

Creating and Alerting on Logs-based Metrics

GSP091



Logs-based metrics are [Cloud Monitoring](#) metrics that are based on the content of log entries. It can help you identify trends, extract numeric values out of the logs, and set up an alert when a certain log entry occurs by creating a metric for that event. You can use both system and user-defined logs-based metrics in Cloud Monitoring to create charts and alerting policies. Logs-based metrics are time series that are generated from data in logs. In this lab you'll do all of the above!

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:

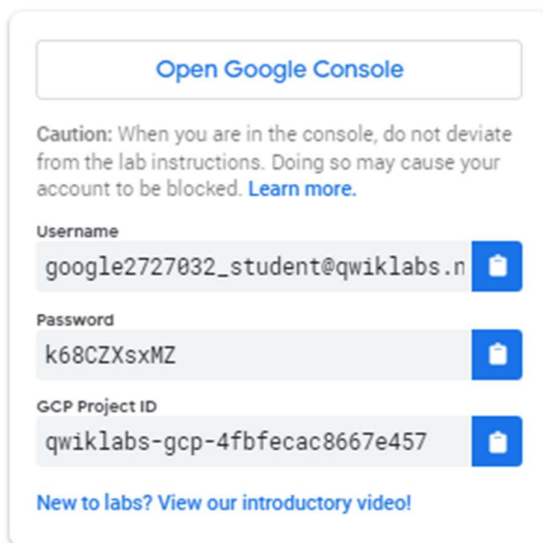
- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.

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.

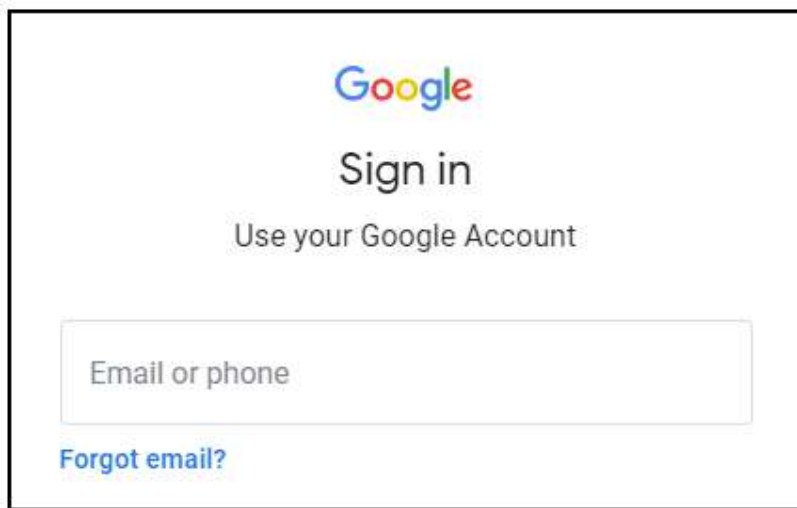


The screenshot shows a sign-in panel for the Google Cloud Console. At the top is a button labeled "Open Google Console". Below it is a caution message: "Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. [Learn more.](#)". The panel contains three input fields, each with a copy icon to its right:

- Username:** google2727032_student@qwiklabs.n
- Password:** k68CZXsxMZ
- GCP Project ID:** qwiklabs-gcp-4fbfecac8667e457

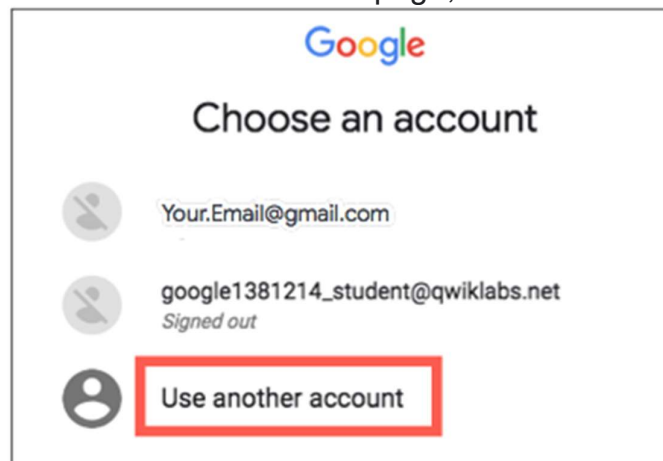
At the bottom, there is a link: "New to labs? [View our introductory video!](#)".

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

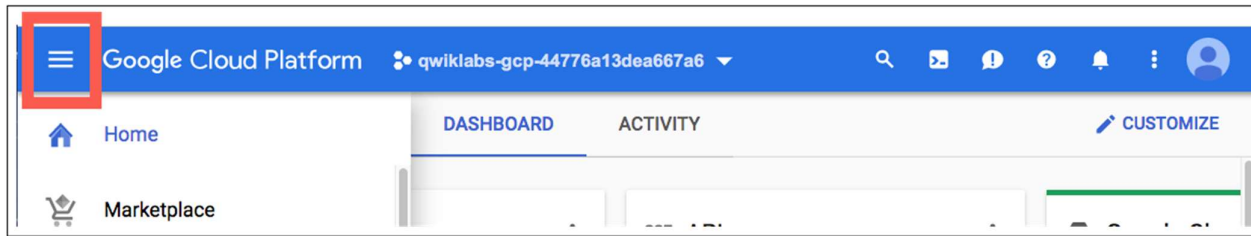
4. Click through the subsequent pages:

- 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-

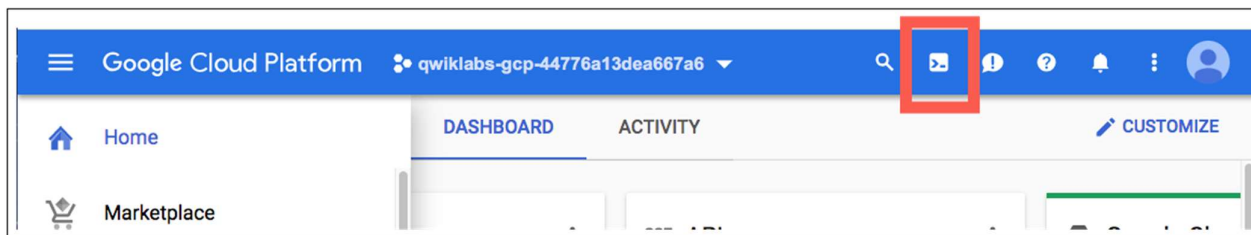
left.



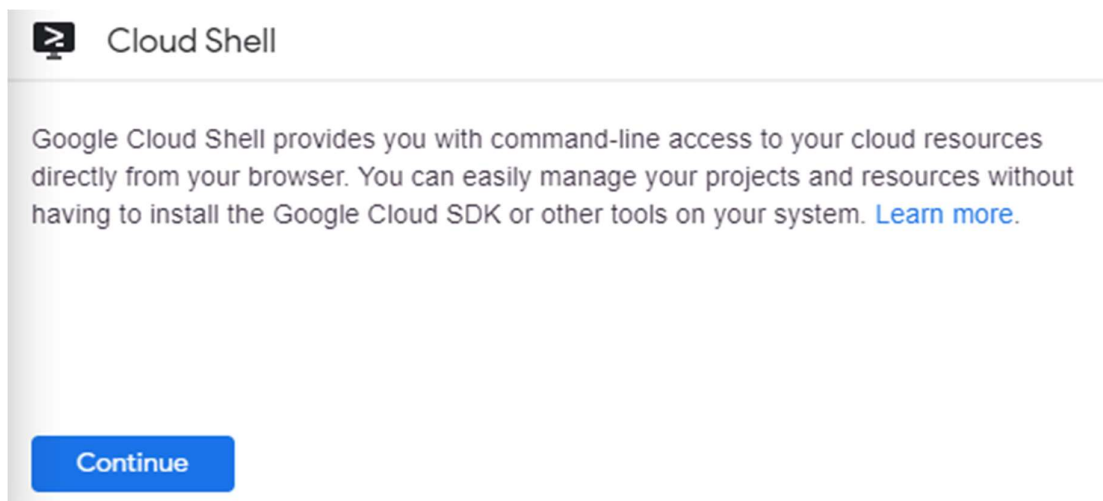
Activate Cloud Shell

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.

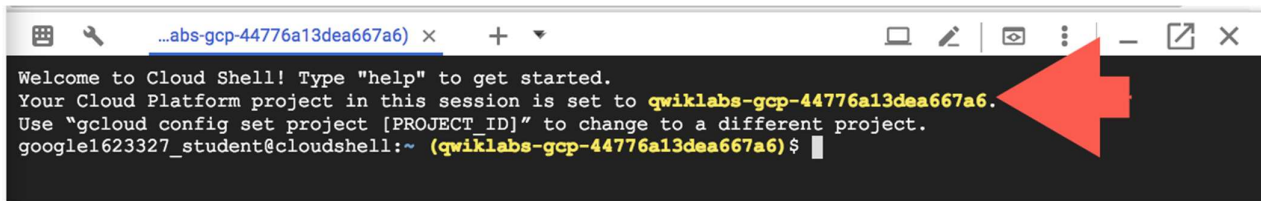
In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.



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:



```
...abs-gcp-44776a13dea667a6) x + ▾
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwiklabs-gcp-44776a13dea667a6.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
google1623327_student@cloudshell:~ (qwiklabs-gcp-44776a13dea667a6) $
```

`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](#).

Resources are being set up for this lab, including a virtual machine. Make sure you see the green Lab Running light on the page where you started the lab before continuing.

Create resources for the lab

The first resource you need for this lab is an app that generates logs. In this section, you deploy a Guestbook app to App Engine, then create some uptime checks for Cloud Monitoring to log.

Use the following commands to clone the app example to your Google Cloud project:

```
git clone https://github.com/GoogleCloudPlatform/appengine-guestbook-python
cd appengine-guestbook-python/
gcloud app create
```

Type in a number for the region where you want the App Engine app created.

Next, run:

```
gcloud app deploy --version 1
```

Enter **Y** to continue.

```
gcloud datastore indexes create index.yaml
```

Enter **Y** to continue.

Monitor indexes

Monitor the indexes by going to **Navigation menu > Datastore > Indexes** in the Console. Give it a couple of minutes to set up. Use the **Refresh** button at the top of the screen. When you see a green check, the index is set up and the status should be "Serving".

Click **Check my progress** to verify the objective.

Deploy a Guestbook app to App Engine and Indexes for datastore
Check my progress

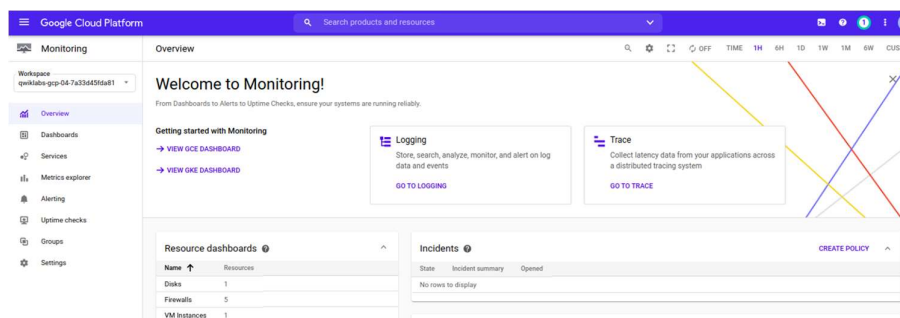
Prepare to install Cloud Monitoring

Create a Monitoring workspace

Now set up 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 **Navigation menu > Monitoring**.
2. Wait for your workspace to be provisioned.

When the Monitoring dashboard opens, your workspace is ready.



Click **Check my progress** to verify the objective.

Create a Monitoring workspace
Check my progress

Install the Monitoring and Logging agents

Agents collect data and then send or stream info to Cloud Monitoring in the Cloud Console.

The *Cloud Monitoring agent* is a collectd-based daemon that gathers system and application metrics from virtual machine instances and sends them to Monitoring. By default, the Monitoring agent collects disk, CPU, network, and process metrics. Configuring the Monitoring agent allows third-party applications to get the full list of agent metrics. [Learn more](#).

The *Cloud Logging agent* streams logs from your VM instances and from selected third-party software packages to Cloud Logging. It is a best practice to run the Cloud Logging agent on all your VM instances. [Learn more](#).

To install the agents on the VM:

1. In the Monitoring Overview window, click **VIEW GCE DASHBOARD**.

Getting started with Monitoring

→ [VIEW GCE DASHBOARD](#)

→ [VIEW GKE DASHBOARD](#)

2. Click on the **VMs Dashboard** link under **Compute Engine**.

Compute Engine

[VMs Dashboard](#) | [Installing the Monitoring Agent](#)

You will see your VM, **instance1**, listed.

3. Under the **Monitoring Agent Status** for **instance1**, click **Install Agent**.

INVENTORY OVERVIEW CPU MEMORY DISK NETWORK EXPLORE							
No agents detected. Monitoring agents collect memory metrics, disk metrics, and more. Learn more about agents and how to manage them across multiple VMs . DISMISS							
Name ^	Zone	Monitoring agent status	Logging agent status	Public IP	Private IP	Size	Connect
instance1	us-central1-a	Install Agent	Install Agent	35.223.197.223	10.128.0.2	n1-standard-2	SSH ▾
Items per page: 25 1 - 1 of 1 < >							

This will bring up an **Agent Details** side window which can be used to help install the agent.

4. In the **Agent Details** window, ensure **Debian** is selected as the operating system, and click **Install Agent**.

Install the Monitoring Agent

Use the Cloud Monitoring agent to gather system and application metrics (disk, CPU, network and process) from VM instances and send them to Monitoring. [Learn more](#)

Confirm VM operating system

Review VM details and select the relevant OS.

[View additional details.](#)

Name	instance1
Boot disk image	debian-9-stretch
Operating System *	Debian ▼

Select the VM OS to install the appropriate agent

INSTALL AGENT

CANCEL

This will auto generate a command in your Cloud Shell that SSHes into your VM and installs the agent.

Copy the generated command from the Agent details window and run it in your Cloud Shell. Type **y** when asked if you want to continue and then press **enter** twice to not use a passphrase.

While the installation of the agent runs, type and enter **y** whenever prompted to continue.

A few minutes after the installation finishes you should see a green check mark in your **Agent Details** window:

Agent details



Agent installation successful!

Additional resources for the lab

Since this lab uses logs, it helps if you had some interesting logs. In this lab, you generate these logs with uptime checks and a VM creation alert.

Create an uptime check

Create a few uptime checks for the app that's running on App Engine. The uptime checks simulate load against the app. App Engine automatically captures logs from all requests and produces one log per minute, which provides data to analyze later.

1. Go back to the Cloud Console and click **Navigation menu > App Engine**. Click on the link in the upper right corner, which ends in ".appspot.com".



Your App Engine Guestbook app opens in a new browser tab. Record the URL to use when you configure the uptime check.

2. Go back to the Cloud Monitoring window (**Navigation menu > Monitoring**).
3. In the left menu, click **Uptime checks**, and then **Create Uptime Check**.
4. In the **Create uptime check** dialog, set the following fields:

Field	Value
Title	pizza check

Click **Next**.

In the **Target** section, set the following fields:

Field	Value
Protocol	HTTP
Resource Type	URL
Hostname	Paste in the URL for your app (that you previously recorded). Remove the <code>https://</code> and the last <code>/</code> from the URL.
Path:	<code>/?food=pizza</code>
Check Frequency	1 min

Click **Next**.

You can use the default settings for **Response Validation** and **Alert & Notification**, so click **Next** one more time.

5. Now, click **Test**.

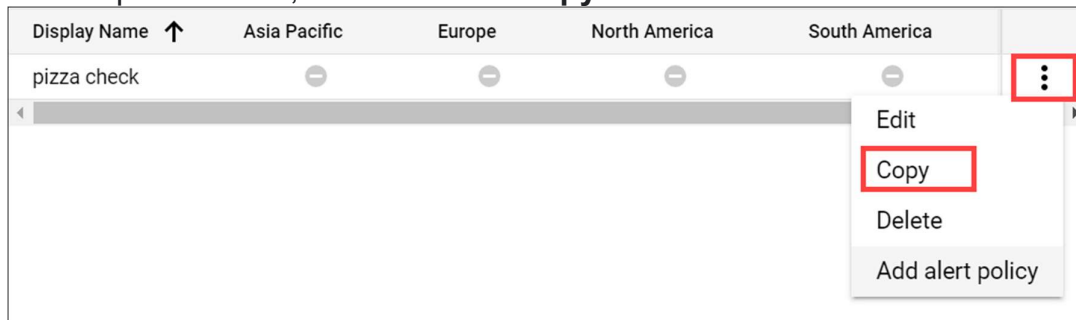
You should see a successful response.

6. Click **Create**.

The Uptime checks window opens and lists the pizza check in the Uptime checks table.

Create 2 more uptime checks

1. Still in the Uptime checks window, click the vertical ellipsis in line with pizza check uptime check, and then click **Copy**.



- Change the Title to "burger check".
- Click **Next** and change the Path to `/?food=burger`.
- Click **Next** twice and then click **Create**.

2. Copy an uptime check one more time.

- Change the Title to "cake check".
- Click **Next** and change the Path to `/?food=cake`
- Click **Next** twice and then click **Create**.

All three uptime checks are listed in the Uptime checks table.

Click **Check my progress** to verify the objective.

System defined and user defined logs-based metrics

Logs-based metrics are divided into System Defined and User Defined.

System defined logs-based metrics

System defined logs-based metrics are ready to use right out of the box. These [system logs-based metrics](#) include:

Metrics around logs ingested

- **Byte_count**: Number of bytes in all log entries ingested. This is broken down by monitored resource type, log stream name, and severity level.

Metrics around logs excluded

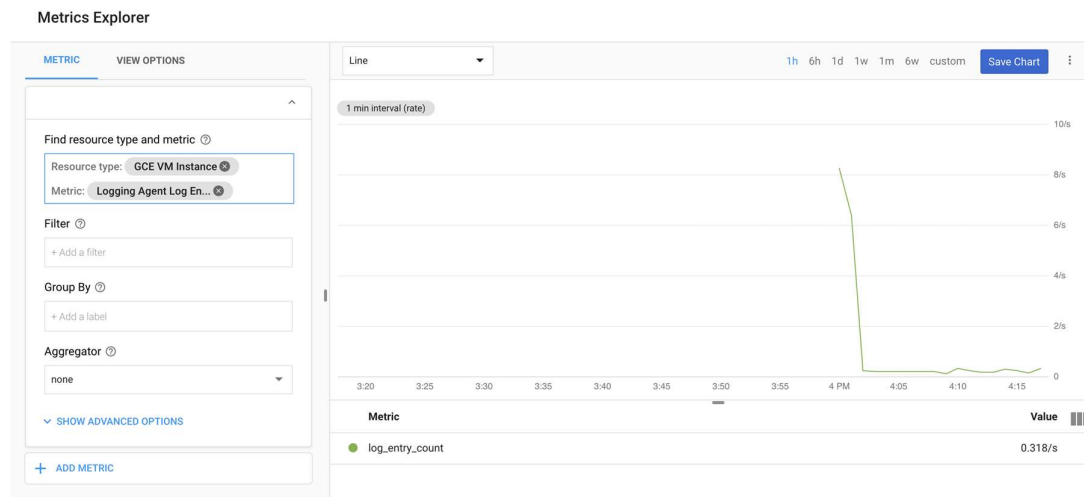
- **Excluded_byte_count**: Number of bytes in log entries that were excluded. This is broken down by the monitored resource type.
- **Excluded_log_entry_count**: Number of log entries that were excluded. This is broken down by the monitored resource type.

Metrics around logs based metrics

- **Dropped_log_entry_count**: Despite the name, this does not show log entries dropped by Cloud logging but rather the number of log entries that did not contribute to logs based metrics because they arrived too late.
- **Log_entry_count**: Number of log entries that contributed to logs based metrics so that $\text{dropped_log_entry_count} + \text{log_entry_count}$ is the total number of log entries ingested by Cloud Logging.
- **Metric_throttled**: Indicates if points are being dropped for logs-based metrics due to exceeding time series limits.
- **Time_series_count**: Estimate of the active time series count for logs-based metrics. Most system logs-based metrics are counter metrics. **Counter metrics** count the number of log entries that match an advanced logs filter.

Now you'll look more closely at a system generated logs-based metric: `Log_entry_count`.

1. In the left menu, click **Metrics Explorer**:
2. To find a metric, start typing "log entries" and choose **Logging Agent Log Entry Count**.
3. Select **Compute Engine VM Instance** as your Resource.

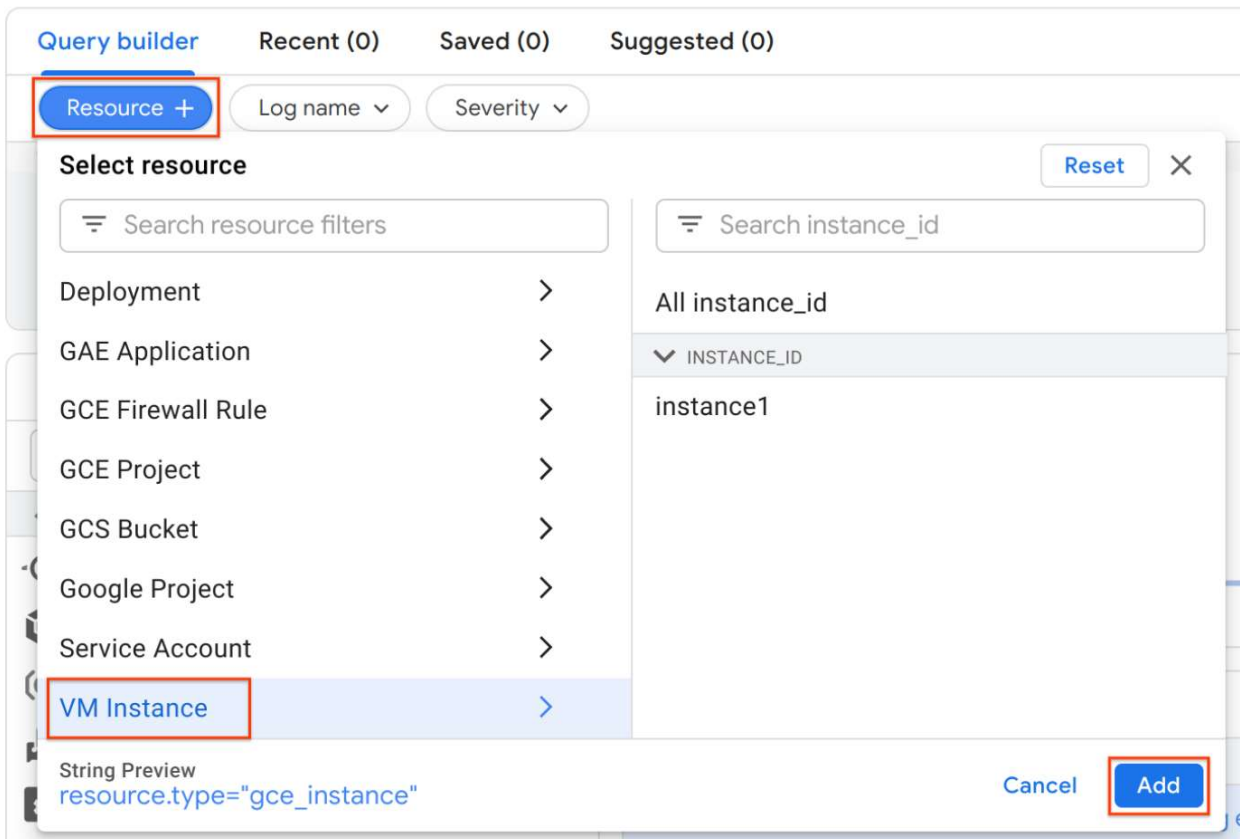


You're looking at the graphical representation of log entries for the machine that was started as one of your resources when you began this lab.

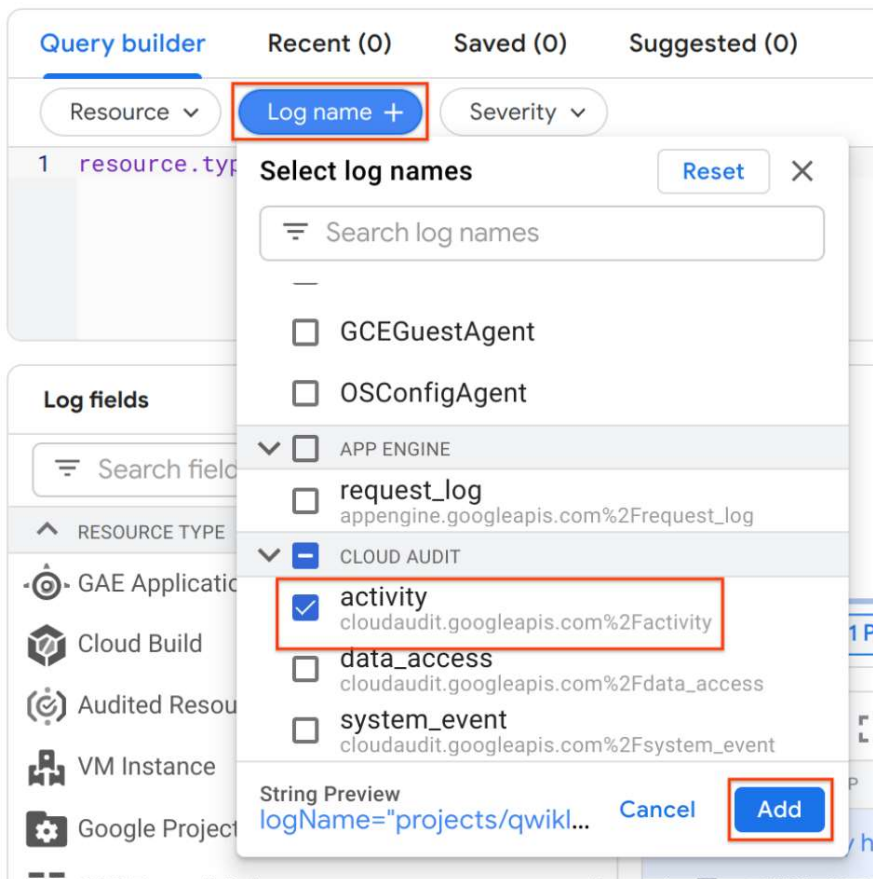
User defined logs-based metrics

Create your own logs-based metrics using data from existing logs. These are called **user defined logs-based metrics**. In this section, you create a metric using a log entry.

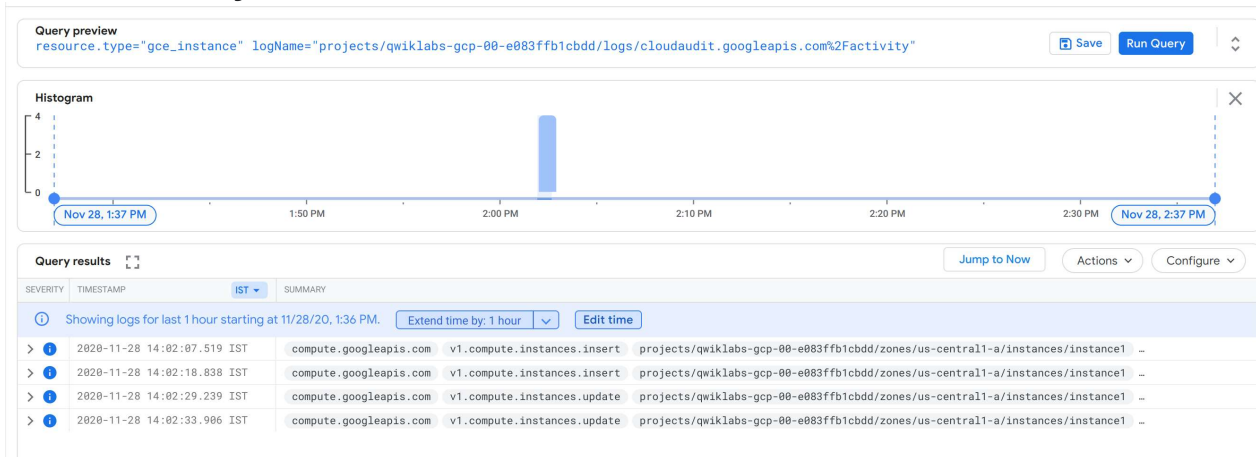
1. Select **Navigation menu > Logging > Logs Explorer**.
The Cloud Logging window opens.
 2. Filter for activity logs from VM instance you previously created.
- In the **Resource** dropdown, select **VM Instance** and click **Add**.



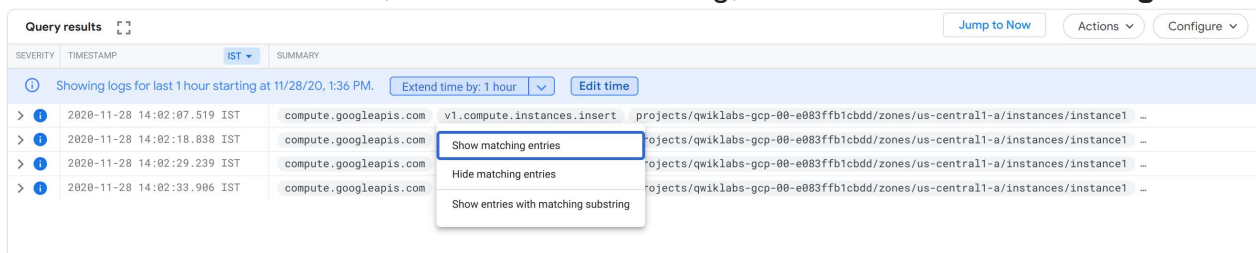
- In the **Log name** dropdown, select **activity**, then click **Add**.



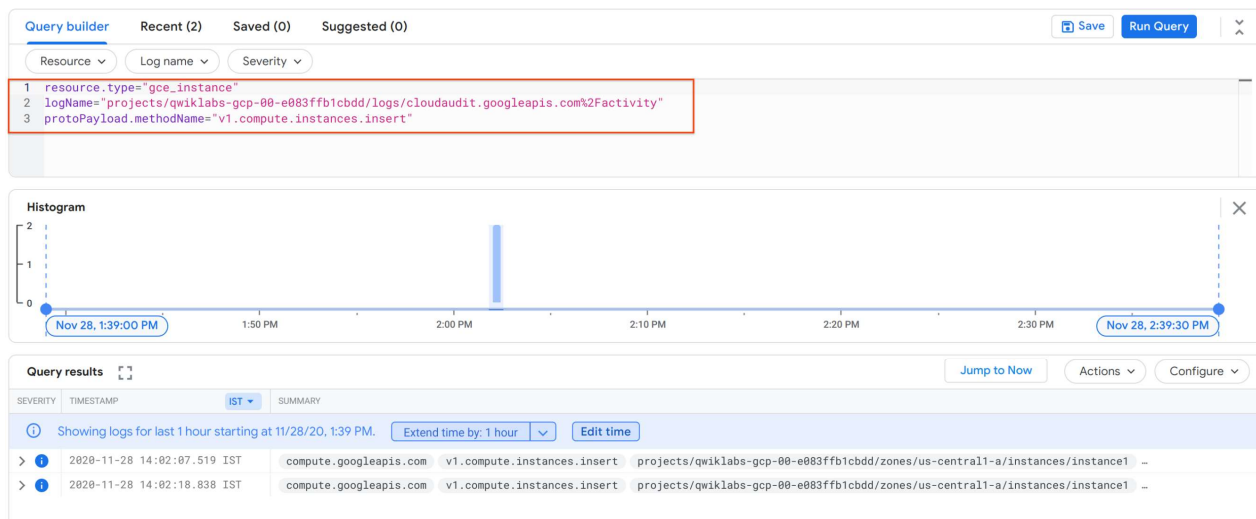
- Click **Run Query**.



- In one of the entries, click on the "insert" tag, then select **Show matching entries**.



The **Query builder** is now displays the criteria for this metric.



The `protoPayload.methodName` in the 3rd row lets you filter on many methods. You can read more about that [here](#).

- On top right of the Query results section, click **Actions** drop down and select **Create Metric**.

- In the Metric Editor name your metric "newVM" then click **Create Metric**.

Your user defined logs-based metric is now listed in the Logs-based metrics page.

User-defined metrics CREATE METRIC DELETE									
Filter Filter user-defined metrics									
<input type="checkbox"/>	Enabled	Name ↑	Type	Description	Previous month usage	Month-to-date usage (MTD)	Created	Last Updated	Filter
<input type="checkbox"/>	✓	newVM	Counter		0 B	0 B	4/20/21, 5:13 PM	4/20/21, 5:13 PM	resource.type="gce_instance" logName="projects/qwikla gcp-04- 9c777304f095/logs/cloudaudit.googleapis.com%2Factiv protoPayload.methodName="v1.compute.instances.inse

Create alerting policy for the VM creation metric

Create an alert to let you know when a new VM gets added to your project.

1. Return to the Cloud Monitoring window (**Navigation menu > Monitoring**).
 2. In the left menu, click **Alerting**, and then click **Create Policy**.
 3. Click **Add Condition** and set the following:
 - For Target, click inside the dropdown menu and choose **VM Instance**.
 - For the metric, start typing "logging/" and select **logging/user/newVM**.
 - For Configuration, set Condition to **is above** the Threshold **0** For **1 minute**.
 - Click **Add**.

If "logging/user/newVM" isn't coming up as a metric when searching, try refreshing your browser. If it still doesn't come up after refreshing, double check that the last step was completed correctly.

Click **Next**.
 4. Click the **Notification Channels** dropdown and click **Manage Notification Channels**.
- In the tab that comes up, find **Email** and click **Add New**. Add your personal email and either your name or an alias for the **Display Name**.

Now, back in the **Create Alerting Policy** tab, select the **Notification Channels** dropdown and click the **refresh** button.

You should see the display name of your email with a check box next to it. Select the checkbox next to your email's display name.

Click **Next**.

5. Enter "New Virtual Machine" as the **Alert Name**.
6. Click **Save**.

Create a new instance

To trigger the alert you just created, go create a new virtual machine instance.

1. In the Cloud Console dashboard, go to **Navigation menu > Compute Engine > VM instances**, then click **Create Instance** at the top of the screen.
2. Name your instance **instance2**, then check the firewall boxes to **allow HTTP and HTTPS traffic**. Leave all the other fields with their default values.
3. Click **Create**.

Wait a couple of minutes for your instance to be launched. You should see an Incident in Cloud Monitoring console in 3-5 minutes.

Continue with the lab, you can check these results at the end.

Click **Check my progress** to verify the objective.

Labels and user defined metrics

User defined labels can be created when you create a metric. An extractor expression is required for each configured label to tell Cloud Logging how to extract values from logs and place them as the labels' value. You cannot add labels to system logs-based metrics.

Create a user-defined metric with a label.

1. Select **Navigation menu > Logging > Logs Explorer**.
2. In **Resource** drop-down, select **GAE Application** and click **Add**.

The screenshot shows the Google Cloud Logs Explorer interface. At the top, there's a header with 'Logs Explorer', an 'OPTIONS' button, a 'REFINE SCOPE' button, and a 'Project' button. Below the header, a light blue banner states 'New features are available in the Logs Explorer.' The main content area has tabs for 'Query builder', 'Recent (2)', 'Saved (0)', and 'Suggested (0)'. Under the 'Query builder' tab, there are filters for 'Resource +', 'Log name', and 'Severity'. The 'Resource +' filter is expanded, showing a 'Select resource' dialog box. This dialog box has two search bars: 'Search resource filters' and 'Search module_id'. The 'Search resource filters' list includes 'Cloud Datastore Index', 'Cloud Pub/Sub Topic', 'Deployment', 'GAE Application' (highlighted), 'GCE Firewall Rule', 'GCE Project', 'GCS Bucket', and 'Google Project'. The 'Search module_id' list shows 'All module_id' and 'MODULE_ID' (expanded) with a 'default' option. At the bottom of the dialog, there's a 'String Preview' section showing 'resource.type="gae_app"' and buttons for 'Cancel' and 'Add'.

3. For **Log name**, select **request_log** and click **Add**.

Logs Explorer **OPTIONS** **REFINE SCOPE** Project

New features are available in the Logs Explorer.

Query builder Recent (2) Saved (0) Suggested (0)

Resource Log name Severity

1 resource.type

Log fields

Search field

RESOURCE TYPE

- ☒ APP ENGINE
 - ☒ request_log
appengine.googleapis.com%2Frequest_log
- ☐ CLOUD AUDIT
 - ☐ activity
cloudaudit.googleapis.com%2Factivity
 - ☐ data_access

String Preview
logName="projects/qwikl..." **Cancel** **Add**

4. Select **Actions > Create Metric**.
5. Name your metric "Foodcount" and add a Description.
6. Click on **Add label** to create a Label.

[←](#) Create logs metric

Configure the settings below to define and create a logs-based metric.

Metric Type

☒ Counter

Counts the number of log entries matching a given filter
[Learn more](#)

☐ Distribution

Collects numeric data from log entries matching a given filter
[Learn more](#)

Details

Log metric name *

Foodcount

Enter a name to describe this metric. 9/100

Description

Enter a description for this metric (optional)

Units

The units of measurement that apply to this metric (for example, bytes or seconds). For counter metrics, leave this blank or insert the digit '1'. For distribution metrics, you can optionally enter units, such as 's', 'ms', etc. [Learn more](#)

Filter selection

[PREVIEW LOGS](#)

Define your logs-based metric

Build filter *

```
1 resource.type="gae_app"
2 logName="projects/qwiklabs-gcp-04-9c777304f095/logs/appengine.googleapis.com%2Frequest_log"
```

Labels

Labels allow logs-based metrics to contain multiple time series [Learn more](#)

[+ ADD LABEL](#)

[CREATE METRIC](#)

[CANCEL](#)

Labels

[Labels](#) allow logs-based metrics to contain multiple time series — one for each label value. All logs-based metrics come with some default labels.

For this lab you'll create a metric for when an uptime check log occurs that has "food" in the log.

1. Go back to what you're doing, which is creating a Metric. In the Metric Editor, you've named your metric Foodcount, and clicked **Add label** to create a label.

2. Set the following:

- Label name: food
- Label type: String
- Field Name: search for "resource" and choose **protoPayload.resource**.
- Regular expressions: `/?food=(.*)`
- Click **Done**

Labels

Labels allow logs-based metrics to contain multiple time series [Learn more](#) 

Label name *

food

Enter a name to describe this metric label

Description

Enter a description for this metric label (optional)

Label type *

STRING

Field Name *

protoPayload.resource

Regular expressions

`/?food=(.*)`

[PREVIEW](#)

Extracts a value from the field that must contain exactly one regex group ()

[DONE](#)

[CANCEL](#)

[CREATE METRIC](#)


[CANCEL](#)


Caution: Be sure to specify the extractor for your label values carefully. A mistake can result in having a large number of active time series. Exceeding time series limits can result in the metric being throttled, a degradation in the performance of charts, or extra time series overage costs.




Click **Create Metric**.

You'll now see your user-defined metric added to the logs-based metrics screen.

User-defined metrics

 CREATE METRIC

 DELETE

Filter user-defined metrics										
<input type="checkbox"/>	Enabled	Name ↑	Type	Description	Previous month usage	Month-to-date usage (MTD)	Created	Last Updated	Filter	
<input type="checkbox"/>		Foodcount	Counter	Foodcount	0 B	0 B	4/20/21, 6:01 PM	4/20/21, 6:01 PM	resource.type="gae_app" logName="projects/qwiklabs-g-9c777304f095/logs/appengine.googleapis.com%2Frequ	⋮
<input type="checkbox"/>		newVM	Counter		0 B	0 B	4/20/21, 5:13 PM	4/20/21, 5:13 PM	resource.type="gce_instance" logName="projects/qwikla-04-9c777304f095/logs/cloudaudit.googleapis.com%2Fa-protoPayload.methodName="v1.compute.instances.inse	⋮

Click **Check my progress** to verify the objective.

Create the Foodcount alerting policy

Now you'll create an alert policy for Foodcount, the metric you just made.

1. Select **Navigation menu > Monitoring > Alerting**, and then click **Create Policy**.
2. Click **Add Condition** and set the following:
 - For Find resource type and metric, start typing "logging/user/" and select **logging/user/Foodcount**. If this metric doesn't appear, close the window and try again.
 - For Configuration, Condition **is above**, with a Threshold of **0**, For **1 minute**.
 - Click **Add**.
 - Click **Next**.
3. To receive an email notification, click on **Notification Channels** dropdown and select the checkbox next to your email's display name you have previously created in the lab.
Click **Next**.
4. Name this policy "food alert".
Click **Save**.

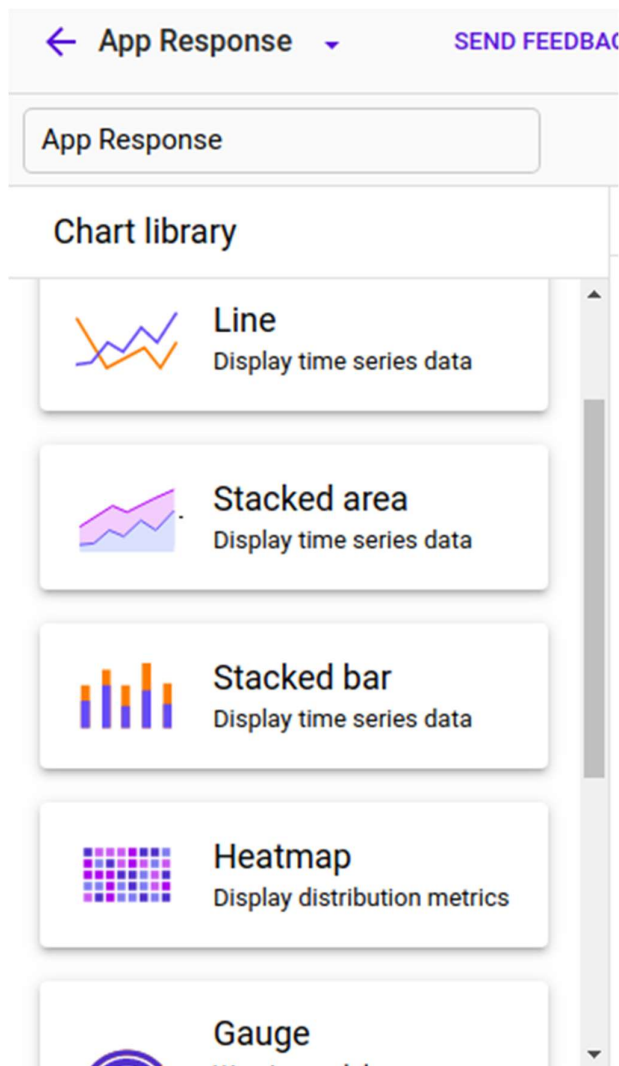
When the next round of uptime checks happen, they should violate the policy since the threshold will be higher than 0, and you'll see an Incident on the Monitoring Overview page.

Click **Check my progress** to verify the objective.

Custom dashboard with heatmap

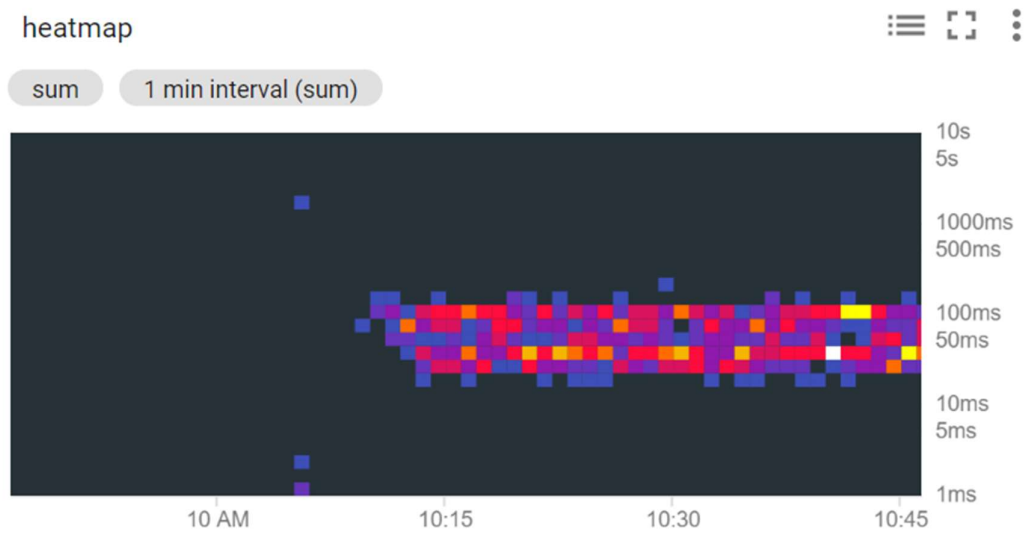
Creating a custom dashboard to show you a specific metric is a great way to see your data. Next you'll create a custom dashboard to show the same latency metric you just used.

1. Still on the Cloud Monitoring window, in the left menu, click **Dashboards > Create Dashboard**.
2. Name the Dashboard "App Response".
3. Select **Heatmap** option in Chart Library, and then set the following:



- **Resource type:** GAE Application
- **Metric:** Response latency (If the option is not visible, deselect the **only show active** checkbox)

- The chart names itself the metric that you're using. You can change the Chart Title to whatever you want - this example uses "Heatmap".



Check for the alert

Still in the **Cloud Monitoring** window, in the left menu, click **Monitoring overview** to check for alerts from the uptime check policies.

To see the new VM alert more clearly, click the policy listed in the **Alerting** section.

To see the status of uptime checks in each region, in the left menu, click **Uptime Checks**.

If you set up any email notifications, check the account you added to your alert to verify that you've been notified. These may take longer to arrive, but you can see the emails even after the lab has ended.

The screenshot shows the Google Cloud Monitoring 'Overview' page. The left sidebar contains a navigation menu with 'Overview' selected. The main content area is titled 'Welcome to Monitoring!' and includes sections for 'Getting started with Monitoring', 'Logging', 'Trace', 'Resource dashboards', 'Uptime checks', 'Groups', 'Incidents', 'Charts', and 'Events'. The 'Uptime checks' section shows a table of checks: 'burger check', 'cake check', and 'pizza check'. The 'Incidents' section shows a table of incidents with columns for 'State', 'Incident summary', and 'Opened'. The 'Events' section shows a table of events with columns for 'Time', 'Event summary', and 'Threshold'. The 'Resource dashboards' section shows a table of resources: 'App Engine', 'Cloud Pub/Sub', 'Cloud Storage', 'Data', 'Firewalls', and 'VM instances'.

Monitoring

Workspace: qwklabs-gcp-01-2285e0bba4

Overview

Welcome to Monitoring!

From Dashboards to Alerts to Uptime Checks, ensure your systems are running reliably.

Getting started with Monitoring

→ VIEW GCE DASHBOARD

→ VIEW GKE DASHBOARD

Logging

Show, search, analyze, monitor, and alert on log data and events

GO TO LOGGING

Trace

Collect latency data from your applications across a distributed tracing system

GO TO TRACE

Resource dashboards

Name	Resources
App Engine	0
Cloud Pub/Sub	4
Cloud Storage	3
Data	2
Firewalls	6
VM instances	2

Uptime checks

Display Name
burger check
cake check
pizza check

Groups

Organize and monitor resources into the groups that make sense for your apps.

Incidents

State	Incident summary	Opened
Open	logging/user/Feedout for qwklabs-gcp-01-2285e0bba4 GAE Application labels (module_id=default, version_id=1, zone-us17) with metric labels (food-burger, log-appeengine.googleapis.co...	Nov 10, 2020, 3:27:45 PM
Open	logging/user/Feedout for qwklabs-gcp-01-2285e0bba4 GAE Application labels (module_id=default, version_id=1, zone-us17) with metric labels (food-pizza, log-appeengine.googleapis.co...	Nov 10, 2020, 3:27:45 PM
Open	logging/user/Feedout for qwklabs-gcp-01-2285e0bba4 GAE Application labels (module_id=default, version_id=1, zone-us17) with metric labels (food-cake, log-appeengine.googleapis.co...	Nov 10, 2020, 3:27:45 PM

Charts

Events

Time	Event summary	Threshold
November 10, 2020 3:27:45 PM	qwklabs-gcp-01-2285e0bba4 GAE Application labels (module_id=default, version_id=1, zone-us17) opened logging/user/Feedout for qwklabs-gcp-01-2285e0bba4 GAE Application labels (module_id=default, version_id=1, zone-us17) with metric labels (food-cake, log-appeengine.googleapis.com/request_log) is above the threshold of 0 with a value of 0.065.	0.065
3:27:45 PM	qwklabs-gcp-01-2285e0bba4 GAE Application labels (module_id=default, version_id=1, zone-us17) opened logging/user/Feedout for qwklabs-gcp-01-2285e0bba4 GAE Application labels (module_id=default, version_id=1, zone-us17) with metric labels (food-burger, log-appeengine.googleapis.com/request_log) is above the threshold of 0 with a value of 0.065.	0.065
3:27:45 PM	qwklabs-gcp-01-2285e0bba4 GAE Application labels (module_id=default, version_id=1, zone-us17) opened logging/user/Feedout for qwklabs-gcp-01-2285e0bba4 GAE Application labels (module_id=default, version_id=1, zone-us17) with metric labels (food-pizza, log-appeengine.googleapis.com/request_log) is above the threshold of 0 with a value of 0.065.	0.065

Rows per page: 10 1 - 10 of many

Congratulations!



Finish Your Quest

This self-paced lab is part of the Qwiklabs [Google Cloud's Operations Suite](#) and [Cloud Logging](#) Quests. A Quest is a series of related labs that form a learning path. Completing a 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 either Quest and get immediate completion credit if you've taken this lab. [See other available Qwiklabs Quests](#).

Take Your Next Lab

Continue your Quest with [Autoscaling an Instance Group with Custom Metrics](#), or check out these suggestions:

- [Monitoring Multiple Projects with Cloud Monitoring](#)

Next Steps / Learn More

- Read about using BigQuery and Cloud Monitoring together in this blog post: <https://cloud.google.com/blog/big-data/2017/10/accelerate-bigquery-solution-development-with-intelligent-log-analysis>

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 April 20, 2021

Lab Last Tested April 20, 2021

Copyright 2021 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.