**Configuring and Using Cloud Logging and Cloud Monitoring**

experimentLabschedule1 hour 15 minutesuniversal\_currency\_alt1 Creditshow\_chartIntroductory

infoThis lab may incorporate AI tools to support your learning.

**Overview**

In this lab, you will learn common configurations and uses of both [Cloud Logging](https://cloud.google.com/logging) and [Cloud Monitoring](https://cloud.google.com/monitoring).

You will learn how to view logs with filtering mechanisms, export logs to BigQuery syncs, and create logging metrics. You will also learn how to use Cloud Monitoring to view consumption metrics and create dashboards..

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.

**Setup and requirements**

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.

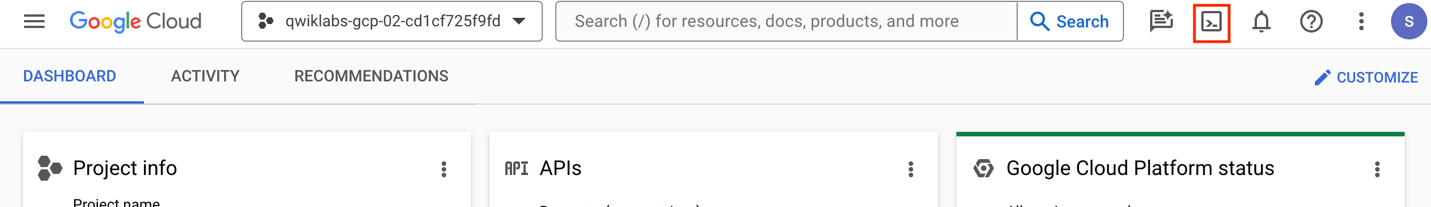
**Note:** Do not click **End Lab** unless you have finished the lab or want to restart it. This clears your work and removes the project.

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.



1. 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

Copied!

content\_copy

**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

Copied!

content\_copy

**Output:**

[core]

project =

**Example output:**

[core]

project = qwiklabs-gcp-44776a13dea667a6

**Note:**Full documentation of **gcloud** is available in the [gcloud CLI overview guide](https://cloud.google.com/sdk/gcloud).

**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.

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

1. If you have not activated cloud shell yet then, activate the Cloud Shell by clicking on **Activate cloud shell**. If prompted, click **Continue**.
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 Ops Agent section with the following:

# install Ops Agent

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

sudo bash add-google-cloud-ops-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-v20240214

Copied!

content\_copy

1. Add the following flag at the end of line 16 to set the machine type for the Linux VM:

--machine-type=e2-micro

Copied!

content\_copy

1. Add the following flag at the end of line 17 to set the machine type for the Windows VM:

--machine-type=e2-standard-2

Copied!

content\_copy

1. After pasting, make sure that your lines of code are properly indented.
2. Save your file.
3. In the Cloud Console, click **Open Terminal** in the top-right corner.

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 Ops Agent installed.
* A Windows VM with Ops Agent installed.
* A Google Kubernetes Engine cluster with an Nginx deployment.
* A Pub/Sub Topic and Subscription.

1. Run the following command to replace the zones in the setup script with a new one:

sed -i 's/us-west1-b/us-east1-c/g' setup.sh

Copied!

content\_copy

1. Now run the following command. If prompted, click **Authorize**.

./setup.sh

Copied!

content\_copy

You can safely ignore errors about service accounts and firewalls already existing.

Ensure you receive a similar output that states that both the Linux and Windows VMs are created:

Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-00-029875ba6255/zones/us-east1-c/instances/linux-server-qwiklabs-gcp-00-029875ba6255].

NAME: linux-server-qwiklabs-gcp-00-029875ba6255

ZONE: us-east1-c

MACHINE\_TYPE: e2-micro

PREEMPTIBLE:

INTERNAL\_IP: 10.138.0.2

EXTERNAL\_IP: 34.83.92.58

STATUS: RUNNING

Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-00-029875ba6255/zones/us-east1-c/instances/windows-server-qwiklabs-gcp-00-029875ba6255].

......

......

NAME: windows-server-qwiklabs-gcp-00-029875ba6255

ZONE: us-east1-c

MACHINE\_TYPE: e2-standard-2

PREEMPTIBLE:

INTERNAL\_IP: 10.138.0.3

EXTERNAL\_IP: 35.247.97.91

STATUS: RUNNING

Updated [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-00-029875ba6255/zones/us-east1-c/instances/linux-server-qwiklabs-gcp-00-029875ba6255].

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**. If prompted, close the notification.
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 a 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. Now click on the **All Log Names** dropdown, and select **syslog**, and then click **Apply**.

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**, type 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**, type project\_logs, and click **Create Dataset**.
7. Click **Next**.
8. In the **Build inclusion filter** list box, copy and paste 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.

Configure HTTP load balancing exports to BigQuery

You will now create an export for the **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**, type 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, copy and paste 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 the arrow next to your project (this starts with qwiklabs-gcp-xxx) and you should see a project\_logs dataset revealed 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, under the table info you will see the full table name in the **Table ID**, 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. To see a subset of your tables fields, paste the below query in the query editor tab (replacing qwiklabs-gcp-xx.project\_logs.syslog\_xxxxx with the table name you copied in the previous step).

SELECT

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

FROM

`qwiklabs-gcp-xx.project\_logs.syslog\_xxxxx`

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** from **Actions** dropdown (right-hand side of the Console) to create a logging metric based on this filter.
2. In the Log-based metric Editor, set **Metric Type** as **Counter**.
3. Under the **Details** section, set the **Log-based 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**

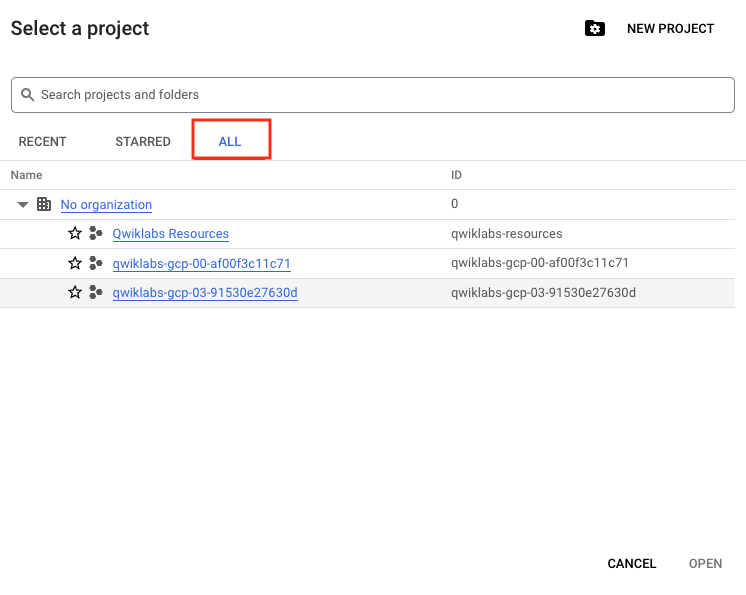
In this task, you switch to the second project created by Qwiklabs and setup a Monitoring workspace.

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.

GCP project identifier

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



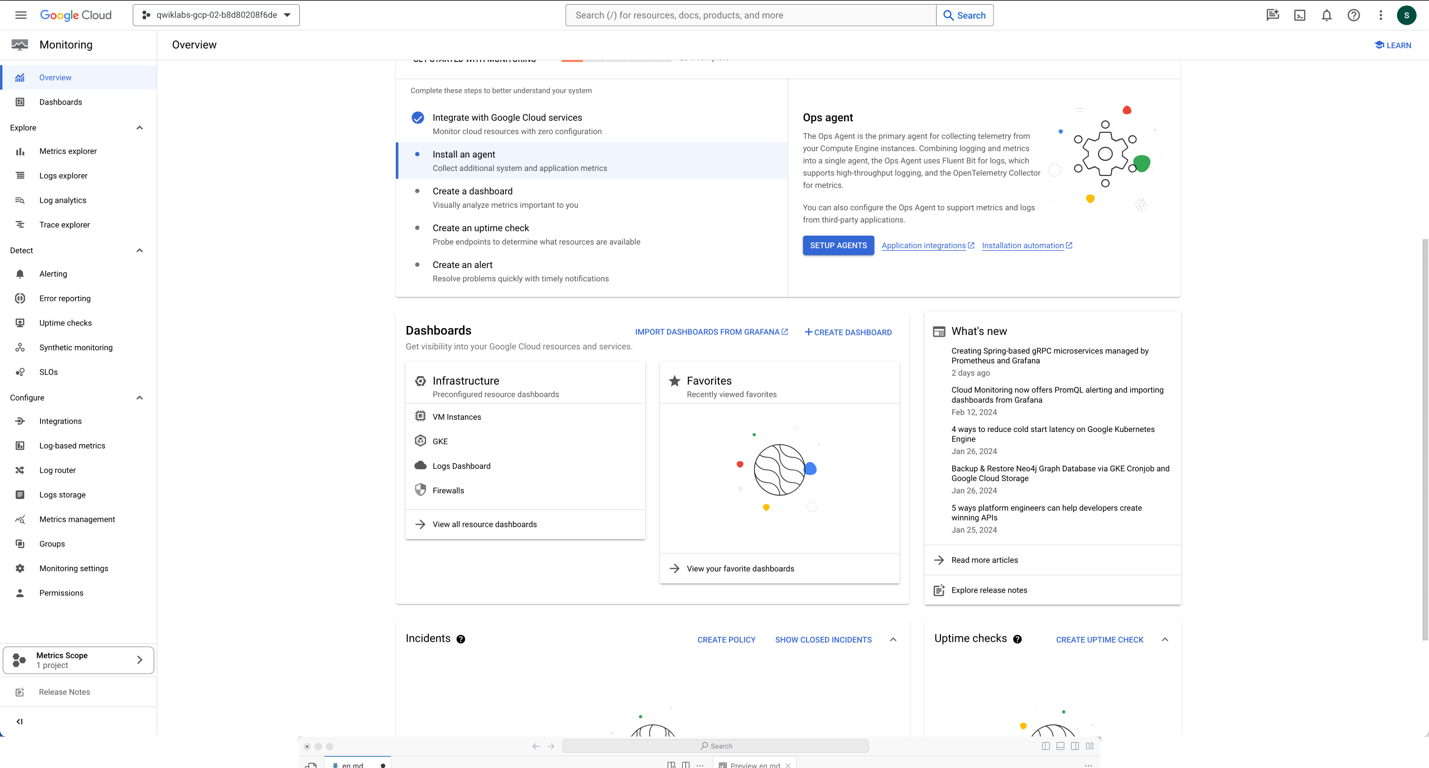
1. Click the second project you want to switch to. Verify it is the **GCP Project ID 2** from the Qwiklabs Connection Details.
2. Click **Open**.

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.



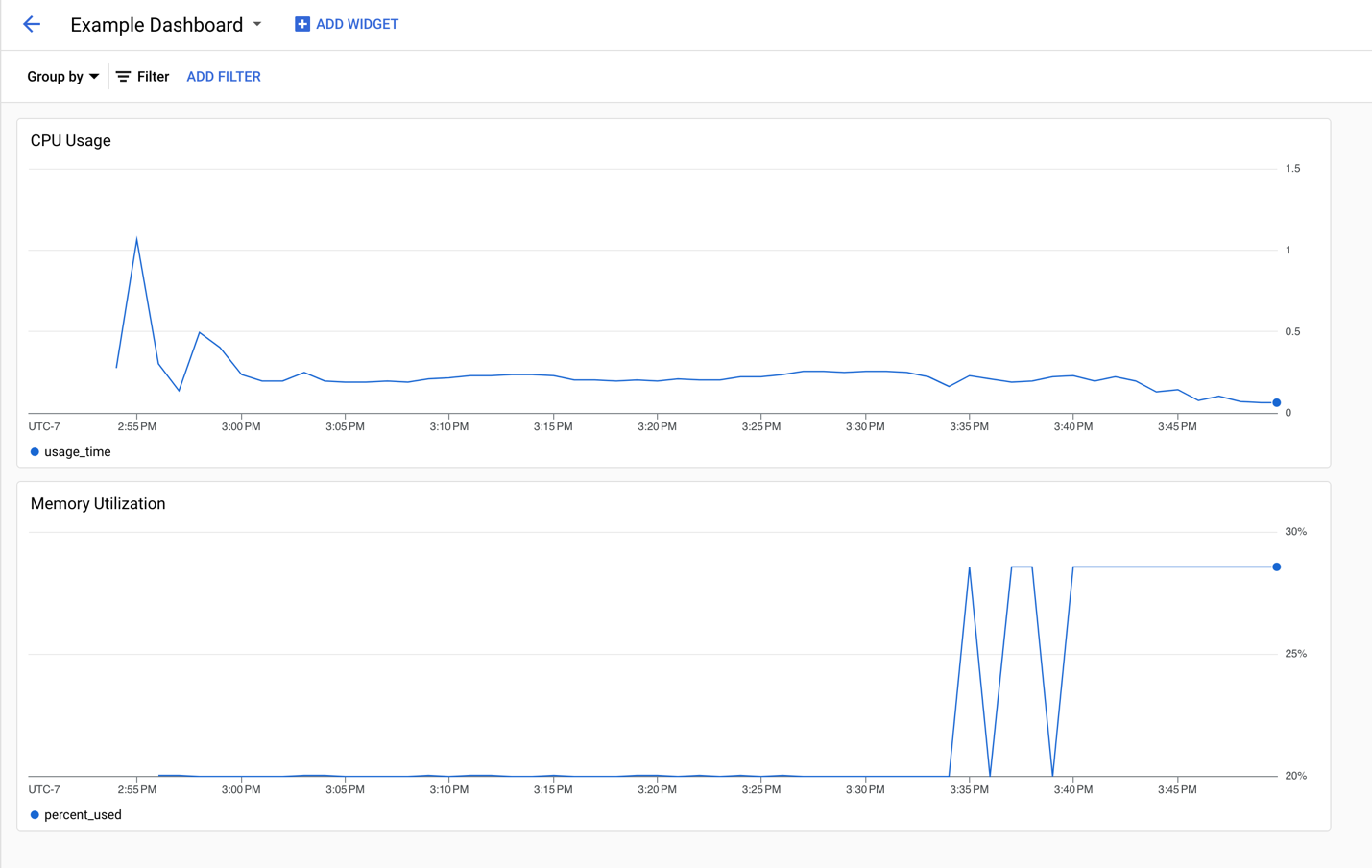
Now add the first project to your Cloud Monitoring workspace.

1. In the left menu, click **Monitoring Settings**, and then click **+ Add GCP Projects**.
2. Click **Select Projects**
3. Select the checkmark next to your first project ID and click **Select**.
4. Click **Add Projects**.

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 **Add Widget** > **Line**.
5. For **Widget Title**, enter in **CPU Usage**.
6. Click the **Metric** dropdown.
7. Click **Active** to deselect it. The tick should disappear.
8. For **Metric**, select **VM Instance > Instance > CPU usage**. Make sure it's the one that follows the format: compute.googleapis.com/instance/cpu/usage\_time.
9. Click **Apply**.
10. Now click **Apply** in the top-right corner.
11. Click **Add Widget** > **Line**.
12. For **Widget Title**, enter in **Memory Utilization**.
13. Click the **Metric** dropdown.
14. Click **Active** to deselect it. The tick should disappear.
15. For **Metric**, select **VM Instance > Memory > Memory Utilization**. Make sure it's the one that follows the format: agent.googleapis.com/memory/percent\_used.
16. Click **Apply**.
17. Now click **Apply** in the top-right corner.

You should now see your two graphs—one for CPU usage and the other for memory utilization—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.

**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.