Pause all 01:59:47

Open Google Console

Caution: When you are in the console, do not delete anything unless you know what it does. Deleting resources can lead to unexpected charges or service disruption. Learn more

Generation

Generation

Generation

GCP Project ID

qwiklabs-gcp-

Serving ML Predictions in batch and real-time

2 hours · Free · ★★★★ · Intermediate

Overview
Setup
Start Cloud Shell
Copy trained model
Deploy trained model
Browse lab files
How the code is organized
Prediction service
Serve predictions for batch requests
Serve predictions real-time with a streaming pipeline
Test your ML

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 the 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 from a 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 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 Cloud Platform.

5. Click **Open Google Console**.

6. Click **Use another account** and copy/paste credentials for **this** step into the prompts.

   If you use other credentials, you'll get errors or **lose 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 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:

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:
 - google1623251224940@qwiklabs.net
```

Example output:

```
Credentialed accounts:
 - student-04-9dd0d0bb9925@qwiklabs.net
```

You can list the project ID with this command:

```
gcloud config list project
```

Output:

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

Example output:

```
[core]
project = qwiklabs-gcp-44776f0f7f0e2bd8
```

Full documentation of **gcloud** is available on Google Cloud gcloud Overview.

Copy trained model

Step 1

Set necessary variables and create a bucket:

```
PROJECT=$(gcloud config get-value project)
REGION=us-central1
gsutil -m -q clone gs://cloud-training/bmlep/bmlep-trained-model
```

Step 2

Copy trained model into your bucket:

```
gsutil -m -q -r gs://cloud-training/bmlep/bmlep-trained-model
gcloud ml-engine models create taxifare
```

Deploy trained model

Step 1

Set necessary variables:

```
MODEL=taxifare
MODEL_VERSION=v1_keras
MODEL_DIRECTORY=$(gsutil ls
gs://${PROJECT}-bmlep/trained-model | tail -1)
```

Step 2

Deploy trained model:

```
gcloud ml-engine models create $MODEL --regions $REGION
gcloud ml-engine versions create $MODEL_VERSION --model
$MODEL --origin $MODEL_DIRECTORY --runtime-version
1.13 --python-version 3.5
```

Browse lab files

Duration is 3 min

Step 1

Clone the course repository:

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

Step 2

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

```
cd training-data-analyst/quwiklabs/serving-ml/solutions
```

Step 3

Run the *ls(1)*, *ls(1) -la*, and *pwd* to see a list of items you need to add/modify to working code to run your app:

```
ls(1) ls(1) -la pwd
```

As a result of this, you will see a list of filenames and how within those files marked with **TODO**. These are the lines where you have to add/modify code. Notice that there are two files ending in *.sh*, two in *.Java*, and one in *.conf*: *RunDataflowGBR.sh*, *RunDataflowNN.sh*, *PredictFromCsv.Java*, *ServePredictions.Java*, *pubsub.conf*.

How the code is organized

Prediction service

In this section, you fix the code in **Solutions/pubsub/serve.java** and test it with the **test_pubsub** script that is provided. If you need help with the code, look at the best practices that provides hints on how to fix code in *Solutions/pubsub/Serve.java*.

Step 1

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

Click on the *pencil* 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:

```
Serving ML/
analyst/quwiklabs/serving-ml/solutions/
```

Step 3

Open the *Solutions/pubsub/Serve.Java* files and replace #TODOs in the code.

Step 4

Once completed, go into your Cloud Shell and run the *test_pubsub.sh* script to test your ML service:

```
./test_pubsub.sh
Serving ML/
analyst/quwiklabs/serving-ml/solutions/
./test_pubsub.sh
```

Serve predictions for batch requests

This section of the lab calls RunPredictions.java that takes a batch input (one file) and a trained model and writes out the prediction (take weight predictions and write them into text files (multiple CSVs).

Step 1

In your Cloud Shell code editor, open the *RunPredictions.Java* file available in the following directory:

```
Serving ML/
analyst/quwiklabs/serving-ml/solutions/
```

Step 2

Look through the code and notice how, based on input argument, it decides to run a local prediction or a cloud prediction. Also notice how the TestInputGroup or PublishOutputs in dataflow.common.Utilities.Utilities handle the reading and writing.

Note: Look back at the diagram in "how code is organized" section to create a mental of all.

Step 3

Test batch mode by running the *run_batch.sh* script provided in the lab directory:

```
./test_pubsub.sh
Serving ML/
analyst/quwiklabs/serving-ml/solutions/
./test_pubsub.sh
```

Serve predictions real-time with a streaming pipeline

In this section of the lab, you will launch a streaming pipeline with Dataflow, that reads from a Cloud Pub/Sub topic and writes into BigQuery. You will call the prediction service in get-weight predictions, weight-predictions, and 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 **taxiin**.

Step 2

Back in your Cloud Shell, modify the script *run_pubsub.sh* (or *.bat* Region ID Bucket) project from command line arguments, and then run as follows:

```
./test_pubsub.sh
Serving ML/
analyst/quwiklabs/serving-ml/solutions/
./test_pubsub.sh
```

This will create a streaming Dataflow pipeline.

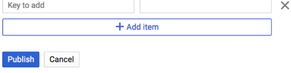
Step 3

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

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 (the system log, elements added, etc.) of that step on the right side.

Graph
Table
Step
Kind
Elements added
Wall time
Max wait time
Input data
Output data
Output collections
Name
Elements added
Estimated size
PubsubIO.Read
ParDo(ExtractFlatten)
ParDo(Anonymous)
ParDo(Anonymous)
ParDo(Anonymous



Step 7

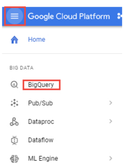
You may go back into Dataflow jobs on your GCP Console, click on your job and see how the run details have changed for the steps, for example click on **write_toBQ** and look at Elements added.

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**:

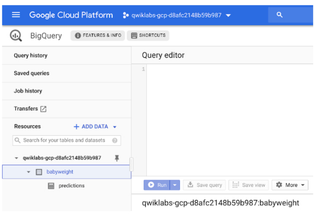


The **Welcome to BigQuery in the Cloud Console** message box opens. This message box provides a link to the quickstart guide and lists UI updates.

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.



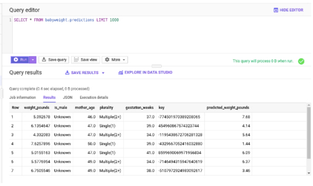
Step 9

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

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

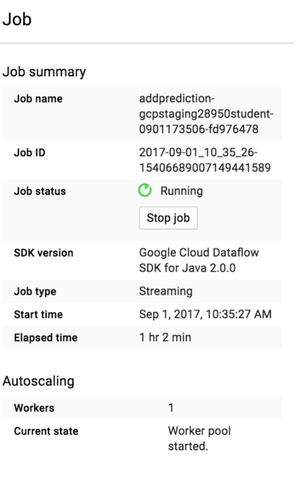
Step 10

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



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**.



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: 07-01-2019
Lab Last Tested: 03-22-2019
©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.

Chat