Configuring and Using Cloud Logging and Cloud Monitoring

### **Objectives**

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

* View logs using a variety of filtering mechanisms.
* Exclude log entries and disable log ingestion.
* Export logs and run reports against exported logs.
* Create and report on logging metrics.
* Use Cloud Monitoring to monitor different Google Cloud projects.
* Create a metrics dashboard.

## Task 1. Set up resources in your first project

In this task, you create the Google Cloud resources for the first project.

In the Qwiklabs Connection Details section, you will see two projects listed. The first project will contain active Google Cloud resources, which will generate logs and monitoring metric data.

The second project will contain your Monitoring account configuration data.

**Make sure that you are working on project 1 for this task!**

1. Activate the Cloud Shell.
2. In the Cloud Shell, download and unpack an archive that contains setup code:

curl https://storage.googleapis.com/cloud-training/gcpsec/labs/stackdriver-lab.tgz | tar -zxf -

Copied!

content\_copy

cd stackdriver-lab

Copied!

content\_copy

1. Click on the **Open Editor** icon in the top-right corner of your Cloud Shell session.
2. Click **Open in a new window** if prompted.
3. Open the **stackdriver-lab** folder and select the **linux\_startup.sh** file.
4. Replace the # install logging agent and # install monitoring agent sections with the following:

# install logging agent

curl -sSO https://dl.google.com/cloudagents/add-logging-agent-repo.sh

sudo bash add-logging-agent-repo.sh --also-install

# install monitoring agent

curl -sSO https://dl.google.com/cloudagents/add-monitoring-agent-repo.sh

sudo bash add-monitoring-agent-repo.sh --also-install

Copied!

content\_copy

1. After pasting, make sure that your lines of code are properly indented.
2. Save your file.
3. Now open the setup.sh file.
4. Update the image version in # create vms section for windows-server (row 17) after --image with the following:

windows-server-2016-dc-core-v20210511

Copied!

content\_copy

1. After pasting, make sure that your lines of code are properly indented.
2. Save your file.
3. Click **Open Terminal** in the top-right corner.
4. Now run the following command:

./setup.sh

Copied!

content\_copy

The created resources will include:

* Service accounts (for use by VMs).
* Role assignments (granting service accounts permissions to write to Monitoring).
* A Linux VM with Apache and the Monitoring agents installed.
* A Windows VM with the Monitoring and Logging agents installed.
* A Google Kubernetes Engine cluster with an Nginx deployment.
* A Pub/Sub Topic and Subscription.

1. Once the setup script is done, run the activity generation script to create a background load on your Web servers and Pub/Sub topic:

./activity.sh

Copied!

content\_copy

**Note:**This command will take about 5 to 10 minutes to complete. Wait for it to complete before continuing. You can safely ignore any warnings.

Click Check my progress to verify the objective.

Set up resources in the first project

Check my progress

## Task 2. View and filter logs in first project

In this task, you view VM instance logs with simple filtering.

### **See which services are writing logs**

1. Ensure that you are on the Google Cloud Console homepage.
2. Verify you are still working in project 1; the project ID in the Console's info panel should match **GCP Project ID 1** in your lab's connection details panel.
3. View Cloud Logging by opening **Navigation menu > Logging > Logs Explorer**.
4. On the left-hand panel, **Log fields** will be visible. Under **Resource Type**, you will see several Google Cloud services that are creating logs.

All of these services are writing log entries. Entries from all these logs appear on the right, in the Query results pane. You can also query for results from specific logs, or that match specific criteria.

### **View VM instance logs with simple filtering**

1. In the **Log fields** panel, under **Resource Type**, click **VM Instance**.

After you click this:

* The contents of the Log fields panel changes. You will see new field named INSTANCE ID. It shows all the instance IDs of the VM instances that are writing log entries.
* The Query box near the top of the page is populated with resource.type="gce\_instance". This means that only entries from VM instances will be logged and displayed.
* The Query results pane also updates automatically—entries from VM Instances are the only logs displayed.

1. In the **Instance Id** field, select one of the instance IDs. Logs for the associated VM instance appear in the Query results pane.
2. Click inside the **Query** box. This now becomes editable.
3. In the **Query** box, remove everything after line 1. You should see only line 1, which contains resource.type="gce\_instance".
4. Click **Run query** (located in the top-right corner). In the Query results, you should see entries from all VM instance logs.
5. Note that the logs panel reverts to its previous state.
6. Turn on streaming logs by clicking **Stream logs** (top-right corner, next to the "Run query" button).
7. You should see new log entries showing up every 1-2 seconds as the background activity is generating unauthorized requests against your Web servers.

**Note:**You might need to click **Restart streaming** if it pauses.

You will now view overall web activity on any Linux Apache server.

1. Stop log streaming by clicking on **Stop stream** in the top-right corner.
2. Switch to viewing just the Apache access logs by selecting the **apache-access** log name (located in the log fields panel). You will see entries that show requests to the Linux Apache server.

**Note:**If you do not see an apache-access item, then click **Show More**.

You will now learn how to view general system activity on a given Linux server.

1. Under Log Name, click on the **Clear** button next to Apache access.
2. Now click on the **syslog** log name. Entries from syslog appear in the Query results pane.

**Note:**You can also control log entry displays by selecting the log severity and time windows.

## Task 3. Use log exports

In this task, you configure and test log exports to BigQuery.

Cloud Logging retains log entries for 30 days. In most circumstances, you'll want to retain some log entries for an extended time (and possibly perform sophisticated reporting on the archived logs).

Google Cloud provides a mechanism to have all log entries ingested into Cloud Monitoring also written to one or more archival sinks.

### **Configure the export to BigQuery**

1. Go to Cloud Logging Exports (**Navigation menu > Logging > Log Router**).
2. Click **Create Sink**.
3. For the **Sink name**, enter vm\_logs and then click **Next**.
4. For **Select sink service**, select **BigQuery dataset**.
5. For **Select BigQuery dataset**, select **Create new BigQuery dataset**.
6. For the **Dataset ID**, enter project\_logs, and click **Create Dataset**.
7. Click **Next**.
8. In the **Build inclusion filter** list box, enter resource.type="gce\_instance".
9. Click **Create Sink**. You will now return to a Log Router Create log sink next steps page (a message at the top may appear that says "Your log sink was successfully created. Data should be available soon.")

**Note:**You could also export log entries to Pub/Sub or Cloud Storage.

Exporting to Pub/Sub can be useful if you want to flow through an ETL process prior to storing in a database (Monitoring **>** Pub/Sub **>** Dataflow **>** BigQuery/Bigtable).

Exporting to Cloud Storage will batch up entries and write them into Cloud Storage objects approximately every hour.

You will now create an export for the **Cloud HTTP Load Balancing** logs to BigQuery.

1. From the left-hand navigation menu, select **Log Router** to return to the service homepage.
2. Click **Create Sink**.
3. For the **Sink name**, enter load\_bal\_logs and then click **Next**.
4. For **Select sink service**, select **BigQuery dataset**.
5. For **Select BigQuery dataset**, select **project\_logs**. (You created this BigQuery dataset in the previous set of steps.)
6. Click **Next**.
7. In the **Build inclusion filter** list box, enter resource.type="http\_load\_balancer".
8. Click **Create Sink**.
9. You will now be on the Create log sink next steps page for the log sink.
10. From the left-hand navigation menu, select **Log Router** to return to the service homepage.

The Log Router page appears, displaying a list of sinks (including the one you just created—load\_bal\_logs).

### **Investigate the exported log entries**

1. Open BigQuery (**Navigation menu > BigQuery**).
2. 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.
3. Click **Done**.
4. In the left pane in the Explorer section, click your project (this starts with qwiklabs-gcp-xxx) and you should see a project\_logs dataset under it.

You will now verify that the BigQuery dataset has appropriate permissions to allow the export writer to store log entries.

1. Click on the three dotted menu item ("View actions") next to the project\_logs dataset and click **Open**.
2. Then from the top-right hand corner of the Console, click the **Sharing** dropdown and select **Permissions**.
3. On the Dataset permission page, you will see that your service accounts have the "BigQuery Data Editor" role.
4. Close the dataset permissions panel.
5. Expand the project\_logs dataset to see the tables with your exported logs—you should see multiple tables (one for each type of log that's receiving entries).
6. Click on the **syslog\_(1)** table, then click **Details** to see the number of rows and other metadata. If the **syslog\_(1)** table is not visible, try refreshing the browser.
7. In **Details** tab, you will see that the full table name is in the format **syslog\_xxxxxxxx**, copy this table name.

**Note:**Because the log entries are being streamed into BigQuery as they arrive to Cloud Monitoring, they are stored in a BigQuery streaming buffer. Roughly 24 hours after arriving in the buffer, they will be moved into regular BigQuery storage. You can perform queries against the table and both the data in regular storage and the buffer will be scanned.

1. You can run all sorts of queries to analyze your archived log entries. For example, to see a subset of your tables fields, paste the below query in the query **Editor** tab (replacing syslog\_xxxxxxxx with the table name you copied in the previous step).

SELECT

logName, resource.type, resource.labels.zone, resource.labels.project\_id,

FROM

`project\_logs.syslog\_xxxxxxxx`

Copied!

content\_copy

1. Then click **Run**.

Feel free to experiment with some other queries that might provide interesting insights.

**Note:**Cloud Logging exports incoming log entries before any decision is made about ingesting the entry into logging storage. As a result, only new log entries will be exported to the sink. As a result, you may not see a syslog\_(1) table as all the syslog entries were generated prior to the export.

Existing log entries already ingested into Cloud Logging can be extracted using commands like:

gcloud logging read "resource.type=gce\_instance AND logName=projects/[PROJECT\_ID]/logs/syslog AND textPayload:SyncAddress" --limit 10 --format json.**Note:**You have set up an export for all the log entries generated by all services in the project. You can also create aggregate exports, which export log entries generated across projects, grouped by billing account, folder, or organization.

Click Check my progress to verify the objective.

Configure the export to BigQuery

Check my progress

## Task 4. Create a logging metric

In this task, you create a metric that you can use to generate alerts if too many web requests generate access denied log entries.

Cloud Monitoring allows you to create custom metrics based on the arrival of specific log entries.

1. Go back to the Logs Explorer page (**Navigation menu > Logging > Logs Explorer**).

**Note:**If prompted, click **Leave** for unsaved work.

1. Select **Create Metric** (right-hand side of the Console) to create a logging metric based on this filter.
2. In the Metric Editor, set **Metric Type** as **Counter**.
3. Under the **Details** section, set the **Log metric name** to **403s**.
4. Under the **Filter selection** for **Build filter**, enter the following and replace PROJECT\_ID with **GCP Project ID 1**:

resource.type="gce\_instance"

log\_name="projects/PROJECT\_ID/logs/syslog"

Copied!

content\_copy

1. Leave all the other fields at their default.
2. Click **Create Metric**.
3. You will make use of this metric in the dashboarding portion of the lab.

Click Check my progress to verify the objective.

Create a logging metric

Check my progress

## Task 5. Create a monitoring dashboard

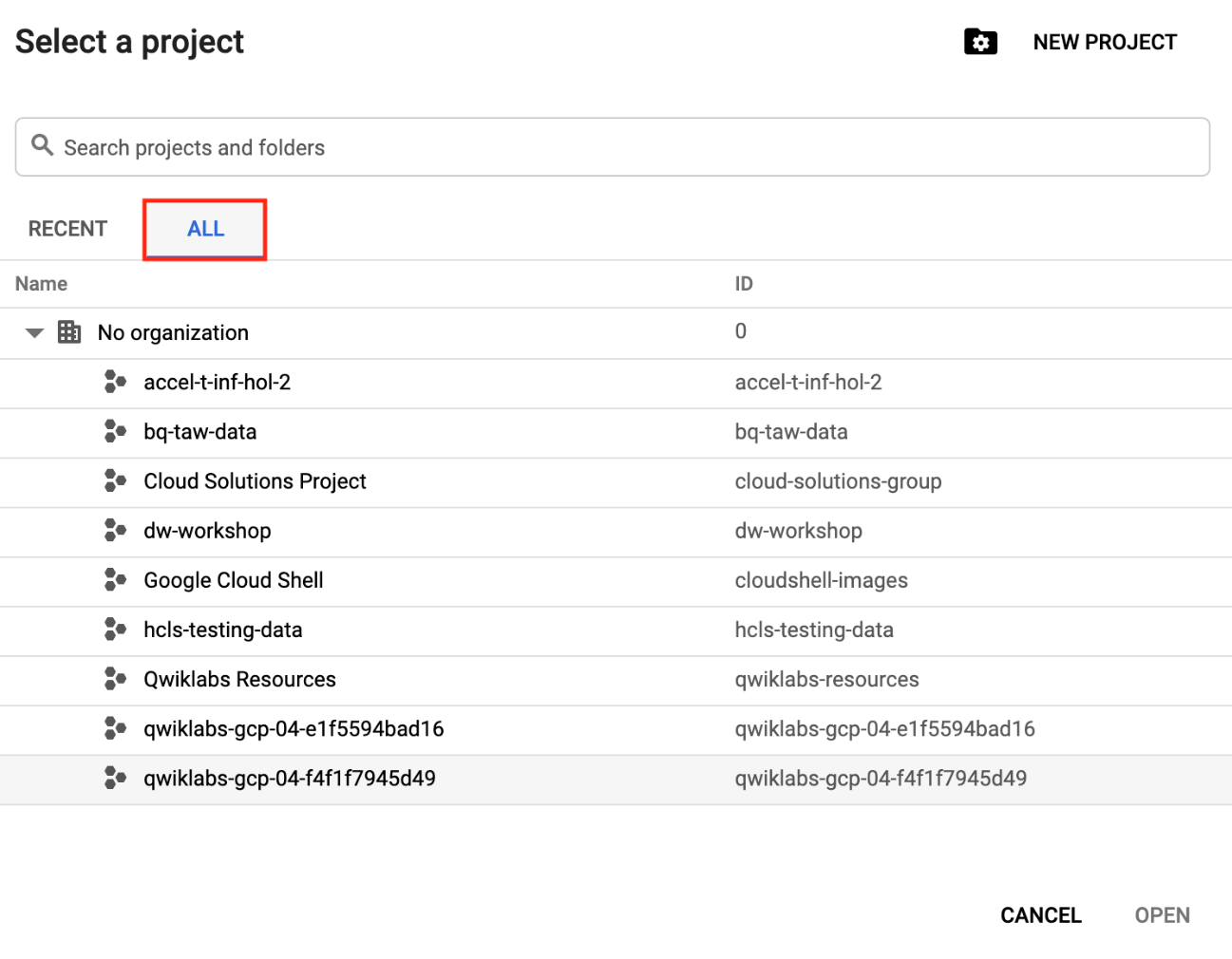
In this task, you switch to the second project created by Qwiklabs and setup a Monitoring workspace.

### **Switch projects**

1. Switch to the second project created by Qwiklabs (use the **GCP Project ID 2** from the Qwiklabs Connection Details). The current project ID is displayed at the top of the console.



1. Click the project name at the top of the Cloud Console and click the **All** tab.



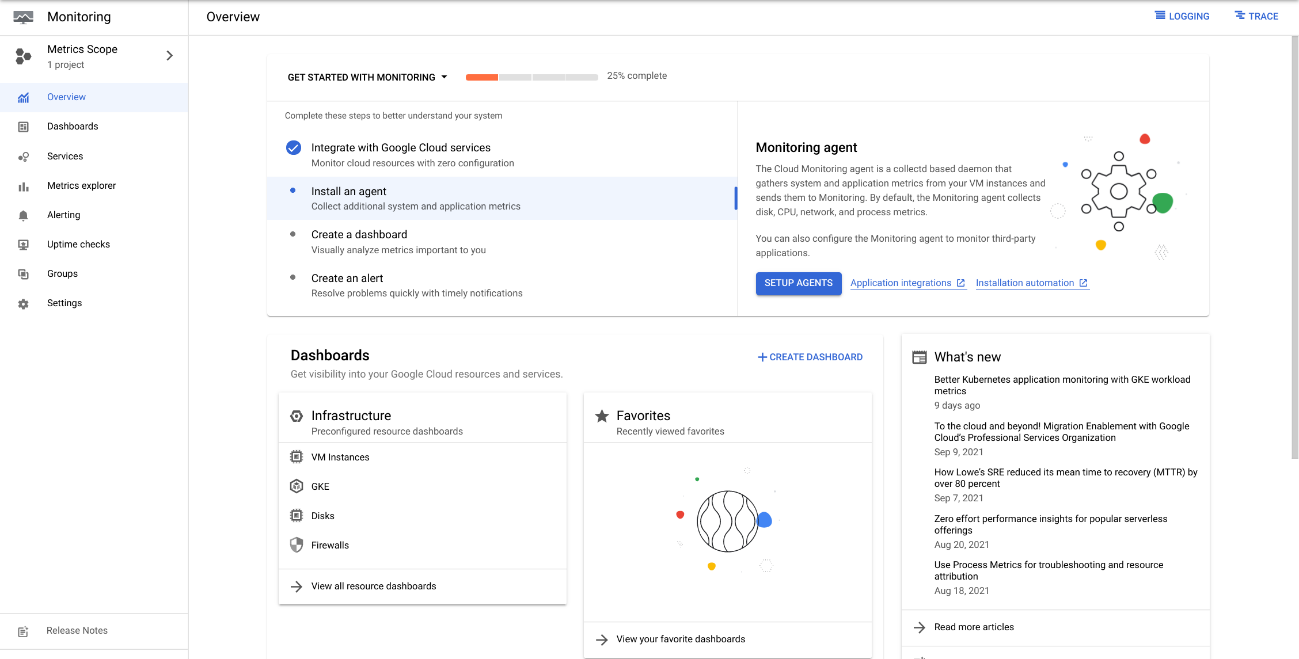
1. Click the second project you want to switch to. Verify it is the **GCP Project ID 2** from the Qwiklabs Connection Details.

### **Create a Monitoring workspace**

You will now setup a Monitoring workspace that's tied to your Google Cloud Project. The following steps create a new account that has a free trial of Monitoring.

1. In the Cloud Console, click on **Navigation menu** > **Monitoring**.
2. Wait for your workspace to be provisioned.

When the Monitoring dashboard opens, your workspace is ready.



Now add the first project to your Cloud Monitoring workspace.

1. In the left menu, click **Settings** and then click **Add GCP Projects** in the GCP Projects section.
2. You'll see one of the projects for this lab as a monitored account. Check the box next to the other project you have, then for the scoping project select **Use this project as the scoping project** and click **Add Projects**.
3. Click **Confirm**.

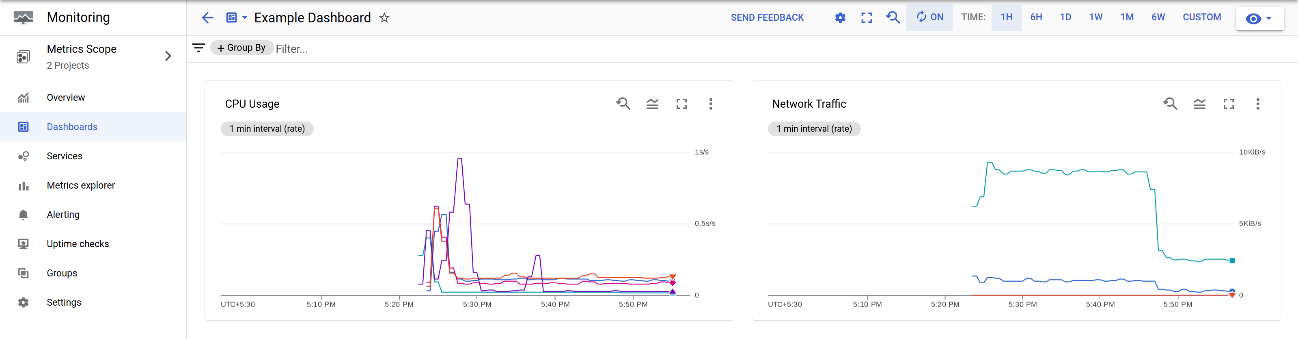
### **Create a monitoring dashboard**

1. In the left pane, click **Dashboards**.
2. Click **+ Create Dashboard**.
3. Replace the generic dashboard name at the top with Example Dashboard.
4. Click **Line**.
5. For **Chart Title**, give your chart a name of **CPU Usage**.
6. For **Resource & Metric**, select **VM Instance > Instance > CPU usage**. Make sure it's the one that follows the format: compute.googleapis.com/instance/cpu/usage\_time. Click **Apply**.

**Note:**If the metric is not visible, make sure to **turn off** the toggle for Show only active resources and metrics.

1. Click **Add Chart**.
2. Click **Line**.
3. Name the chart Network Traffic and set the **Resource & Metric** to **VM Instance > Interface > Network traffic**. Make sure it's the one that follows the format: agent.googleapis.com/interface/traffic. Click **Apply**.
4. Click **Toggle auto-refresh** to ON for getting the graph results (located in the top-right hand of the Monitoring toolbar).

When it's done loading, you should see your two graphs — one for CPU usage and the other for network traffic — populated.



You can now explore some other options by editing the charts such as Filter, Group By, and Aggregation.

## Congratulations!

In this lab, you learned how to do the following:

1. View logs using a variety of filtering mechanisms.
2. Exclude log entries and disable log ingestion.
3. Export logs and run reports against exported logs.
4. Create and report on logging metrics.
5. Use Cloud Monitoring to monitor different Google Cloud projects.
6. Create a metrics dashboard.