Lab Overview: Getting Started with BigQuery

Overview

In this lab you stream log data from the Guestbook application to Cloud Logging. Using BigQuery, you analyze and query the log data.

Duration

The timing of this lab is as follows:

Component	Timing
Introduction	5 minutes
Lab	15 minutes
Total	**20 minutes**

What you need

To complete this lab, you need:

- The Google Cloud SDK installed and configured on your labs instance
- A Google Cloud project and project ID

What you will learn

In this lab, you:

- Set up Google Cloud Logging to enable streaming of log data from the Guestbook application
- Use the BigQuery Web console to view and query the log data

---

Python Lab: Getting Started with BigQuery

Overview

In this lab you stream log data from the Guestbook application to Cloud Logging. Using BigQuery, you analyze and query the log data.

Clone the project

The instructions in this section repeat the steps in one of the earlier labs. You do not need to follow these steps if you completed Lab 8: Getting Started with App Engine. If you have already cloned the Git repository used in Lab 8, skip to the next section.

When you created your Compute Engine labs instance, you installed Git on the machine. In this section of the lab, you use Git to clone the repository containing your application.

To clone the project:

Step	Action
1	Access the Google Developers Console by typing the following URL in your browser: https://console.developers.google.com
2	In the navigator pane, click the **Gallery** icon (to the left of Google Developers Console at the top of the page).
3	Click **Compute Engine > VM instances**.
4	To the right of the **cp100-labs** instance, in the **Connect** column, click **SSH**.

5	Type the following command to clone the code repository.
	
	`git clone \`
	`https://github.com/GoogleCloudPlatformTraining/`
	`cp100-appengine-memcache-python.git`

12	Leave the SSH connection open.

Deploy the app

To deploy and test the Guestbook application:

Step	Action
1	If you skipped the previous section: ● Access the Google Developers Console. ● In the navigator pane, click **Compute** to expand the section. ● Click **Compute Engine > VM instances**. ● To the right of the **cp100-labs** instance, in the **Connect** column, click **SSH**.
2	In the SSH window, type the following command to change to the cp100-appengine-memcache-python directory. **`cd cp100-appengine-memcache-python`**
3	To deploy the application to production, type the following command and replace <Project ID> with the ID of your cp100 project. You can view your Project ID in the title bar of the SSH window (just after /projects/). **`appcfg.py -A <Project ID> update . --noauth_local_webserver`**
4	When the application is successfully deployed, open your browser, type the following URL, and replace <Project ID> with the ID of your cp100 project. **http://<Project ID>.appspot.com**
5	You should see the Guestbook application loaded in the browser. Leave the Guestbook tab open.
6	Switch to your SSH window and type **`exit`** to close it.

Setup Cloud Logging

To set up streaming of application log data:

Step	Action

1	Create several entries in the Guestbook UI.
2	In the Google Developers Console, click **Gallery > Logging**. Note that by default, App Engine logs are selected in the drop-down list. You can see the application logs in the window below the toolbar.
3	At the top of the page, click the **Exports** tab.
4	In the **Export These Sources** section, click **+ Add item**. This should automatically add an entry for appengine.googleapis.com/request_log.
5	In the **Select export destinations** section, for **Stream to BigQuery dataset**, choose **Add new dataset**.
6	In the 'Create Logs Dataset' dialog, in the **Name** field, type **Logs** and then click **Create**.
7	Click **Save**.

Use BigQuery

To view and query log data using the BigQuery web console:

Step	Action
1	In the Google Developers Console, click **Gallery > BigQuery** or click this link: BigQuery Web Console. The BigQuery console opens in a separate window.
2	In the navigator, you should see a **Logs** dataset under your cp100 project.
3	Switch to the Guestbook UI and create several entries.
4	Switch to the BigQuery Web console and refresh the page.
5	Click **Logs**. You should see a request log entry below Logs similar to: appengine_googleapis_com_request_log_<YYYYMMDD>.
6	Click

	appengine_googleapis_com_request_log_<YYYYMMDD> and review the schema for the log data you are streaming to Big Query. In particular, note the following attributes that may be useful in queries: `protoPayload.resource` - The resource being invoked `protoPayload.status` - The HTTP status code `protoPayload.userAgent` - The browser user agent string `metadata.timestamp` - The date/time stamp of the request `protoPayload.latency` - The network latency
7	In the top, right corner of the window, click **Query Table**.
8	In the **New Query** window, delete the existing query and type the following query to view the top 10 requests ordered by latency. Replace <YYYYMMDD> (including the brackets) with the date appended to your request log; for example, 20150814. **`SELECT protoPayload.resource, protoPayload.latency FROM FLATTEN([Logs.appengine_googleapis_com_request_log_<YYYYMMDD>], labels.value) ORDER BY protoPayload.latency DESC LIMIT 10`**
9	Click **RUN QUERY**. The results will appear in the window below.
10	In the **New Query** window, type the following query to view the count of HTTP status codes. Replace <YYYYMMDD> (including the brackets) with the date appended to your request log; for example, 20150814. **`SELECT protoPayload.status, count(protoPayload.status) AS Count FROM FLATTEN([Logs.appengine_googleapis_com_request_log_<YYYYMMDD>], labels.value) GROUP BY protoPayload.status`**
11	Click **RUN QUERY**. The results will appear in the

	window below.

Clean up

Google Cloud Logging will charge you for streaming your log data to BigQuery, so we recommend that you disable the streaming to BigQuery once you are done with this lab. To remove the resources used in the lab:

Step	Action
1	In the Google Developers Console, click **Gallery > Logging**.
2	Click the **Exports** tab.
3	In the **Select export destinations** section, for **Stream to BigQuery dataset**, choose **No existing datasets**. This stops log streaming to BigQuery.
4	Click **Save**.
