

# Monitoring BigQuery Workloads | Google Cloud Skills Boost

Qwiklabs : 12-15 minutes

---

## Overview

Storing and querying massive datasets can be time consuming and expensive without the right infrastructure. BigQuery is a serverless and fully managed enterprise data warehouse that enables fast and cost-effective queries using the processing power of Google's infrastructure. In BigQuery, storage and compute resources are decoupled, which provides you the flexibility to store and query your data according to your organization's needs and requirements.

BigQuery makes it easy to estimate query resource usage and costs using a variety of tools including the BigQuery query validator in the Google Cloud console, the `dry-run` flag in the bq command-line tool, the Google Cloud Pricing Calculator, and using the API and client libraries.

In this lab, you use the BigQuery query validator and the bq command-line tool to estimate the amount of data to be processed before running a query. You also use a SQL query and the API to determine resource usage after a query has run successfully.

## Objectives

In this lab, you learn how to:

- Use the BigQuery query validator to estimate the amount of data to be processed by a query.
- Determine slot usage for executed queries using a SQL query and the API.
- Complete a dry run of a query to estimate the amount of data to be processed by a query.

## Setup and Requirements

### Qwiklabs setup

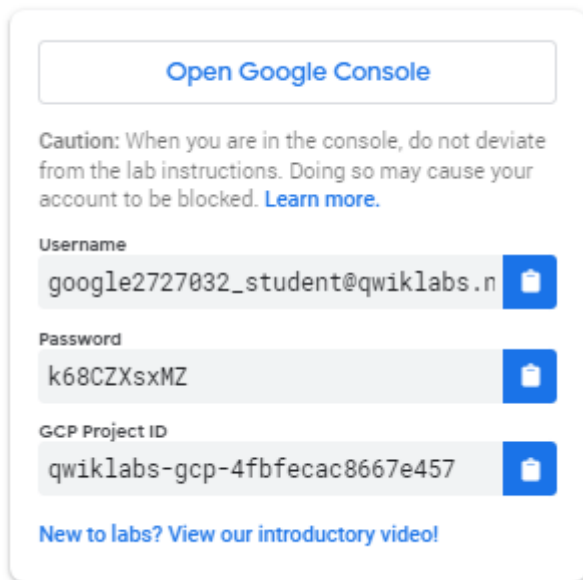
For each lab, you get a new Google Cloud project and set of resources for a fixed time at no cost.

1. Sign in to Qwiklabs using an incognito window.
2. Note the lab's access time (for example, 1:15:00), and make sure you can finish within that time.  
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 (**Username** and **Password**). 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 receive errors or **incur charges**.
7. Accept the terms and skip the recovery resource page.

### How to start your lab and sign in to the 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.

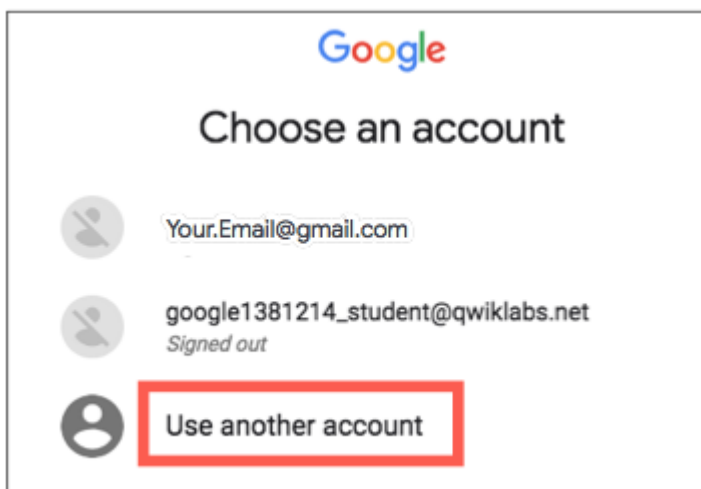


A screenshot of a web panel titled "Open Google Console". At the top is a button labeled "Open Google Console". Below it is a caution message: "Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. [Learn more.](#)". The panel contains three input fields, each with a copy icon to its right: "Username" with the value "google2727032\_student@qwiklabs.n", "Password" with the value "k68CZXsxMZ", and "GCP Project ID" with the value "qwiklabs-gcp-4fbfecac8667e457". At the bottom is a link: "New to labs? View our introductory video!"

2. Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Choose an account** page.

**Note:** Open the tabs in separate windows, side-by-side.

3. On the Choose an account page, click **Use Another Account**. The Sign in page opens.



A screenshot of the "Choose an account" page from Google. It features the Google logo at the top. Below the title, there are three options, each with a circular icon to its left: "Your.Email@gmail.com", "google1381214\_student@qwiklabs.net Signed out", and "Use another account". The "Use another account" option is highlighted with a red rectangular border.

4. Paste the username that you copied from the Connection Details panel. Then copy and paste the password.

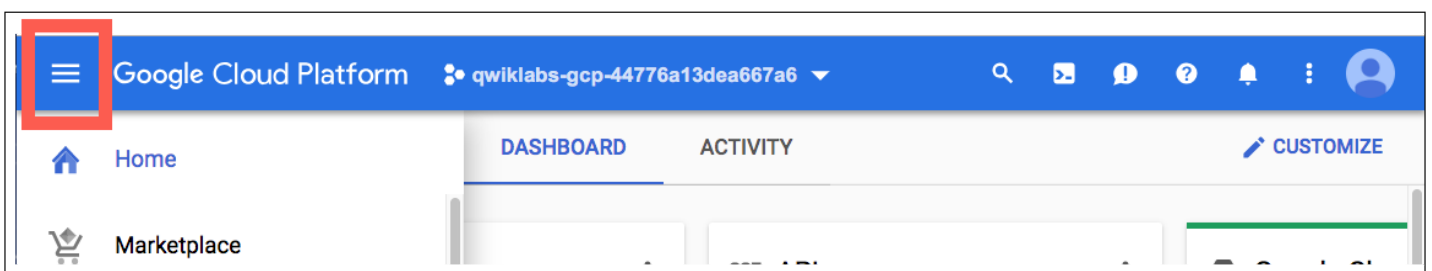
**Note:** You must use the credentials from the Connection Details panel. Do not use your Google Cloud Skills Boost credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

5. 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.

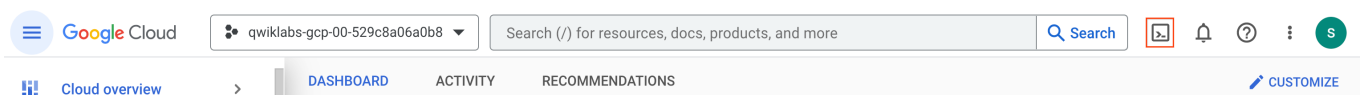


## Activate Google Cloud Shell

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.

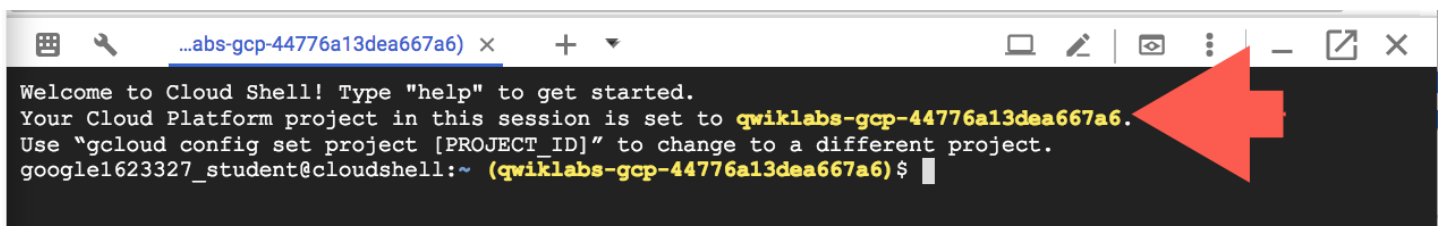
Google Cloud Shell provides command-line access to your Google Cloud resources.

1. In Cloud 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. 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: - @.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 =

**Example output:**

[core] project = qwiklabs-gcp-44776a13dea667a6 **Note:** Full documentation of **gcloud** is available in the [gcloud CLI overview guide](#) .

## Task 1. Use the query validator to estimate the amount of data to be processed

When you enter a query in the Google Cloud console, the BigQuery query validator verifies the query syntax and provides an estimate of the number of bytes to be processed by the query.

In this task, you query a public dataset (New York Citi Bikes) maintained by the BigQuery public datasets program. Using this dataset, you learn how to use the query validator to validate a SQL query and to estimate the amount of data to be processed by a query *before* you run it.

1. In the Google Cloud console, in the **Navigation menu** (≡), under Analytics, click **BigQuery**.

The Welcome to BigQuery in the Cloud Console message box opens. This message box provides a link to the quickstart guide and the release notes.

2. Click **Done**.
3. In the SQL Workspace toolbar, click the **Editor** tab to open the SQL query editor.



4. In the **BigQuery query editor**, paste the following query but **do not** run the query:

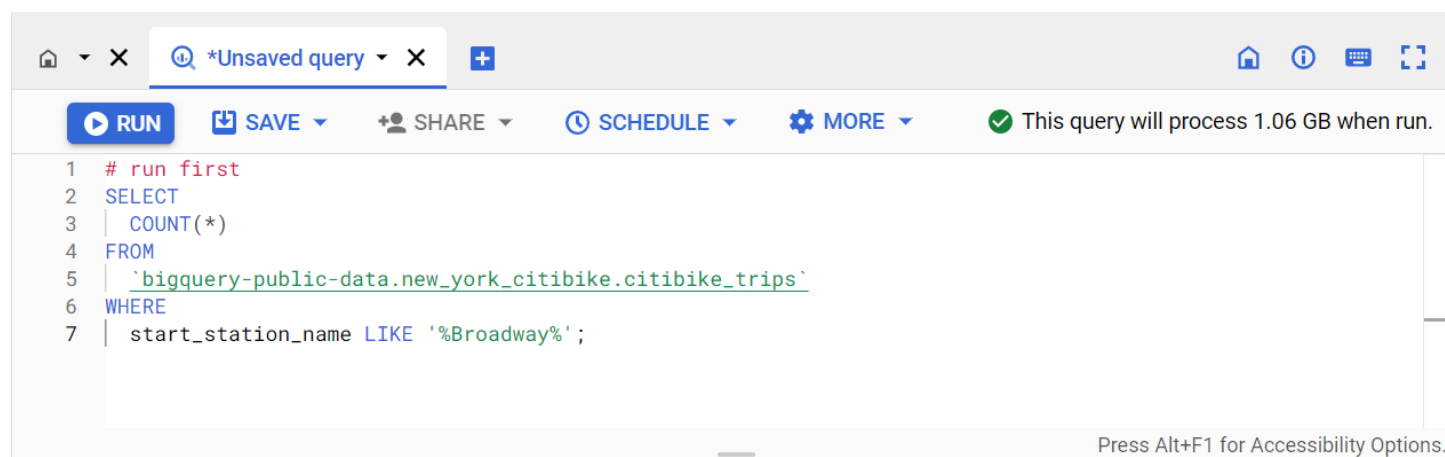
```
SELECT COUNT(*) FROM `bigquery-public-data.new_york_citibike.citibike_trips` WHERE  
start_station_name LIKE '%Broadway%';
```

When executed, this query returns the count of the station names that contain the text "Broadway" in the column named **start\_station\_name** in the **citibike\_trips** table.

5. In the query editor toolbar, notice the circular check icon, which activates the query validator and confirms that the query is valid.

BigQuery will automatically run the query validator when you add or modify the code in the query editor.

A green or red check displays above the query editor depending on whether the query is valid or invalid. If the query is valid, the validator also displays the amount of data to be processed if you choose to run the query.





According to the query validator, this query will process 1.06 GB when run.


6. Click **Run**.

The query returns the number of records (5,414,611) that contain the text "Broadway" in the column named **start\_station\_name**.

Query results

 SAVE RESULTS

 EXPLORE DATA



JOB INFORMATION

RESULTS

JSON

EXECUTION DETAILS

EXECUTION GRAPH

PREVIEW

Row	f0_	
1	5414611	

Click **Check my progress** to verify the objective. Estimate the amount of data processed by a query

## Task 2. Determine slot usage using a SQL query

BigQuery uses **slots** (virtual CPUs) to execute SQL queries, and it automatically calculates how many slots each query requires, depending on query size and complexity. After you run a query in the Google Cloud console, you receive both the results and a summary of the amount of resources that were used to execute the query.

In this task, you identify the job ID of the query executed in the previous task and use it in a new SQL query to retrieve additional information about the query job.

1. In the Query results, click on the **Job information** tab.

Query results	
JOB INFORMATION	RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW
Job ID	qwiklabs-gcp-01-5f4dee7a15a3:US.bquxjob_403a14df_185dd37737a
User	student-01-d98c86c30592@qwiklabs.net
Location	US
Creation time	Jan 23, 2023, 11:31:17 AM UTC+5:30
Start time	Jan 23, 2023, 11:31:17 AM UTC+5:30
End time	Jan 23, 2023, 11:31:17 AM UTC+5:30
Duration	0 sec
Bytes processed	1.06 GB
Bytes billed	1.06 GB
Job priority	INTERACTIVE
Use legacy SQL	false
Destination table	<a href="#">Temporary table</a>

2. Identify the line for the **Job ID**, and use the provided value to select the project ID and job ID.

For example, the value `qwiklabs-gcp-01-5f4dee7a15a3:US.bquxjob_403a14df_185dd37737a` begins with the project ID, followed by the location where the job was executed, and ends with the job ID. The syntax for `:US.` identifies the location where the job was executed.

The project ID is the first part `qwiklabs-gcp-01-5f4dee7a15a3` (before `:US.`), while the job ID is the last part `bquxjob_403a14df_185dd37737a` (after `:US.`).

**Note:** You can copy the full value to a text editor or document to make it easier to select the project ID and the job ID.

## Query results

JOB INFORMATION	RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW
Job ID	qwiklabs-gcp-01-5f4dee7a15a3:US.bquxjob_403a14df_185dd37737a

3. In the query editor, copy and paste the following query, replacing '`YOUR_ID`' with your job ID (such as '`bquxjob_403a14df_185dd37737a`')

```
SELECT query, reservation_id, CONCAT('*****@', REGEXP_EXTRACT(user_email, r'@(.+)')) AS
user_email, total_bytes_processed, total_slot_ms, job_stages FROM `region-
us`.INFORMATION_SCHEMA.JOBS_BY_PROJECT WHERE job_id = 'YOUR_ID';
```

```

1 # run second - grab ID from first query
2 SELECT
3   query,
4   reservation_id,
5   CONCAT('*****@', REGEXP_EXTRACT(user_email, r'@(.+)')) as user_email,
6   total_bytes_processed,
7   total_slot_ms,
8   job_stages
9 FROM
10  `region-us`.INFORMATION_SCHEMA.JOBS_BY_PROJECT
11 WHERE
12  job_id = 'bquxjob_403a14df_185dd37737a' ;|

```

Press Alt+F1 for Accessibility Options

When executed, this query returns the slot usage of the query job previously executed on the Citi Bikes public dataset.

4. Click **Run**.

The output of this query provides a table that shows the query stages and the associated slot usage for each stage.

Since an individual task in a query is executed by one slot, the sum of values in the column named **job\_stages.completed\_parallel\_inputs** is the number of total slots used to run the query.

However, after a single slot has completed the first task assigned to it, it can be reassigned to complete another task.

So understanding the total slot time used to run the query (value provided in the column named **total\_slot\_ms**) is also important. Specifically, the slot time (in milliseconds, or ms) is provided for the entire query job and for each stage of the query, which represents the amount of slot time used to complete that stage.

For example, a query may complete 150 tasks, but if each task executes quickly, the query may actually use a lower number of slots, such as 100, rather than 150 slots.

Query results											
JOB INFORMATION		RESULTS		JSON	EXECUTION DETAILS		EXECUTION GRAPH		PREVIEW		
Row	shuffler_output	shuffler_output	records_read	records_written	parallel_inputs	job_stages_completed_parallel_inputs	job_stages.status	job_stages.steps.kind	job_stages.steps.substeps	job_...	slot_ms
1	5	531	0	58937715	59	59	COMPLETE	READ	\$1:start_station_name		2219
3	9	0	59	1	1	1	COMPLETE	READ	FROM bigquery-public-data.ne...		12
									WHERE like(\$1, '%Broadway%')		
									\$20 := COUNT_STAR()		
									TO __stage00_output		
3	9	0	59	1	1	1	COMPLETE	READ	\$20		12
									FROM __stage00_output		
									\$10 := SUM_OF_COUNTS(\$20)		
									TO __stage01_output		

Click **Check my progress** to verify the objective. Determine slot usage using SQL query

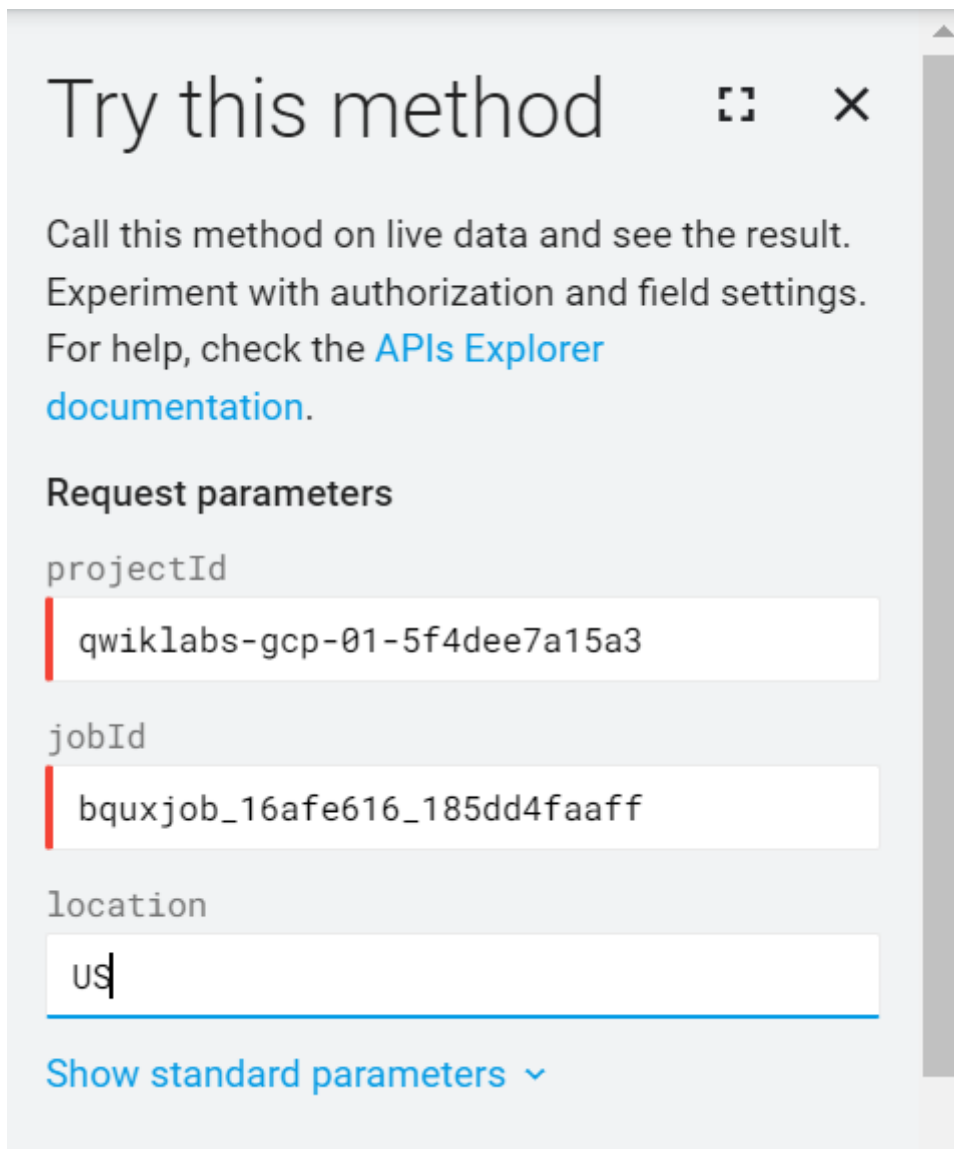
### Task 3. Determine slot usage using an API call

You can also retrieve information about a specific query job using the API. In BigQuery, you can use the API directly by making requests to the server, or you can use client libraries in your preferred language: C#, Go, Java, Node.js, PHP, Python, or Ruby.

In this task, you use the Google APIs Explorer to test the BigQuery API and retrieve the slot usage for the query that you ran in a previous task.

1. In a new Incognito browser tab, navigate to the BigQuery API page for the [jobs.get](#) method.
2. In the **Try this method** window, input your **project ID** and **job ID** that you identified in the previous task.

For example, `qwiklabs-gcp-01-5f4dee7a15a3` for the project ID and `bquxjob_403a14df_185dd37737a` for the job ID.



Try this method

Call this method on live data and see the result.  
Experiment with authorization and field settings.  
For help, check the [APIs Explorer documentation](#).

**Request parameters**

projectId  
qwiklabs-gcp-01-5f4dee7a15a3

jobId  
bquxjob\_16afe616\_185dd4faaff

location  
us

[Show standard parameters](#) ▾

3. Click **Execute**.

If asked to confirm your login, select the student username that you used to login to Google Cloud for the previous tasks:

4. Review the API response for each stage and for the entire query job.



To see the value for completed parallel inputs for the first stage, scroll down to **statistics > query > queryPlan > name: S00 > completedParallelInputs**.

200 X

```
    "readRatioMax": 0.25,
    "readMsMax": "231",
    "computeRatioAvg": 0.016666666666666666,
    "computeMsAvg": "11",
    "computeRatioMax": 0.25,
    "computeMsMax": "165",
    "writeRatioAvg": 0.0030303030303030303,
    "writeMsAvg": "2",
    "writeRatioMax": 0.024242424242424242,
    "writeMsMax": "16",
    "shuffleOutputBytes": "9743",
    "shuffleOutputBytesSpilled": "0",
    "recordsRead": "2",
    "recordsWritten": "2",
    "parallelInputs": "1014",
    "completedParallelInputs": "1014",
    "status": "COMPLETE",
    "steps": [
      {
```

To see the total slots used for the entire query job, scroll down to the end of the results to review the value for **totalSlotMs**.



```
200 X
{
  "projectId": "bigquery-public-data",
  "datasetId": "new_york_citibike",
  "tableId": "citibike_trips"
},
{
  "statementType": "SELECT",
  "transferredBytes": "0"
},
{
  "totalSlotMs": "2833",
  "finalExecutionDurationMs": "148"
},
{
  "status": {
    "state": "DONE"
  },
  "principal_subject": "user:student-02-e0a7edk"
}
```

## Task 4. Complete a dry run of a query to estimate the amount of data processed

In the `bq` command-line tool, you can use the `--dry_run` flag to estimate the number of bytes read by the query *before* you run the query. You can also use the `dryRun` parameter when submitting a query job using the API or client libraries. Dry runs of queries do not use query slots, and you are not charged for performing a dry run.

In this task, you learn how to complete a dry run of a query using the `bq` command-line tool in Cloud Shell.

1. In Cloud Shell, run the following command:

```
bq query \ --use_legacy_sql=false \ --dry_run \ 'SELECT COUNT(*) FROM `bigquery-public-data`.new_york_citibike.citibike_trips WHERE start_station_name LIKE "%Lexington%"'
```

The output shows you the estimated amount of bytes to be processed by the query *before* you run the query to retrieve any results.

Query successfully validated. Assuming the tables are not modified, running this query will process 1135353688 bytes of data.

Now that you know how many bytes will be processed by the query, you have the information needed to decide on your next steps for your workflow.

Click **Check my progress** to verify the objective. Complete a dry run of a query

## End your lab

When you have completed your lab, click **End Lab**. Google Cloud Skills Boost 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.

Copyright 2022 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.