

# [ML on GCP C7] Serving ML Predictions in batch and real-time

2 hours

Free

[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 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.


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


3. When ready, click  .


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 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 **thislab** 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.

## Start Cloud Shell

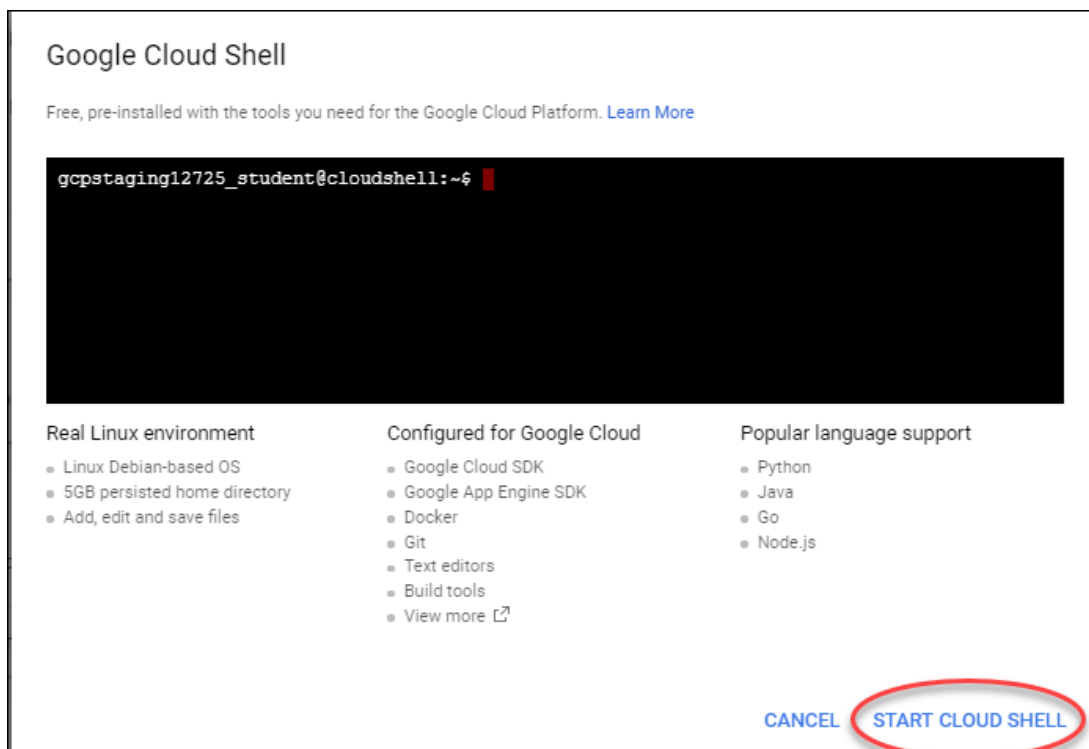
### Activate Google Cloud Shell

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

From the GCP Console click the **Cloud Shell** icon on the top right toolbar:

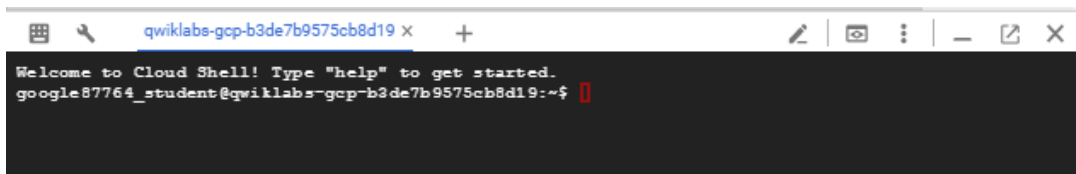


Then click **START CLOUD SHELL**:



You can click **START CLOUD SHELL** immediately when the dialog comes up instead of waiting in the dialog until the Cloud Shell provisions.

It takes a few moments to provision and connects to the environment:



The Cloud Shell is a virtual machine loaded with all the development tools you'll need. It offers a persistent 5GB home directory, and runs on the Google Cloud, greatly enhancing network performance and authentication.

Once connected to the cloud shell, you'll see that you are already authenticated and the project is set to your *PROJECT\_ID*:

```
gcloud auth list
```

Output:

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

**Note:** `gcloud` is the powerful and unified command-line tool for Google Cloud Platform. Full documentation is available on [Google Cloud gcloud Overview](#). It comes pre-installed on Cloud Shell and supports tab-completion.

```
gcloud config list project
```

Output:

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

## Copy trained model

### Step 1

Set necessary variables and create a bucket:

```
REGION=us-central1
BUCKET=$(gcloud config
get-value project)
TFVERSION=1.7
gsutil mb -l ${REGION}
gs://${BUCKET}
```

## Step 2

Copy trained model into your bucket:

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



# Deploy trained model

## Step 1

Set necessary variables:

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



## Step 2

Deploy trained model:

```
gcloud ml-engine models
create ${MODEL_NAME} --
regions $REGION
gcloud ml-engine versions
create ${MODEL_VERSION} --
model ${MODEL_NAME} --
```

```
origin ${MODEL_LOCATION} -  
-runtime-version  
$TFVERSION
```

## Browse lab files

*Duration is 5 min*

### Step 1

Clone the course repository:

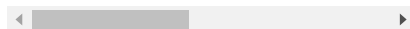
```
cd ~  
git clone  
https://github.com/GoogleClou  
data-analyst
```



### Step 2

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

```
cd ~/training-data-  
analyst/courses/machine_learn
```



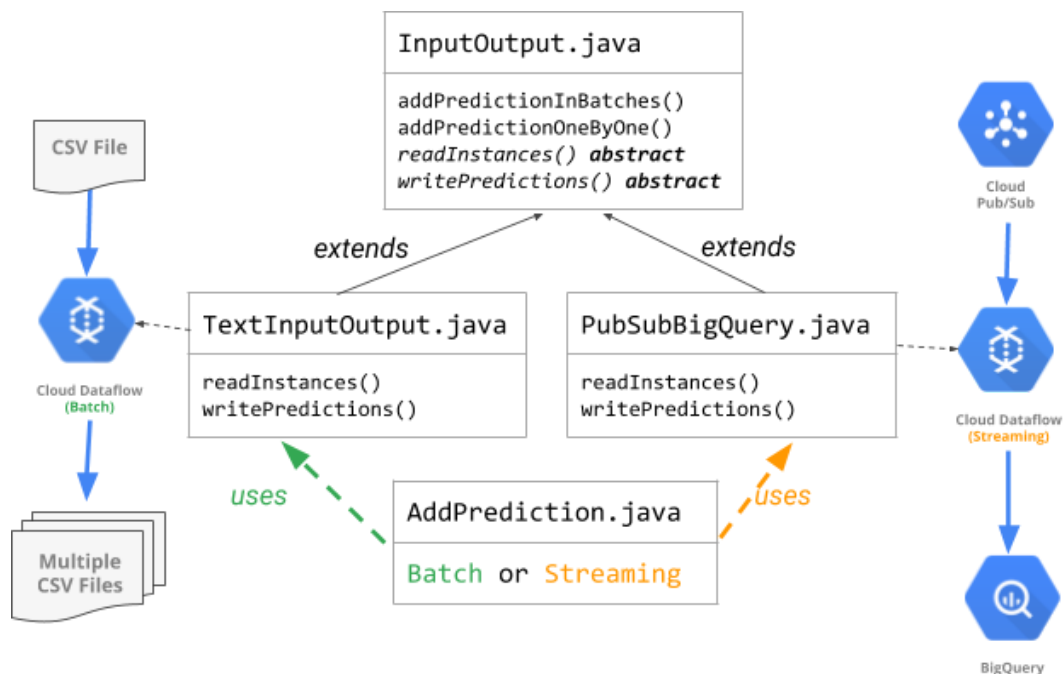
### Step 3

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.sh
```

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

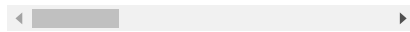
You may use the Cloud Shell code editor to view and edit the contents of these files.

Click on the (  ) icon on the top right of your Cloud Shell window to launch Code Editor.

### Step 2

After it is launched, navigate to the following directory:

```
training-data-  
analyst/courses/machine_learn
```



## 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_learn  
  
./run_once.sh
```



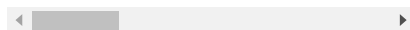
# Serve predictions for batch requests

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_learn
```



## 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_oncontext.sh` script provided in the lab directory:

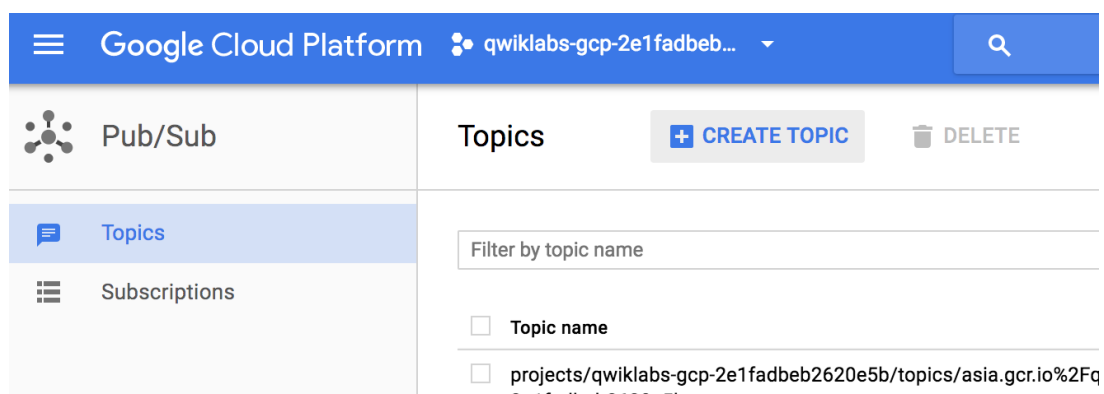
```
cd ~/training-data-analyst/courses/machine_learn  
  
./run_oncontext.sh
```

## 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**.



## 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_learn  
  
./run_dataflow.sh
```



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.

	Dataflow	Jobs	<a href="#">+ CREATE JOB FROM TEMPLATE</a>
<hr/>			
<div> Filter jobs</div>			
<hr/>			
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.

LOGS

Step

combined:read  
Running  
0 sec

parse  
Running  
0 sec

Window.Into()  
Running  
0 sec

CreateKeys  
Running  
0 sec

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_learn

zcat exampledata.csv.gz
```

## Step 6

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

## Publish message

 The topic has no subscriptions in the project. This message might not be delivered.

### Topic

projects/qwiklabs-gcp-2e1fadbeb2620e5b/topics/babies

### Message

```
7.6279942652,False,29,White,1,43.0,False,True,True,74931465496927487
5.3131405142,True,21,Black,1,38.0,False,True,True,74931465496927487
7.6941329438,True,18,White,1,39.0,False,True,True,74931465496927487
7.06140625186,True,24,White,1,39.0,False,True,True,74931465496927487
6.81448851842,False,20,White,1,39.0,True,True,True,74931465496927487
7.1870697412,False,21,White,1,40.0,False,True,True,74931465496927487
```

### Attributes (Optional)

#### Key

#### Value



 Add item

Publish

Cancel

## Step 7

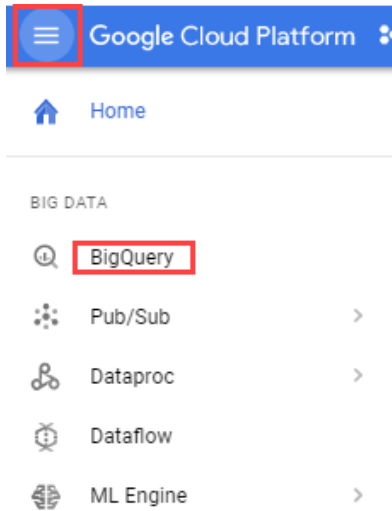
You may go back into Dataflow jobs on your GCP Console, click on your job and see how the run details have changed for the steps, for example click on **write\_toBQ** and look at Elements added.

## 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**:

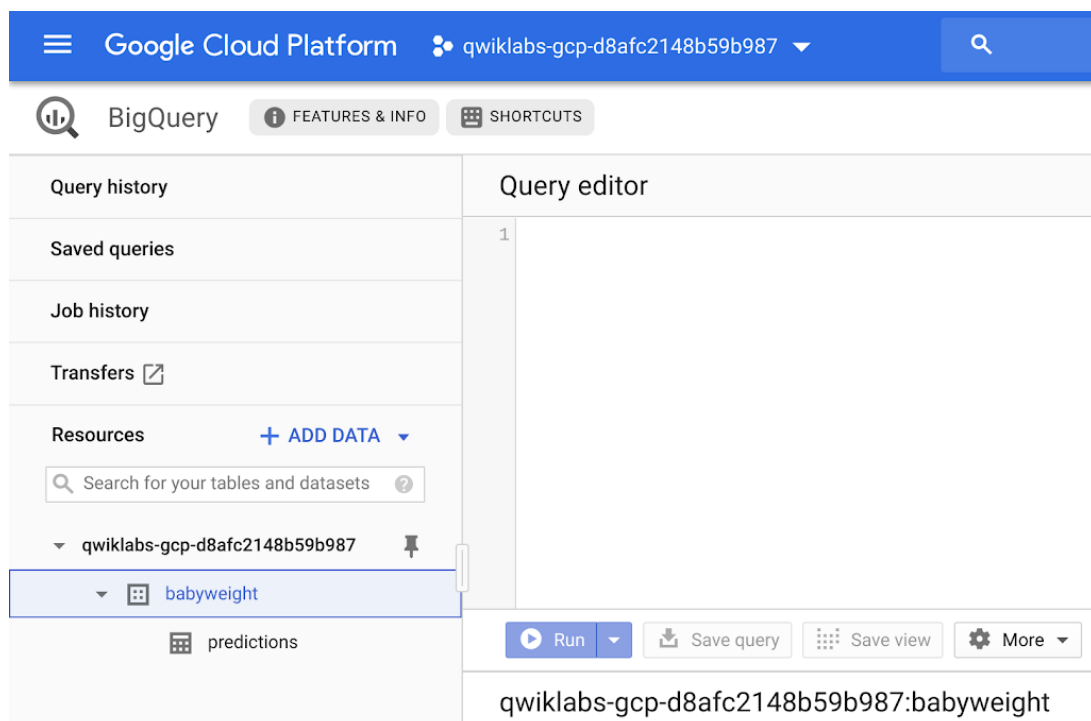


The **Welcome to BigQuery in the Cloud Console** message box opens. This message box provides a link to the quickstart guide and lists UI updates.

Click **Done**.

Look at the left-side menu and you should see the **babyweightdataset**. 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 1000
```

## Step 10

Click the **Run** button. Notice the **predicted\_weights\_pounds** column in the result.

The screenshot shows the 'Query editor' interface with the SQL query: `SELECT * FROM babyweight.predictions LIMIT 1000`. Below the editor is a toolbar with buttons for 'Run', 'Save query', 'Save view', and 'More'. A status message indicates 'This query will process 0 B when run.' with a green checkmark. The 'Query results' section shows the query is complete (0.4 sec elapsed, 0 B processed). Below this are tabs for 'Job information', 'Results', 'JSON', and 'Execution details'. The 'Results' tab is active, displaying a table with 8 columns: Row, weight\_pounds, is\_male, mother\_age, plurality, gestation\_weeks, key, and predicted\_weight\_pounds. The table contains 7 rows of data.

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 job** button on the right side Job summary window. Select **Drain** and click **Stop Job**.

# Job

---

## Job summary

Job name	addprediction-gcpstaging28950student-0901173506-fd976478
Job ID	2017-09-01_10_35_26-15406689007149441589
Job status	 Running <div>Stop job</div>
SDK version	Google Cloud Dataflow SDK for Java 2.0.0
Job type	Streaming
Start time	Sep 1, 2017, 10:35:27 AM
Elapsed time	1 hr 2 min

## Autoscaling

Workers	1
Current state	Worker pool started.

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: March 22, 2019

Lab Last Tested: March 22, 2019

©2019 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.