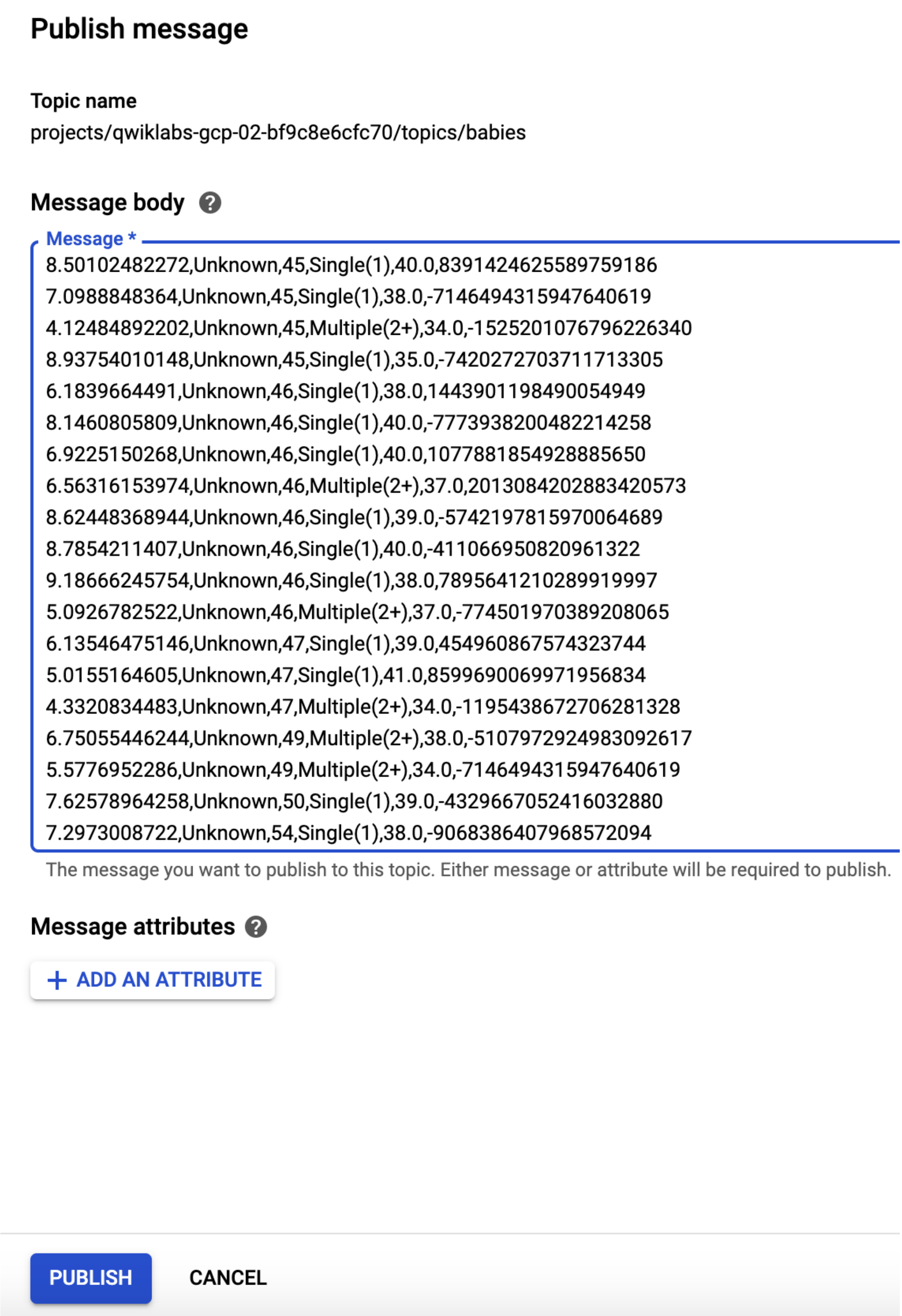
Step 5

Copy some lines from your example.csv.gz:

```
cd ~/training-data-
analyst/courses/machine_learning/deepdive/06_structured/labs/serving
zcat exampledata.csv.gzcontent_copy
```

Step 6

On your GCP Console, go back into **Pub/Sub**, click on the **babies** topic, and then click on **Publish message** button on top and then click on **Publish single message**. In the message box, paste the lines you just copied from exampledata.csv.gz and click on **Publish** button.



Publish message

Topic name

projects/qwiklabs-gcp-02-bf9c8e6cfc70/topics/babies

Message body

Message *

```
8.50102482272,Unknown,45,Single(1),40.0,8391424625589759186
7.0988848364,Unknown,45,Single(1),38.0,-7146494315947640619
4.12484892202,Unknown,45,Multiple(2+),34.0,-1525201076796226340
8.93754010148,Unknown,45,Single(1),35.0,-7420272703711713305
6.1839664491,Unknown,46,Single(1),38.0,1443901198490054949
8.1460805809,Unknown,46,Single(1),40.0,-7773938200482214258
6.9225150268,Unknown,46,Single(1),40.0,1077881854928885650
6.56316153974,Unknown,46,Multiple(2+),37.0,2013084202883420573
8.62448368944,Unknown,46,Single(1),39.0,-5742197815970064689
8.7854211407,Unknown,46,Single(1),40.0,-4110669508209613222
9.18666245754,Unknown,46,Single(1),38.0,7895641210289919997
5.0926782522,Unknown,46,Multiple(2+),37.0,-7745019703892080065
6.13546475146,Unknown,47,Single(1),39.0,4549608675743237444
5.0155164605,Unknown,47,Single(1),41.0,8599690069971956834
4.3320834483,Unknown,47,Multiple(2+),34.0,-1195438672706281328
6.75055446244,Unknown,49,Multiple(2+),38.0,-5107972924983092617
5.5776952286,Unknown,49,Multiple(2+),34.0,-7146494315947640619
7.62578964258,Unknown,50,Single(1),39.0,-4329667052416032880
7.2973008722,Unknown,54,Single(1),38.0,-9068386407968572094
```

The message you want to publish to this topic. Either message or attribute will be required to publish.

Message attributes

Step 7

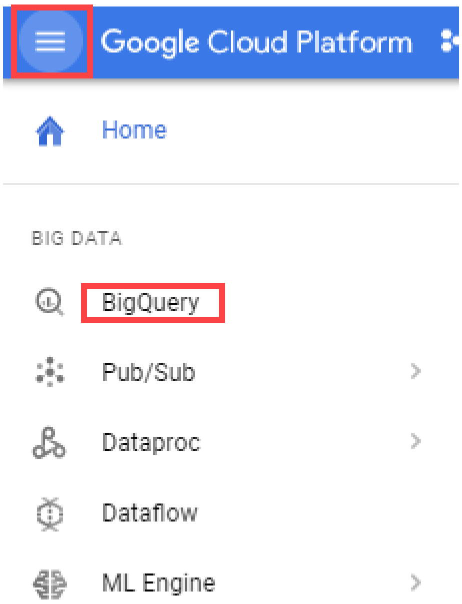
You may go back into Dataflow jobs on your GCP Console, click on your job and see how the run details have changed for the steps, for example click on **write_toBQ** and look at Elements added.

Step 8

Lets verify that the predicted weights have been recorded into the BigQuery table.

Open BigQuery Console

In the Google Cloud Console, select **Navigation menu** > **BigQuery**:

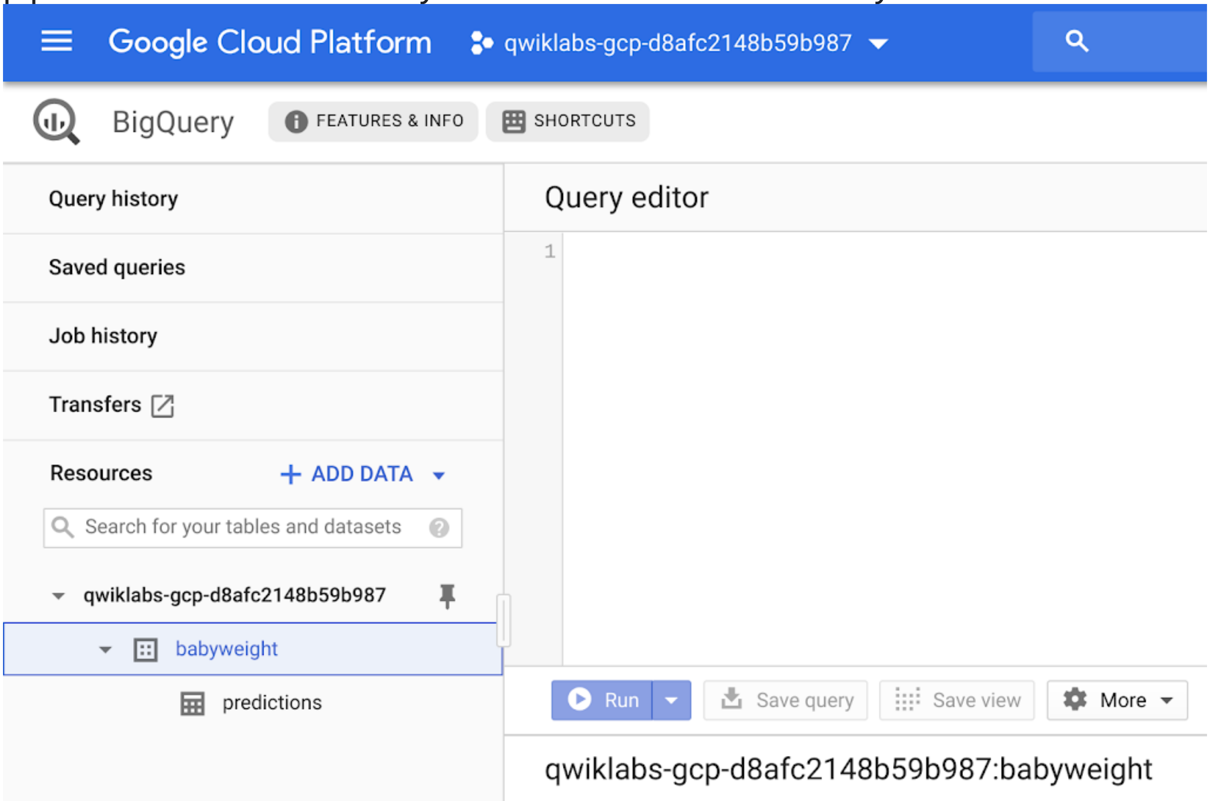


The **Welcome to BigQuery in the Cloud Console** message box opens. This message box provides a link to the quickstart guide and lists UI updates.

Click **Done**.

Look at the left-side menu and you should see the **babyweight** dataset. Click on the blue down arrow to its left, and you should see your **predictions** table.

Note: If you do not see the prediction table, give it a few minutes as the pipeline has allowed-latency and that can add some delay.



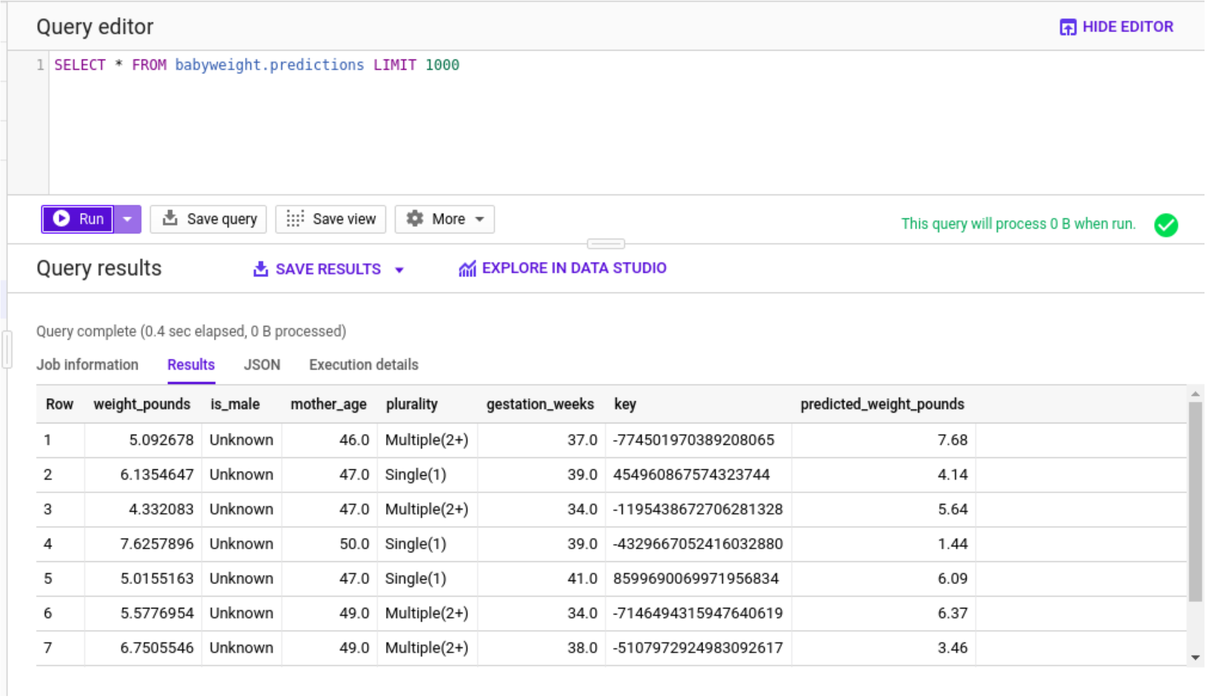
Step 9

Type the query below in the **Query editor** to retrieve rows from your predictions table.

```
SELECT * FROM babyweight.predictions LIMIT 1000content_copy
```

Step 10

Click the **Run** button. Notice the **predicted_weights_pounds** column in the result.



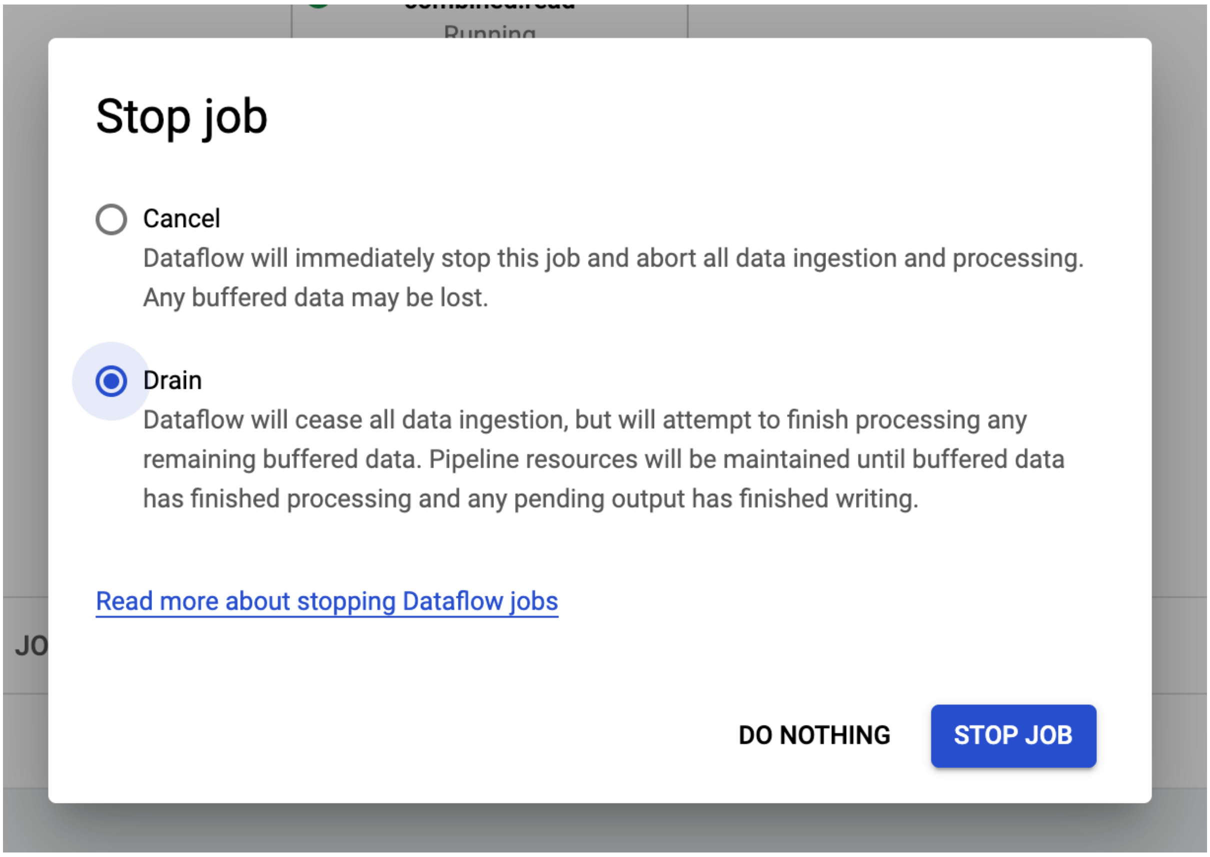
```
SELECT * FROM babyweight.predictions LIMIT 1000
```

Query complete (0.4 sec elapsed, 0 B processed)

Row	weight_pounds	is_male	mother_age	plurality	gestation_weeks	key	predicted_weight_pounds
1	5.092678	Unknown	46.0	Multiple(2+)	37.0	-774501970389208065	7.68
2	6.1354647	Unknown	47.0	Single(1)	39.0	454960867574323744	4.14
3	4.332083	Unknown	47.0	Multiple(2+)	34.0	-1195438672706281328	5.64
4	7.6257896	Unknown	50.0	Single(1)	39.0	-4329667052416032880	1.44
5	5.0155163	Unknown	47.0	Single(1)	41.0	8599690069971956834	6.09
6	5.5776954	Unknown	49.0	Multiple(2+)	34.0	-7146494315947640619	6.37
7	6.7505546	Unknown	49.0	Multiple(2+)	38.0	-5107972924983092617	3.46

Step 11

Remember that your pipeline is still running. You can publish additional messages from your example.csv.gz and verify new rows added to your predictions table. Once you are satisfied, you may stop the Dataflow pipeline by going into your Dataflow Jobs page, and click the **Stop** button on the top. Select **Drain** and click **Stop Job**.



Stop job

○ Cancel
Dataflow will immediately stop this job and abort all data ingestion and processing. Any buffered data may be lost.

● Drain
Dataflow will cease all data ingestion, but will attempt to finish processing any remaining buffered data. Pipeline resources will be maintained until buffered data has finished processing and any pending output has finished writing.

Read more about stopping Dataflow jobs

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.