Overview

The purpose of this lab is to show learners how to instantiate a Jupyter notebook running on Google Cloud Platform's AI Platform service. To aid in the demonstration, a dataset with various flight departure and arrival times will be leveraged.

Objectives

In this lab, you learn to perform the following tasks:

- Instantiate a Jupyter notebook on AI Platform.
- Execute a BigQuery query from within a Jupyter notebook and process the output using Pandas.

Set up your environment

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, 

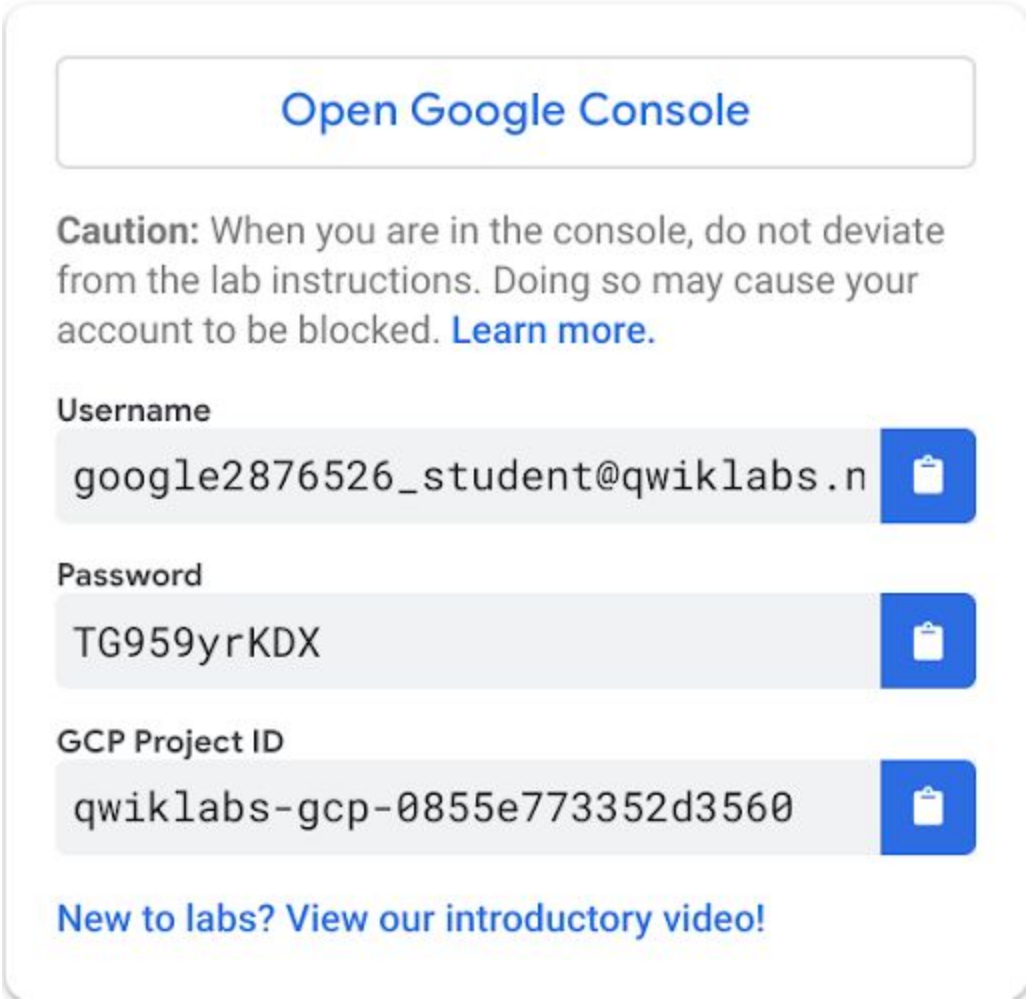
and make sure you can finish in that time block.

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 the Google Cloud Console.



5. Click **Open Google Console**.
6. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

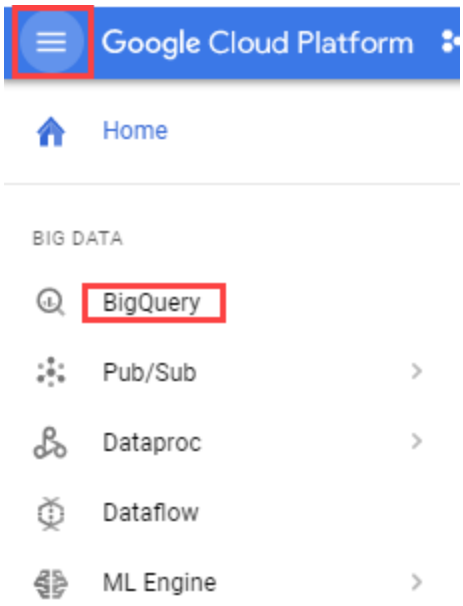
If you use other credentials, you'll get errors or **incur charges**.

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.

Open BigQuery Console

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



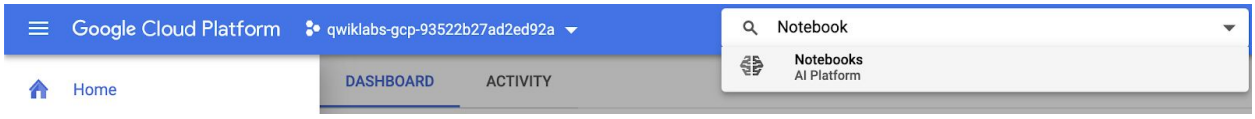
The **Welcome to BigQuery in the Cloud Console** message box opens. This message box provides a link to the quickstart guide and lists UI updates.

Click **Done**.

Start a JupyterLab Notebook Instance

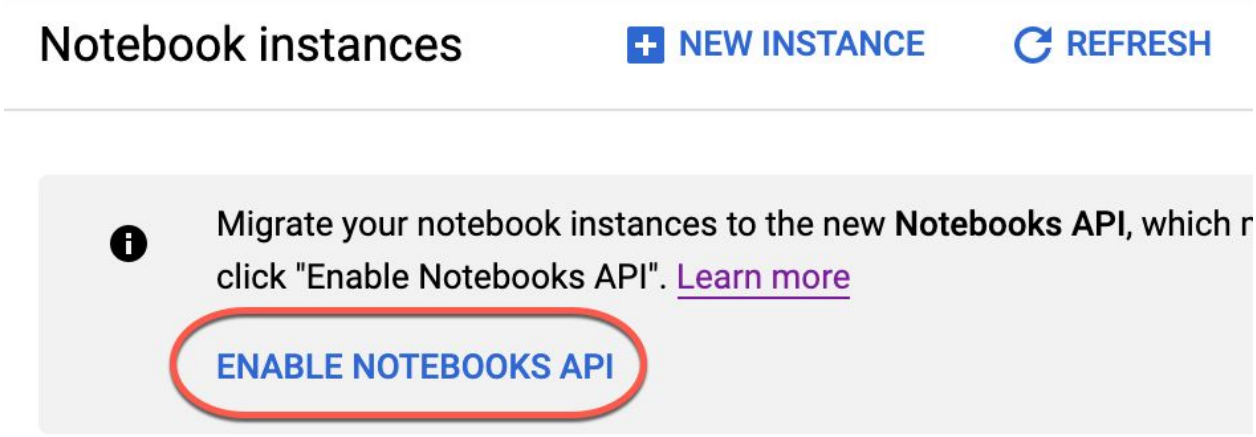
1. Navigate to the search bar in the GCP console and type in **Notebook**.

2. Select **Notebooks** for **AI Platform**.



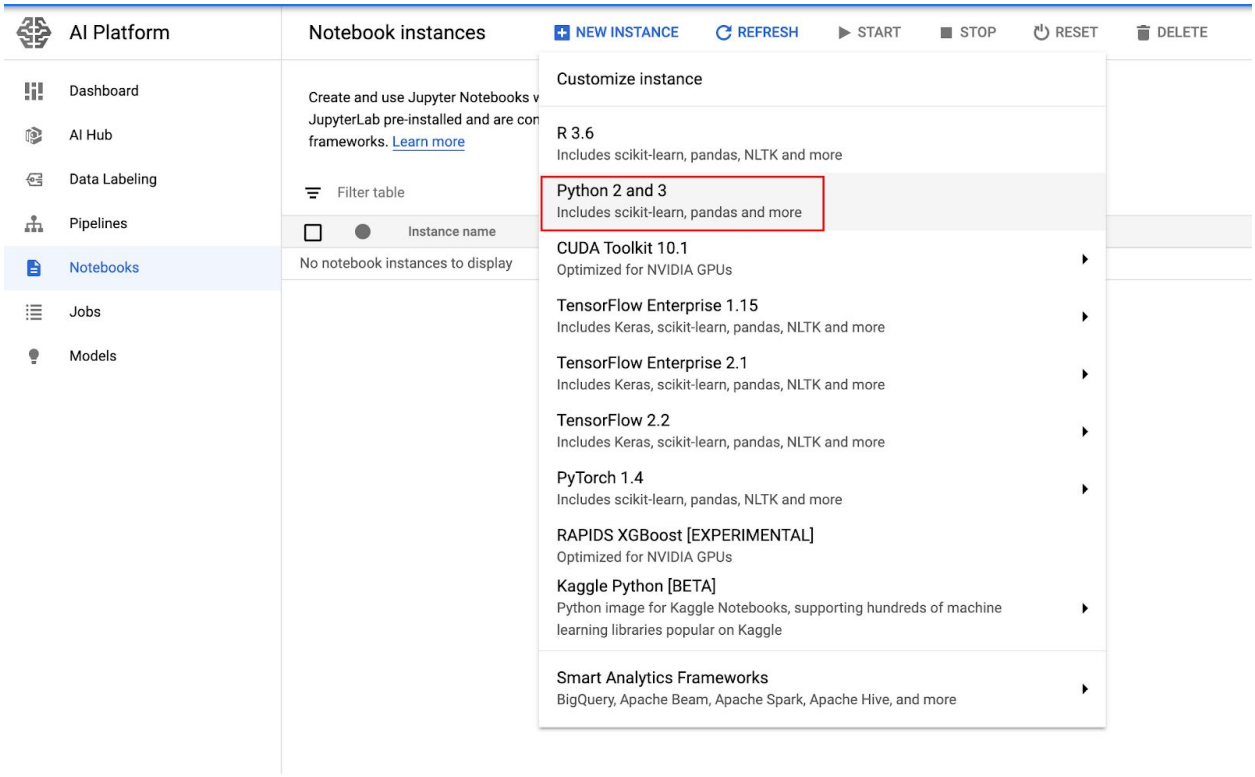
3. You'll be redirected to a page for notebook instances on AI Platform.

When the notebook main page loads if you notice a link entitled **Enable Notebooks API**, click that link to allow the background Notebooks API to be upgraded. The upgrade will occur promptly.

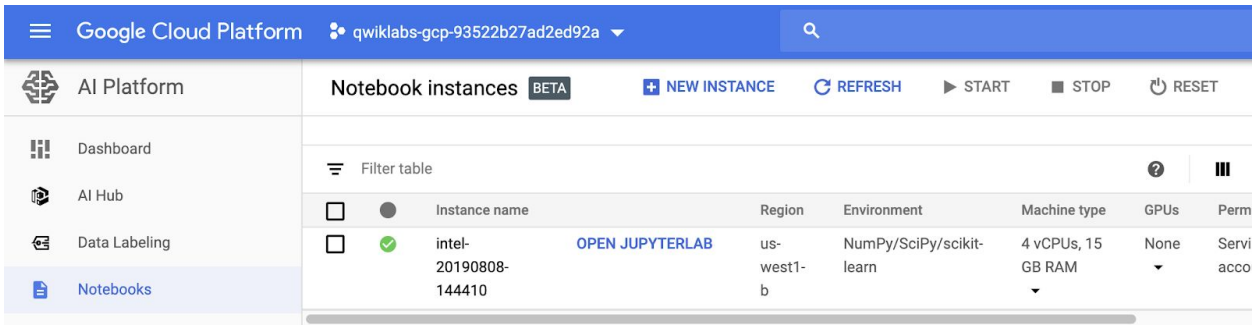


Click on the **NEW INSTANCE** icon on the top of the page.

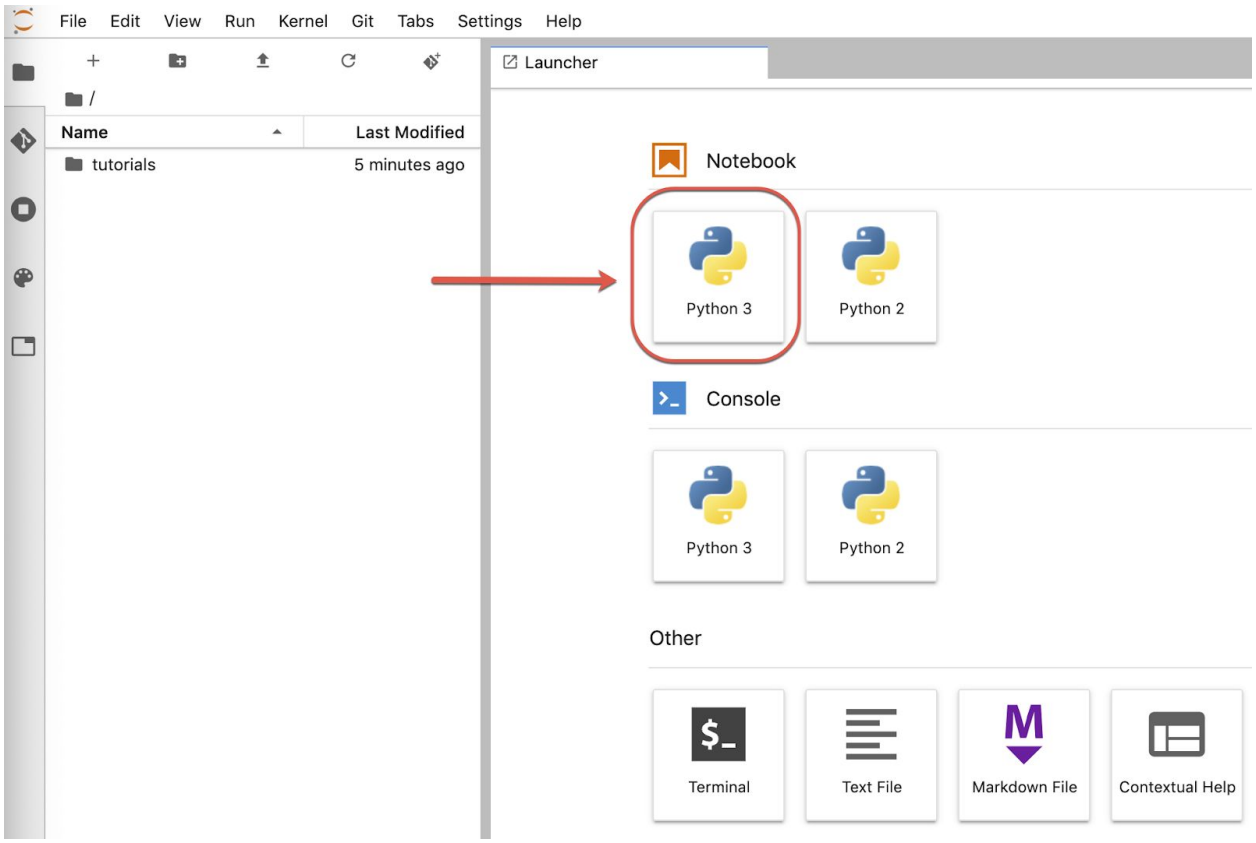
4. In the menu that pops down, select the **Python 2 and 3** option.



5. A screen titled **New notebook instance** will be shown. Leave the default options and click on **CREATE**.
6. After a few minutes, the AI Platform Notebooks console will have your instance name followed by **OPEN JUPYTERLAB**. Click **OPEN JUPYTERLAB**.



7. A new tab will open in your browser with the JupyterLab environment. Select **Python 3** under **Notebook**.



Your notebook is now set up.

Click *Check my progress* to verify the objective.

Start a JupyterLab Notebook Instance.

Check my progress

Execute a BigQuery query

1. Execute the following query in the first cell of the notebook to install the `google-cloud-bigquery` library at version `1.25.0`.

```
!pip install google-cloud-bigquery==1.25.0 --use-feature=2020-resolver
```

Note: You may ignore any error message related to **google-cloud-storage**.

Restart the kernel by clicking **Restart kernel** icon > **Restart**.

2. Enter the following query in the second cell of the notebook.

```
%%bigquery df
SELECT
  departure_delay,
  COUNT(1) AS num_flights,
  APPROX_QUANTILES(arrival_delay, 10) AS arrival_delay_deciles
FROM
  `bigquery-samples.airline_ontime_data.flights`
GROUP BY
  departure_delay
HAVING
  num_flights > 100
ORDER BY
  departure_delay ASC
```

The command makes use of the magic function `%%bigquery`. Magic functions in notebooks provide an alias for a system command. In this case, `%%bigquery` runs the query in the cell in BigQuery and stores the output in a Pandas DataFrame object named `df`.

3. Run the cell by hitting **Shift + Enter**, when the cursor is in the cell. Alternatively, if you navigate to the **Run** tab you can click on **Run Selected Cells**. Note the keyboard shortcut for this action in case it is not Shift + Enter. There should be no output when executing the command.

Click *Check my progress* to verify the objective.

Execute a BigQuery query

Check my progress

4. View the first five rows of the query's output by executing the following code in a new cell:

```
df.head()
```

```
[4]: df.head()
```

[4]:

	departure_delay	num_flights	arrival_delay_deciles
0	-37.0	107	[-66.0, -44.0, -41.0, -35.0, -30.0, -23.0, -17...
1	-36.0	139	[-74.0, -43.0, -39.0, -37.0, -32.0, -25.0, -18...
2	-35.0	191	[-68.0, -45.0, -40.0, -36.0, -28.0, -19.0, -14...
3	-34.0	195	[-58.0, -44.0, -40.0, -35.0, -30.0, -25.0, -19...
4	-33.0	227	[-59.0, -43.0, -39.0, -36.0, -32.0, -28.0, -20...

Make a Plot with Pandas

We're going to use the Pandas DataFrame containing our query output to build a plot that depicts how arrival delays correspond to departure delays. Before continuing, if you are unfamiliar with Pandas the Ten Minute Getting Started Guide is recommended reading.

1. To get a DataFrame containing the data we need we first have to wrangle the raw query output. Enter the following code in a new cell to convert the list of `arrival_delay_deciles` into a Pandas Series object. The code also renames the resulting columns.

```
import pandas as pd

percentiles = df['arrival_delay_deciles'].apply(pd.Series)
percentiles.rename(columns = lambda x : '{0}%'.format(x*10), inplace=True)
percentiles.head()
```

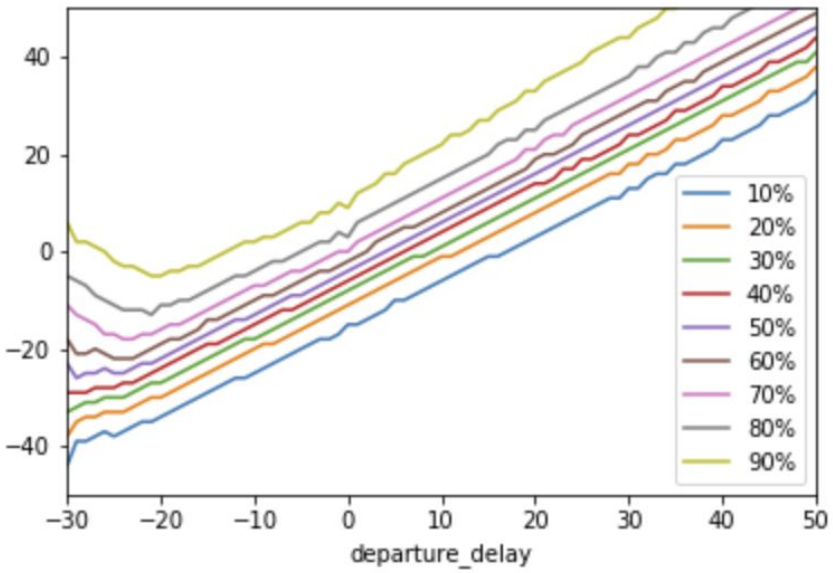
2. Since we want to relate departure delay times to arrival delay times we have to concatenate our `percentiles` table to the `departure_delay` field in our original DataFrame. Execute the following code in a new cell:

```
df = pd.concat([df['departure_delay'], percentiles], axis=1)
df.head()
```

3. Before plotting the contents of our DataFrame, we'll want to drop extreme values stored in the `0%` and `100%` fields. Execute the following code in a new cell:

```
df.drop(labels=['0%', '100%'], axis=1, inplace=True)
df.plot(x='departure_delay', xlim=(-30,50), ylim=(-50,50));
```

```
[23]: df.drop(labels=['0%', '100%'], axis=1, inplace=True)
      df.plot(x='departure_delay', xlim=(-30,50), ylim=(-50,50));
```



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.