Loading Data into Google Cloud SQL

GSP196

Overview

In this lab, you will learn how to import data from CSV text files into Cloud SQL and then carry out some basic data analysis using simple queries.

The dataset used in this lab comes from the US Bureau of Transport Statistics and contains historical information about internal flights in the United States. This dataset can be used to demonstrate a wide range of data science concepts and techniques and is used in all of the labs in the [Data Science on Google Cloud Platform](#) Quest.

Objectives

- Create Cloud SQL instance
- Create a Cloud SQL database
- Import text data into Cloud SQL
- Build an initial data model using queries

Setup and Requirements

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.

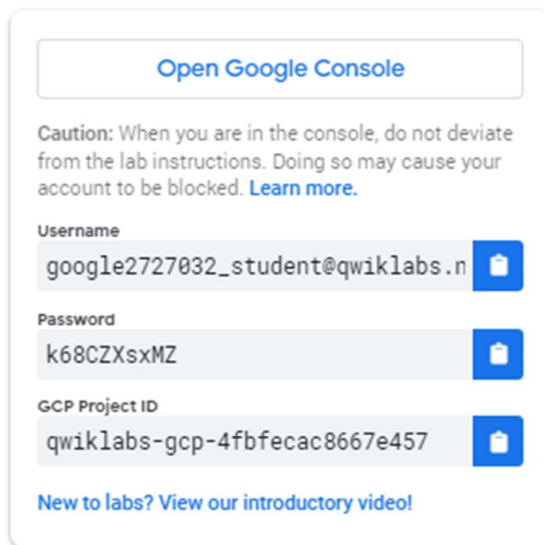**What you need**

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.

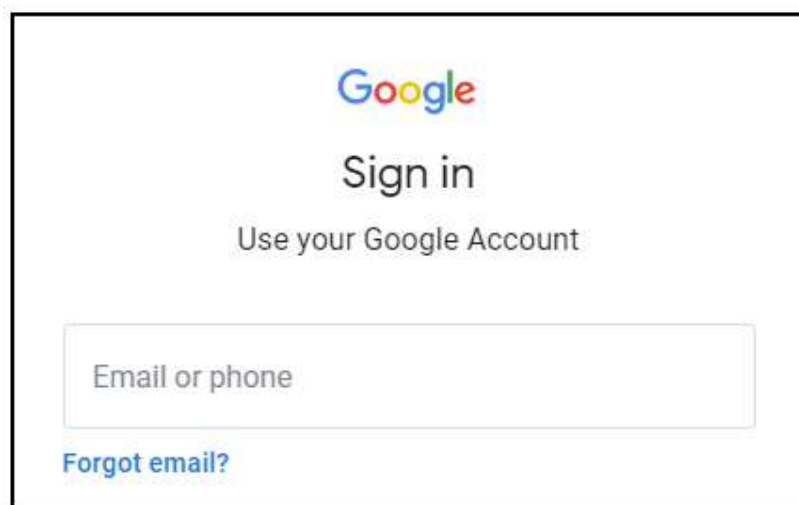 **How to start your lab and sign in to the Google Cloud Console**

  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.

  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.

  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-
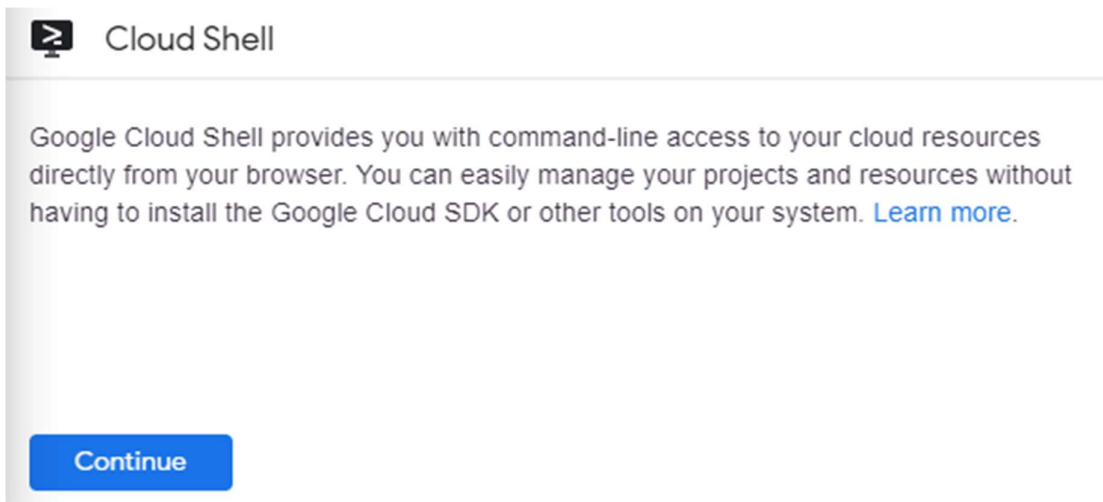left.

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.
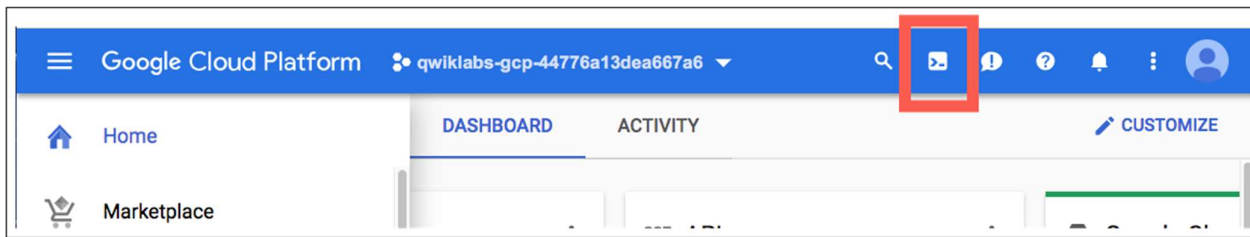
Click **Continue**.

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:
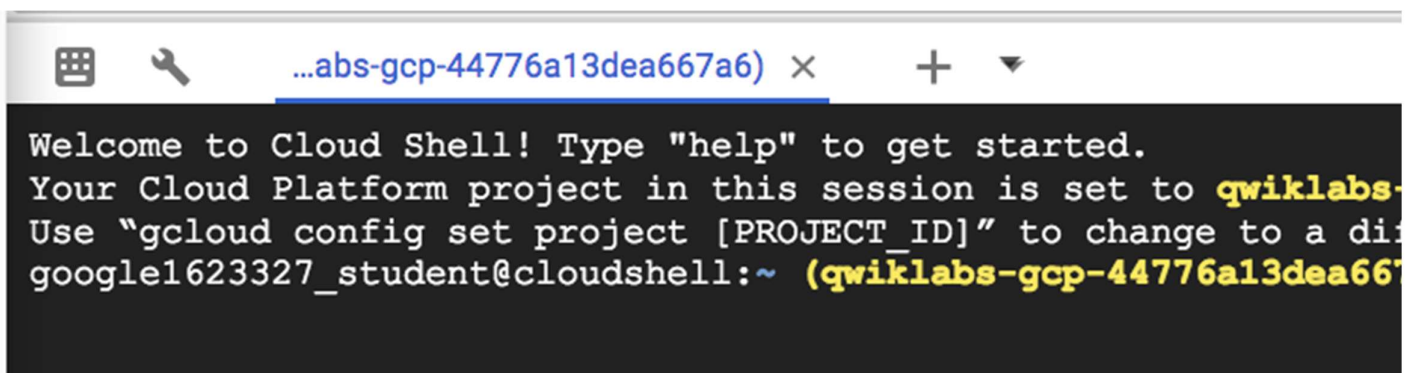
`gcloud` is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

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@qwiklabs.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](#).

Preparing your Environment

This lab uses a set of code samples and scripts developed for the *Data Science on Google Cloud Platform* book from O'Reilly Media, Inc. and specifically covers the configuration of Google Cloud SQL and importing data tasks covered in the first part of Chapter 3, "Creating Compelling Dashboards". You will clone the sample repository used in Chapter 2 from Github to the Cloud Shell and carry out all of the lab tasks from there.

Clone the Data Science on Google Cloud Repository

In Cloud Shell enter the following commands to clone the repository:

```
git clone \
    https://github.com/GoogleCloudPlatform/data-science-on-gcp/
```

Change to the repository directory:

```
cd data-science-on-gcp/03_sqlstudio
```

Create environment variables that will be used later in the lab for your project ID and the storage bucket that will contain your data:

```
export PROJECT_ID=$(gcloud info --format='value(config.project)')
export BUCKET=${PROJECT_ID}-ml
```

Create a Cloud SQL instance

Enter the following commands to create a Cloud SQL instance:

```
gcloud sql instances create flights \
    --tier=db-n1-standard-1 --activation-policy=ALWAYS
```

This will take a few minutes to complete.

Test Completed Task

Click **Check my progress** to verify your performed task. If you have successfully created a Cloud SQL instance, you will see an assessment score.

Create a Cloud SQL instance.

Check my progress

Set a root password for the Cloud SQL instance:

```
gcloud sql users set-password root --host % --instance flights \
 --password Passw0rd
```

When prompted for the password type `Passw0rd` and press enter this will update root password.

Now create an environment variable with the IP address of the Cloud Shell:

```
export ADDRESS=$(wget -qO - http://ipecho.net/plain)/32
```

Allowlist the Cloud Shell instance for management access to your SQL instance.

```
gcloud sql instances patch flights --authorized-networks $ADDRESS
```

When prompted press **Y** to accept the change.

Test Completed Task

Click **Check my progress** to verify your performed task. If you have successfully allowlisted Cloud Shell to access SQL instance, you will see an assessment score.

Allowlist the Cloud Shell instance to access your SQL instance.

Check my progress

Get the IP address of your Cloud SQL instance by running:

```
MYSQLIP=$(gcloud sql instances describe \
flights --format="value(ipAddresses.ipAddress)")
```

Check the variable MYSQLIP:

```
echo $MYSQLIP
```

you should get an IP address as an output.

Create the flights table using the `create_table.sql` file.

```
mysql --host=$MYSQLIP --user=root \
      --password --verbose < create_table.sql
```

When prompted for a password enter `Passw0rd`.

Test Completed Task

Click **Check my progress** to verify your performed task. If you have successfully created a bts database and flights table using the create_table.sql file, you will see an assessment score.

Create a bts database and flights table using the create_table.sql file.

Check my progress

Connect to the `mysql` command line interface:

```
mysql --host=$MYSQLIP --user=root  --password
```

When prompted for a password enter `Passw0rd`.

In the `mysql` command line interface check the import by entering the following commands:

```
use bts;
describe flights;
```

Query the flights table:

```
select DISTINCT(FL_DATE) from flights;
```

This will return an empty set as there is no data in the database yet.

Exit the `mysql` interactive console:

```
exit
```

Add data to Cloud SQL instance

Now you'll copy the CSV files stored on Cloud Storage locally. These are the source data files that you learned how to retrieve in the **Ingesting Data into the Cloud using Google App Engine** lab. For this lab, they have been provided. You'll only be importing two months of data, about a million records, to keep resource usage low.

Run the following:

```
counter=0
for FILE in 201501.csv 201502.csv; do
    gsutil cp gs://$BUCKET/flights/raw/$FILE \
            flights.csv-${counter}
    counter=$((counter+1))
done
```

Import the CSV file data into Cloud SQL using `mysql`:

```
mysqlimport --local --host=$MYSQLIP --user=root --password \
--ignore-lines=1 --fields-terminated-by=',' bts flights.csv-*
```

When prompted for a password enter `Passw0rd`.

Connect to the mysql interactive console:

```
mysql --host=$MYSQLIP --user=root  --password
```

When prompted for a password enter `Passw0rd`.

Build the initial data model

In the `mysql` interactive console select the database:

```
use bts;
```

Query the flights table for unique dates:

```
select DISTINCT(FL_DATE) from flights;
```

This should return 59 days.

Query the flights table for unique carrier identifiers:

```
select DISTINCT(CARRIER) from flights;
```

This should return 14 carriers.

The initial data model is based on the assertion that if a flight is greater than 15 minutes late departing it will also be greater than 15 minutes arriving. You will use queries to see how well this holds true. There are four scenarios that will be considered for this initial model:

1. Where the arrival delay is less than 15 minutes and departure delay was also less than 15 minutes. This is a true negative.
2. Where the arrival delay is greater than 15 minutes and departure delay was less than 15 minutes. This is a false negative.
3. Where the arrival delay is less than 15 minutes and departure delay was greater than 15 minutes. This is a false positive.
4. Where the arrival delay is greater than 15 minutes and the departure delay is also greater than 15 minutes. This is a true positive.

Run the following:

```
select count(dest) from flights where arr_delay < 15 and dep_delay < 15;
select count(dest) from flights where arr_delay >= 15 and dep_delay < 15;
select count(dest) from flights where arr_delay < 15 and dep_delay >= 15;
select count(dest) from flights where arr_delay >= 15 and dep_delay >= 15;
```

This will provide the following totals:

- True Negative : 672038
- False Negative: 44855
- False Positive: 35991
- True Positive: 146275

You can now use environment variables to test different values for arrival and departure delay thresholds:

```
SET @ARR_DELAY_THRESH = 15;
SET @DEP_DELAY_THRESH = 10;
# Correct - true negative
select count(dest) from flights where arr_delay < @ARR_DELAY_THRESH and dep_delay <
@DEP_DELAY_THRESH;
# False negative
select count(dest) from flights where arr_delay >= @ARR_DELAY_THRESH and dep_delay <
@DEP_DELAY_THRESH;
# False positive
select count(dest) from flights where arr_delay < @ARR_DELAY_THRESH and dep_delay >=
@DEP_DELAY_THRESH;
# True positive
select count(dest) from flights where arr_delay >= @ARR_DELAY_THRESH and dep_delay >=
@DEP_DELAY_THRESH;
```

This will provide the following totals:

- True Negative: 642461
- False Negative: 35435
- False Positive: 65568
- True Positive: 155695

Try once more with a longer delay threshold:

```
SET @ARR_DELAY_THRESH = 15;
SET @DEP_DELAY_THRESH = 20;
# Correct - true negative
select count(dest) from flights where arr_delay < @ARR_DELAY_THRESH and dep_delay <
@DEP_DELAY_THRESH;
# False negative
```

```
select count(dest) from flights where arr_delay >= @ARR_DELAY_THRESH and dep_delay <
@DEP_DELAY_THRESH;
# False positive
select count(dest) from flights where arr_delay < @ARR_DELAY_THRESH and dep_delay >=
@DEP_DELAY_THRESH;
# True positive
select count(dest) from flights where arr_delay >= @ARR_DELAY_THRESH and dep_delay >=
@DEP_DELAY_THRESH;
```

This will provide the following totals:

- True Negative: 689710
- False Negative: 56046
- False Positive: 18319
- True Positive: 135084
  Exit the 'mysql' interactive console:

```
exit
```

Test your Understanding

Below are multiple-choice questions to reinforce your understanding of this lab's concepts.
Answer them to the best of your abilities.

The Tiers resource represents a service configuration that can be used to define a Cloud SQL instance. Which type of
tier resource is used while creating a Cloud SQL instance in the lab?
db-n1-standard-1

Congratulations!

Now you know how to create tables and import text data that has been stored on Cloud Storage into Google Cloud SQL.

Finish Your Quest
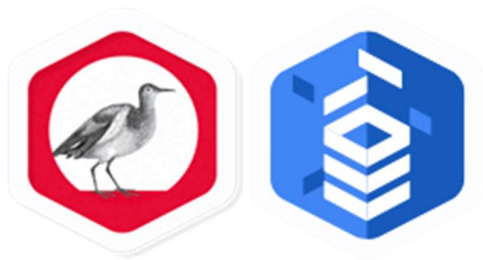
This self-paced lab is part of the Qwiklabs Quest Data Science on Google Cloud and Cloud SQL. A Quest is a series of related labs that form a learning path. Completing this Quest earns you the badge above, to recognize your achievement. You can make your badge (or badges) public and link to them in your online resume or social media account. Enroll in Data Science on Google Cloud or Cloud SQL and get immediate completion credit if you've taken this lab. See other available Qwiklabs Quests.

Take Your Next Lab

Continue your Quest with Visualizing Data with Google Data Studio, or check out these suggestions:
- Processing Data with Google Cloud Dataflow
- Visualize Real Time Geospatial Data with Google Data Studio

Next Steps / Learn More

Here are some follow-up steps:

- Data Science on the Google Cloud: O'Reilly Media, Inc.

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 July 29, 2020

Lab Last Tested October 9, 2019