Creating and Alerting on Logs-based Metrics

qwiklabs.com/focuses/619

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.

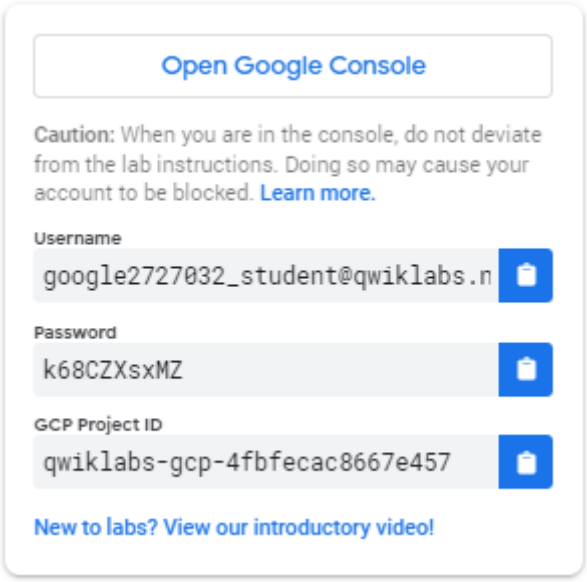
Note: If you already have your own personal Google Cloud account or project, do not use it for this lab.

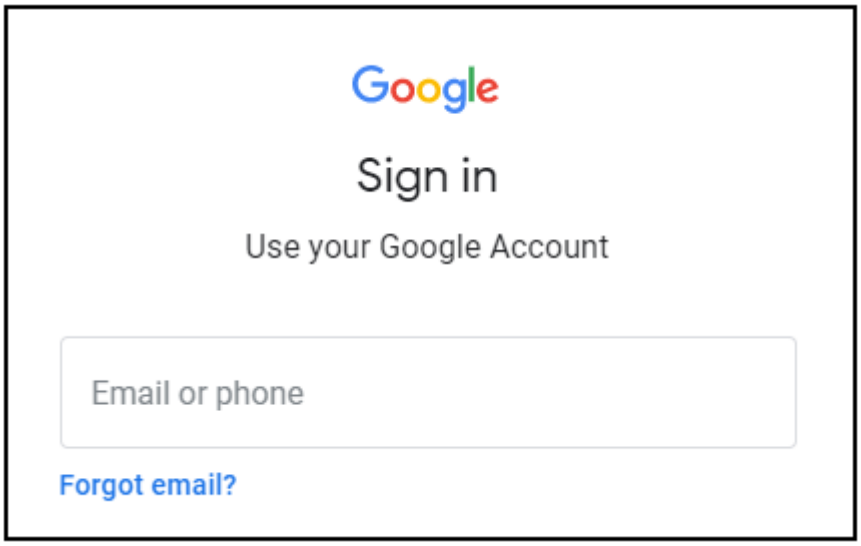
Note: If you are using a Pixelbook, open an Incognito window to run this lab.

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.

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.



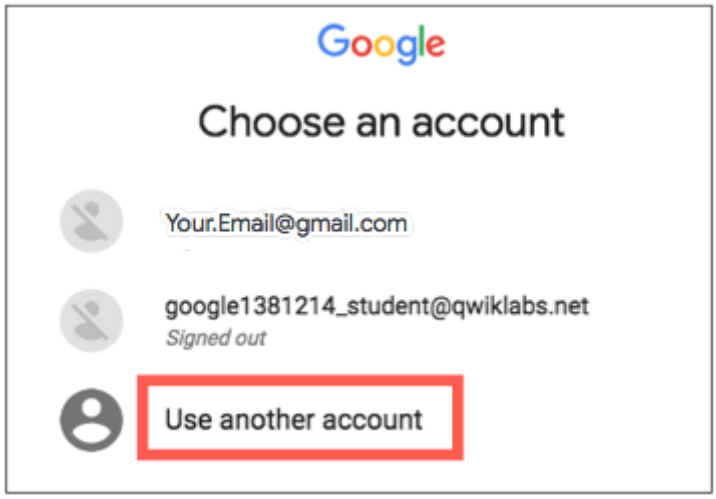


Tip: Open the tabs in separate windows, side-by-side.

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).

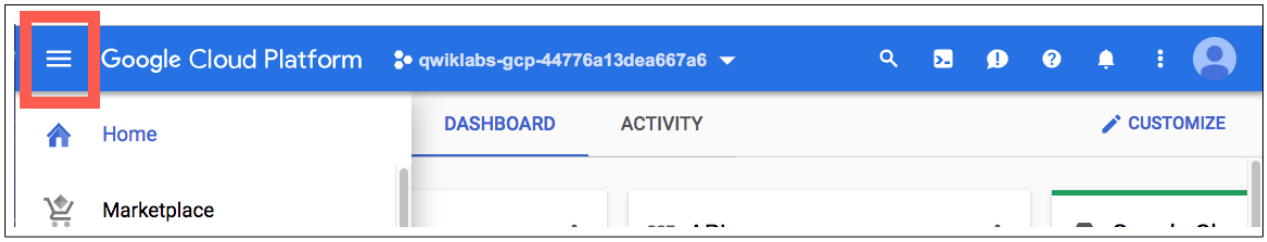


4. Click through the subsequent pages:

    - 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.

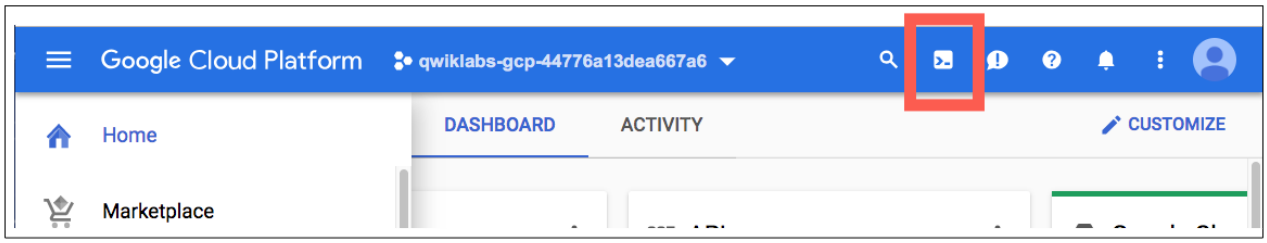
Note: You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-left.



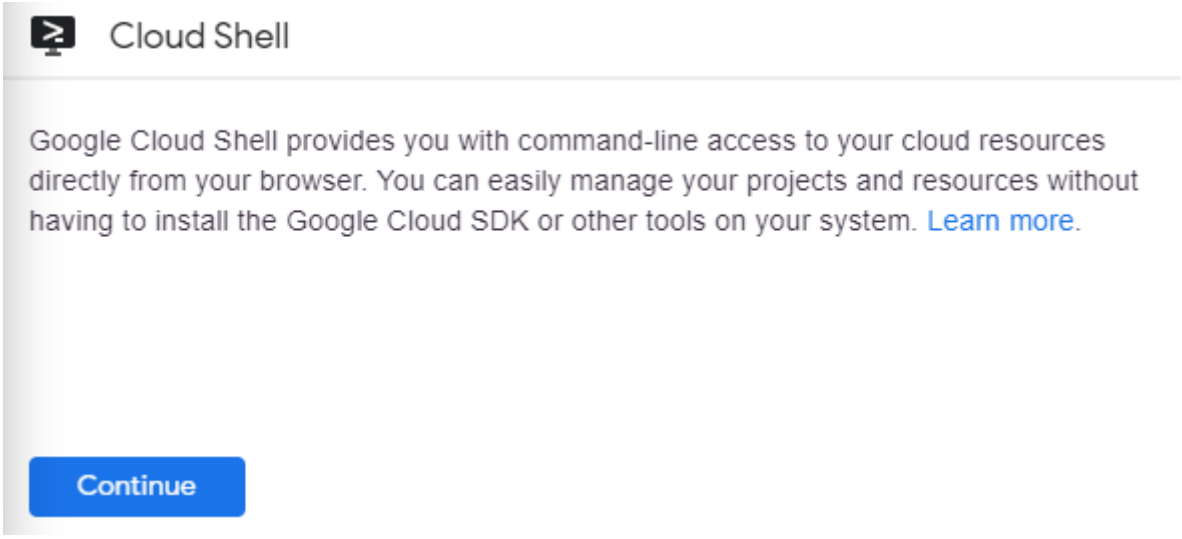
Activate Cloud Shell

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.

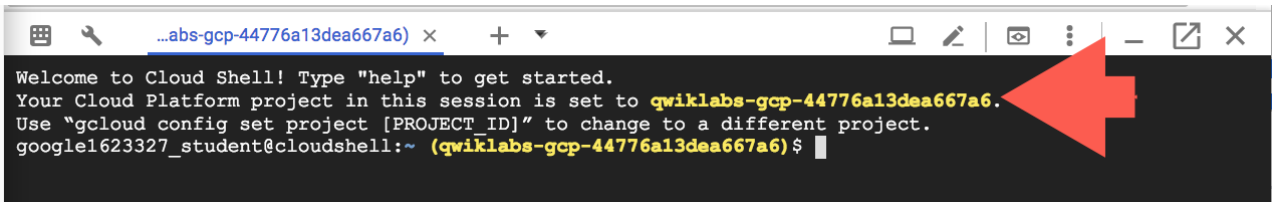
In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.



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:



`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.

Create an environment variable for your Project ID, which you can find on the lab page:

```
export PROJECT_ID=<YOUR_PROJECT_ID>
```

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 --project $PROJECT_ID
```

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

Next, run:

```
gcloud app deploy --project $PROJECT_ID --version 1
```

Enter **Y** to continue.

```
gcloud datastore indexes create --project $PROJECT_ID index.yaml
```

Enter **Y** to continue.

Monitor indexes

Monitor the indexes by going to **Navigation menu** > **Datastore** > **Indexes** in the Cloud 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

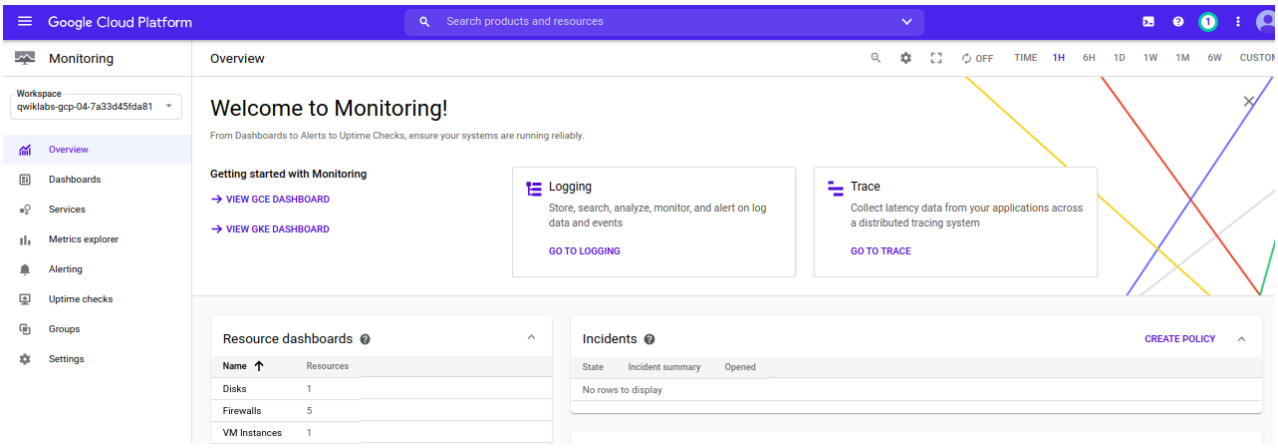
Prepare to install Cloud Monitoring

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

Install the Monitoring and Logging agents

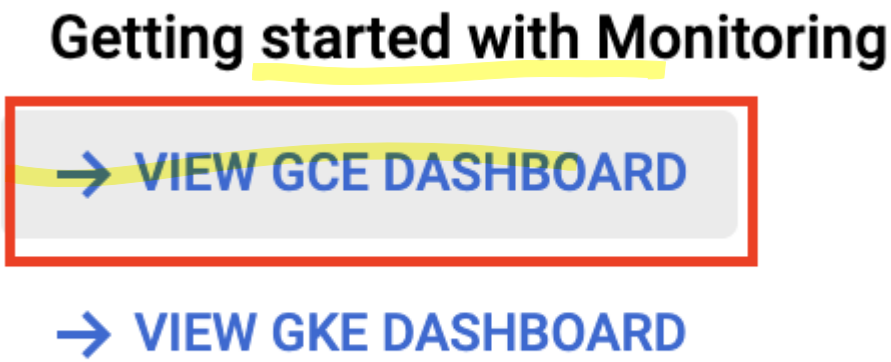
Agents collect data and then send or stream info to Cloud Monitoring in the Cloud Console.

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.

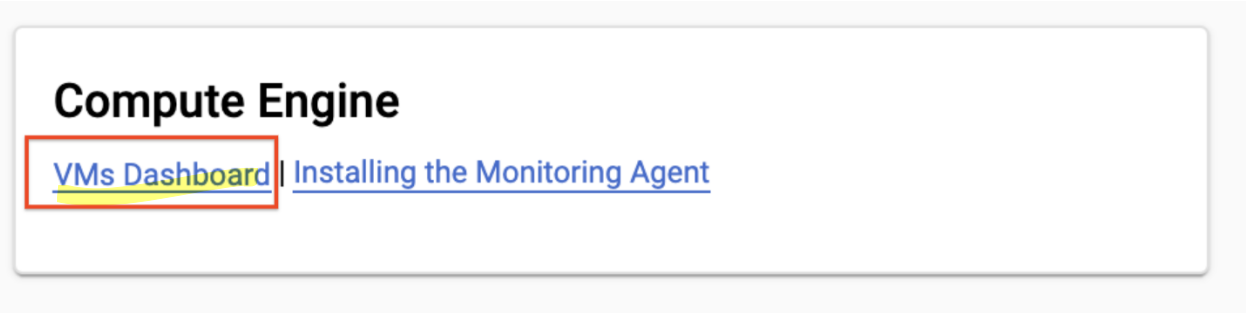
The *Cloud Logging agent* streams logs from your VM instances and from selected third-party software packages to Cloud Logging. It is a best practice to run the Cloud Logging agent on all your VM instances. Learn more.

To install the agents on the VM:

1. From the **Navigation Menu** go to **Monitoring**.
2. In the Monitoring Overview window, click **VIEW GCE DASHBOARD**.

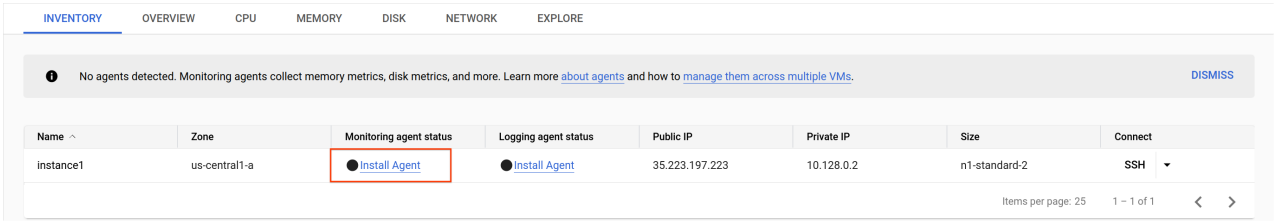


3. Click on the **VMs Dashboard** link under **Compute Engine**.



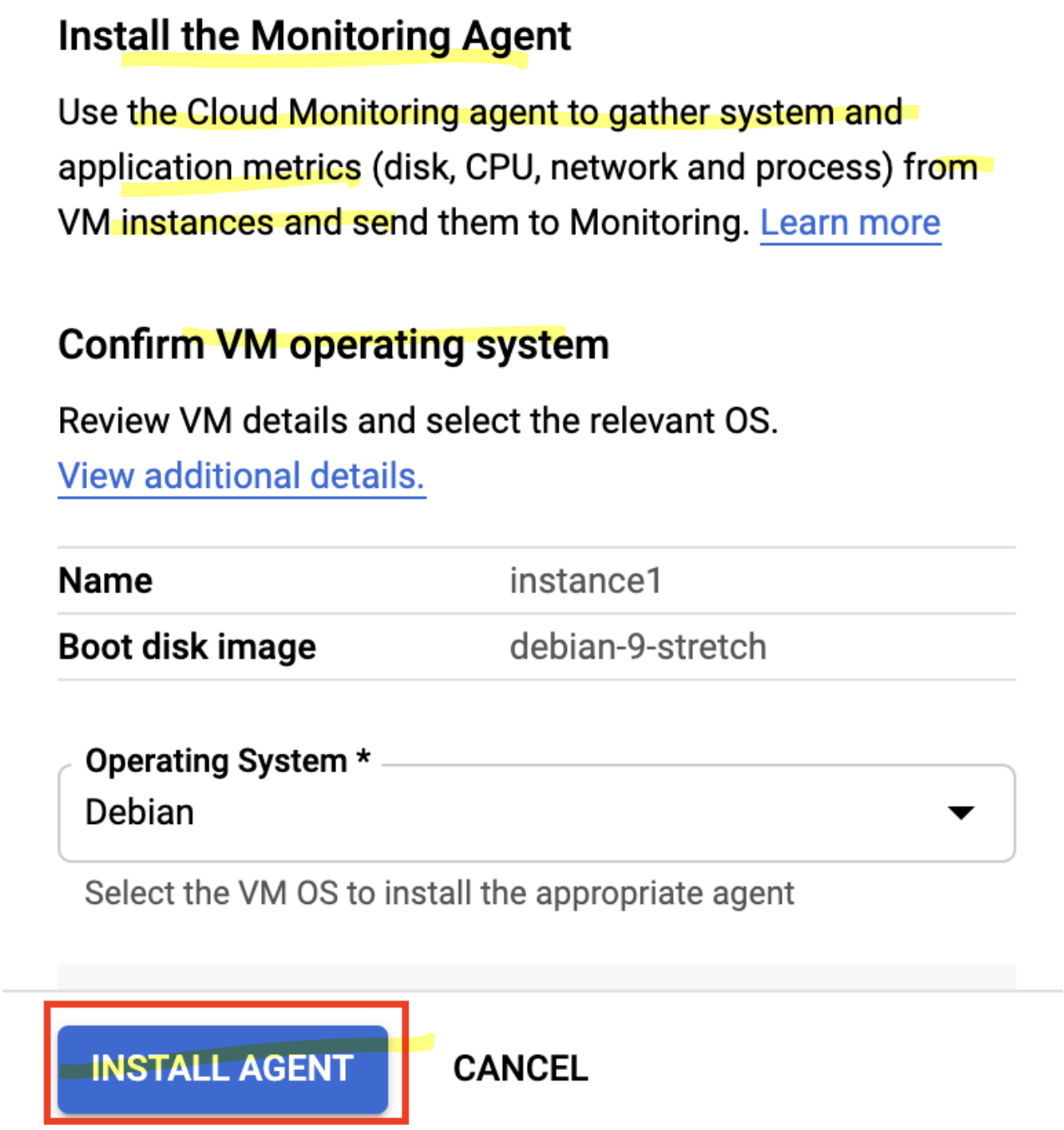
You will see your VM, **instance1**, listed.

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



This will bring up an **Agent Details** window on the right side, which is used to install the agent.

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



This will generate a command that will let you SSH into your VM and installs the agent and place it into your `gcloud` command line. Press **Enter** to run the command. You can also see the command in the Agent details area if you need to copy it and paste into the command line.

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 under Monitoring Agent Status.

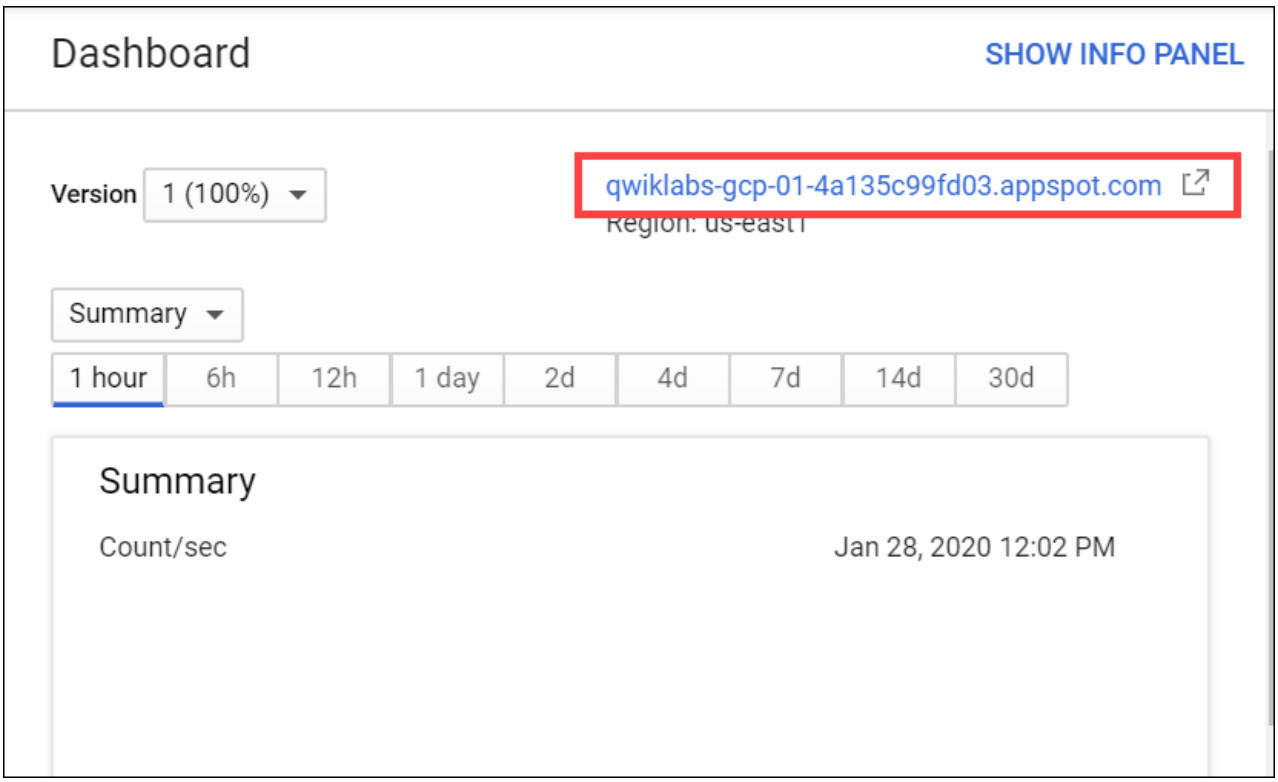
Additional resources for the lab

Since this lab uses logs, it helps if you had some interesting logs. You will 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. From the **Navigation menu** go to **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 `https://` and the last `/` from the URL.
Path:	/?food=pizza
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**.

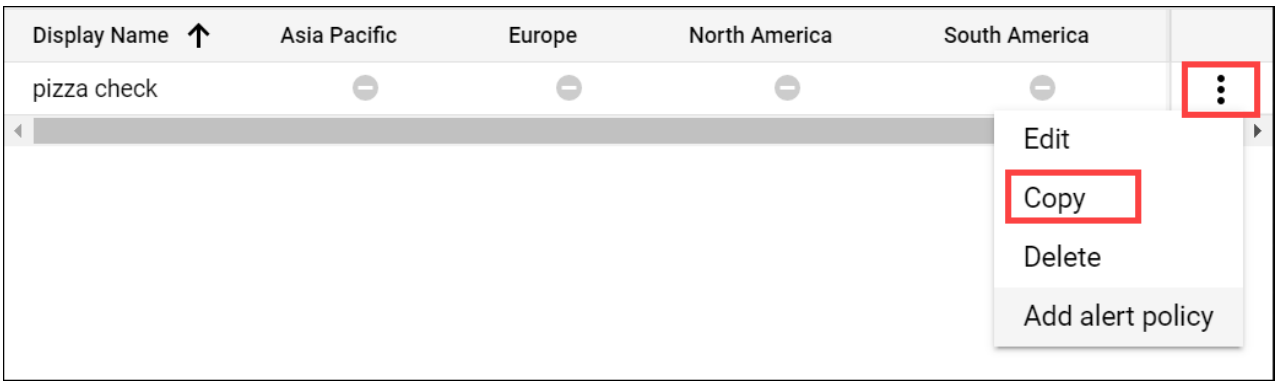
You should see a successful response.

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.

Create an uptime check

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.

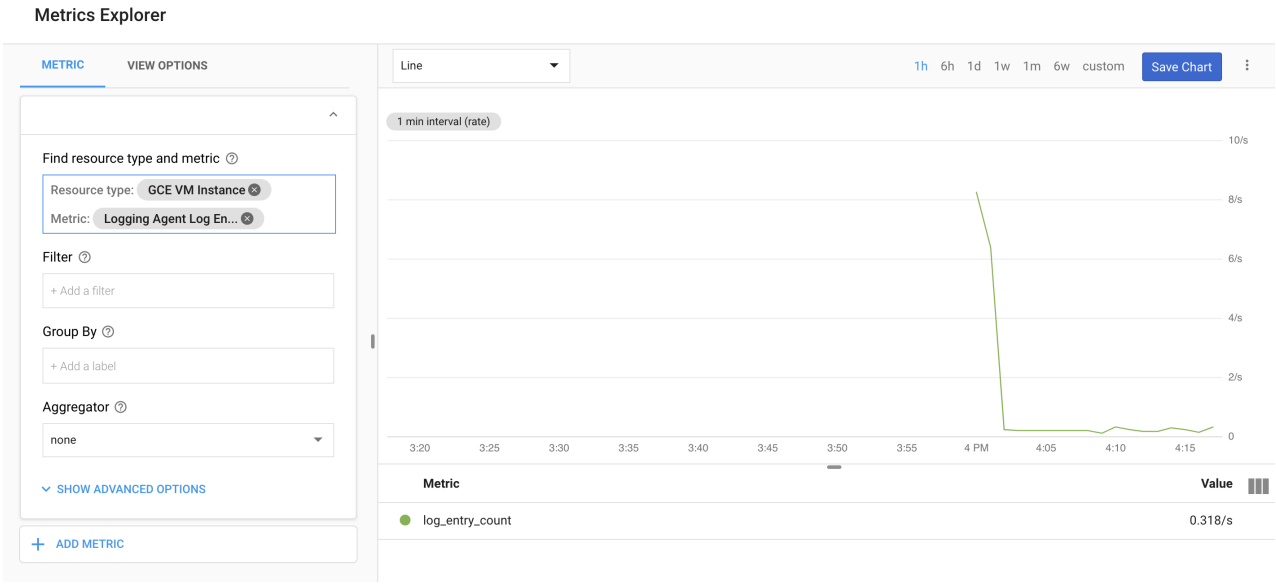
Metrics around logs based metrics

- `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 `dropped_log_entry_count` + `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 **. Choose the option for GCE instances.
3. Select **VM Instance (GCE)** 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

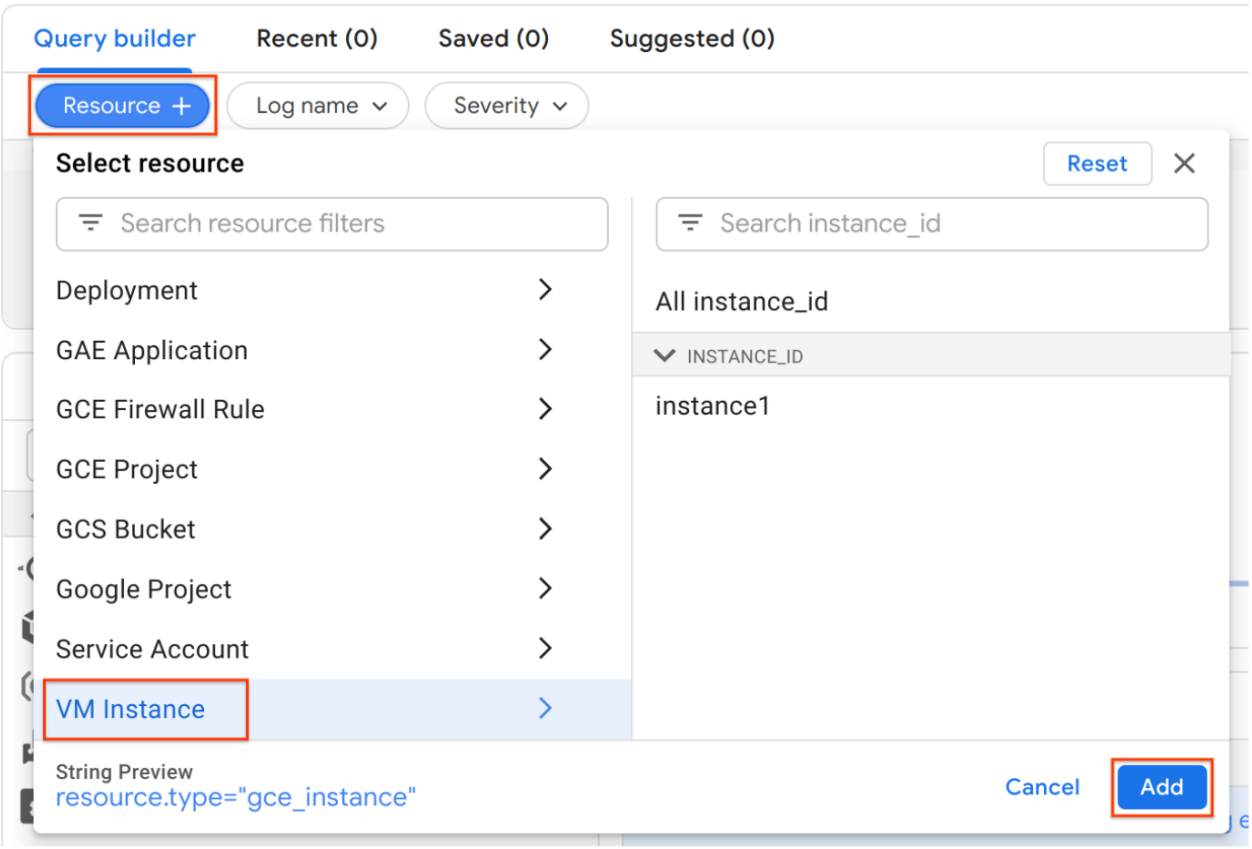
Create your own logs-based metrics using data from existing logs. These are called **user defined logs-based metrics**. In this section, you create a metric using a log entry.

1. Select **Navigation menu** > **Logging** > **Logs Explorer**.

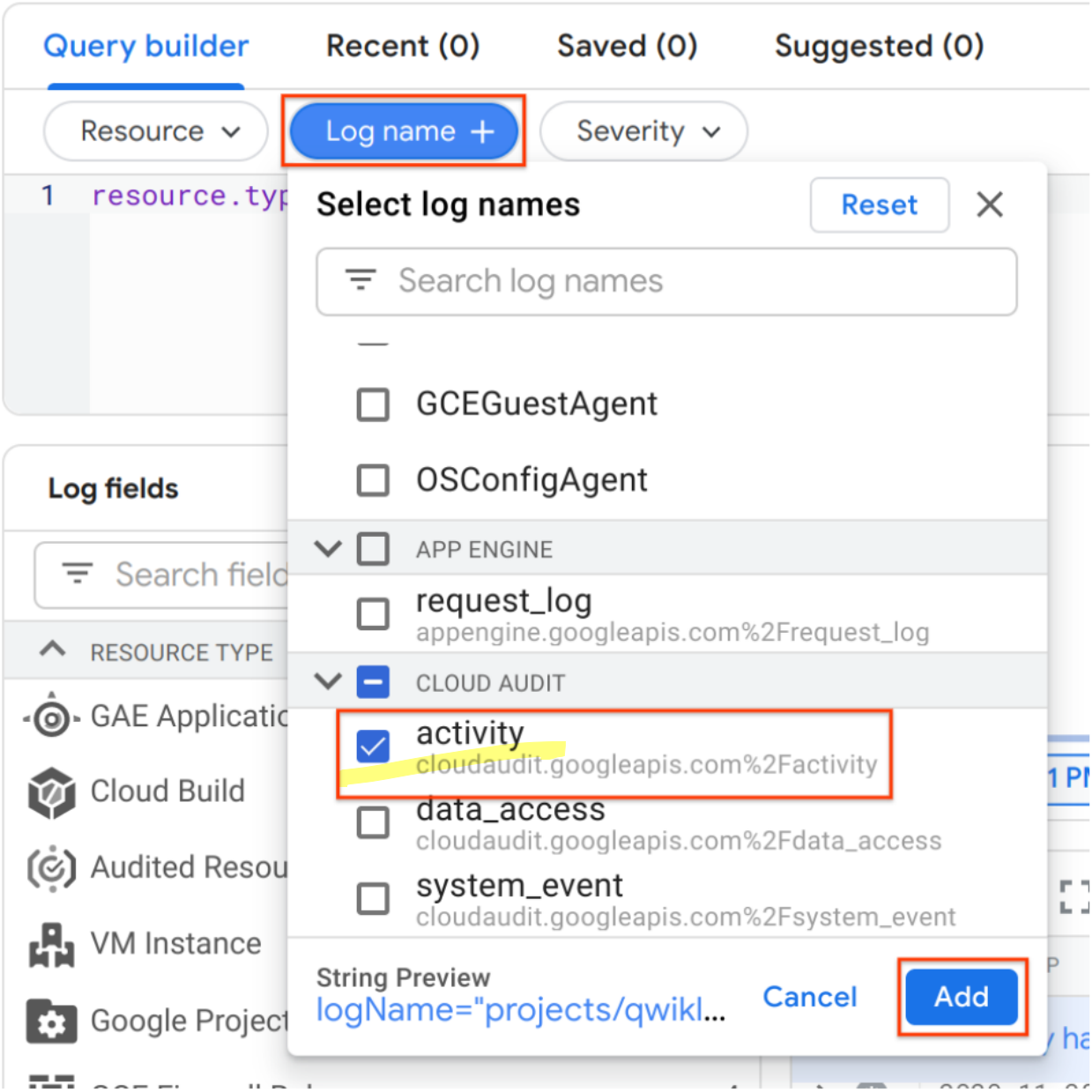
The Cloud Logging window opens.

2. Filter for activity logs from VM instance you previously created.

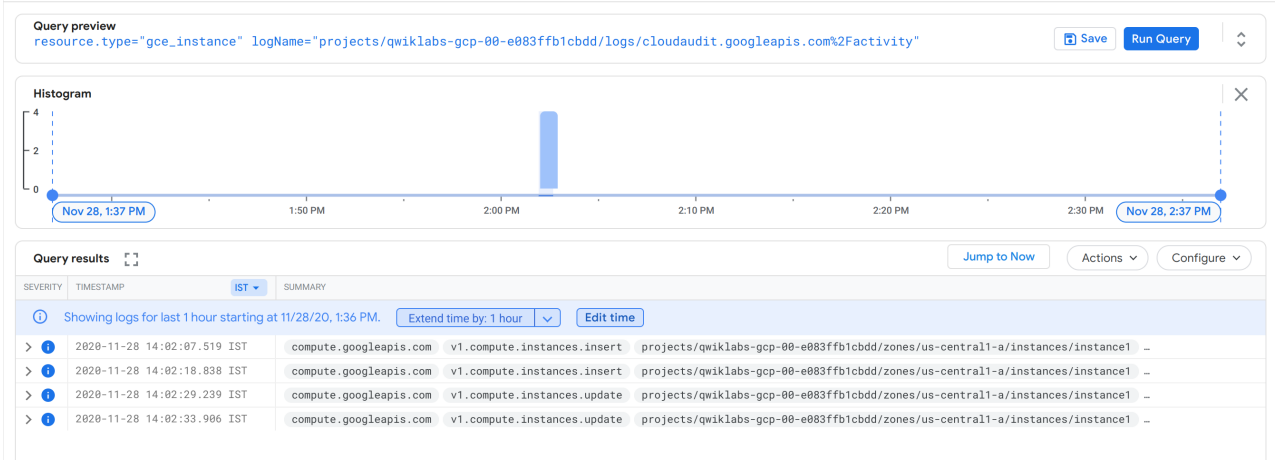
   In the **Resource** dropdown, select **VM Instance** and click **Add**.



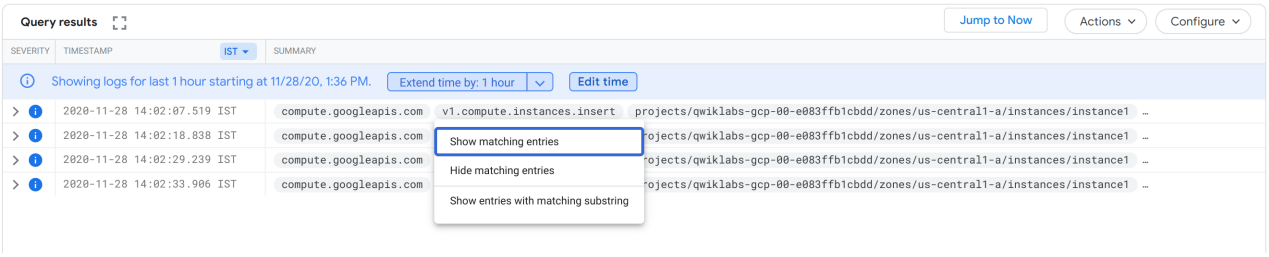
In the **Log name** dropdown, select **activity**, then click **Add**.



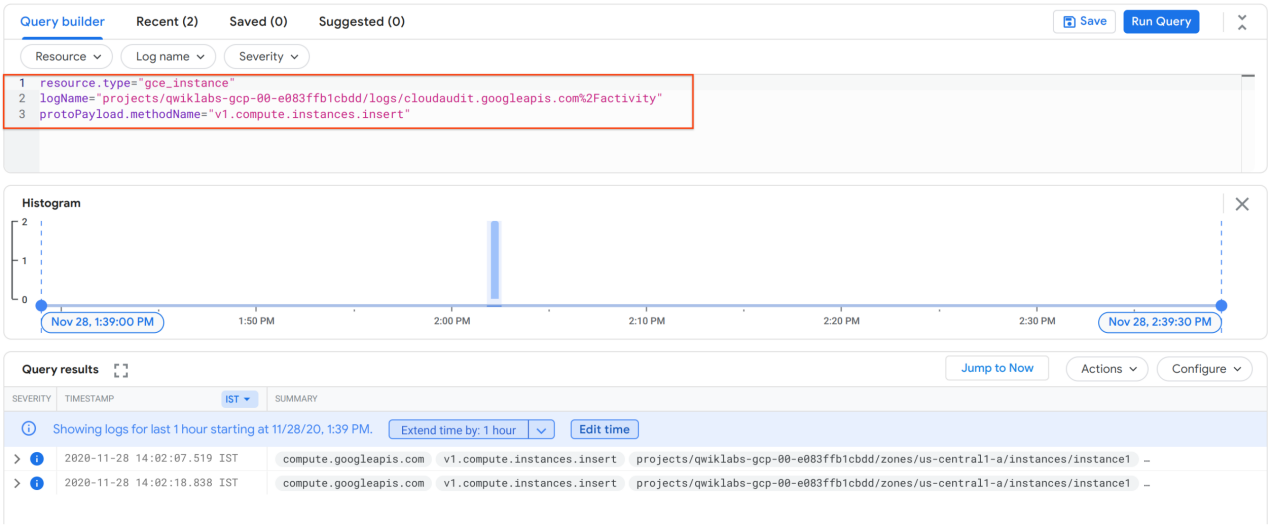
Click **Run Query**.



3. 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.

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

5. For the Log Metric Name type "newVM" then click **Create Metric**.

You will see a success message at the top of the page.

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 scroll up 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.

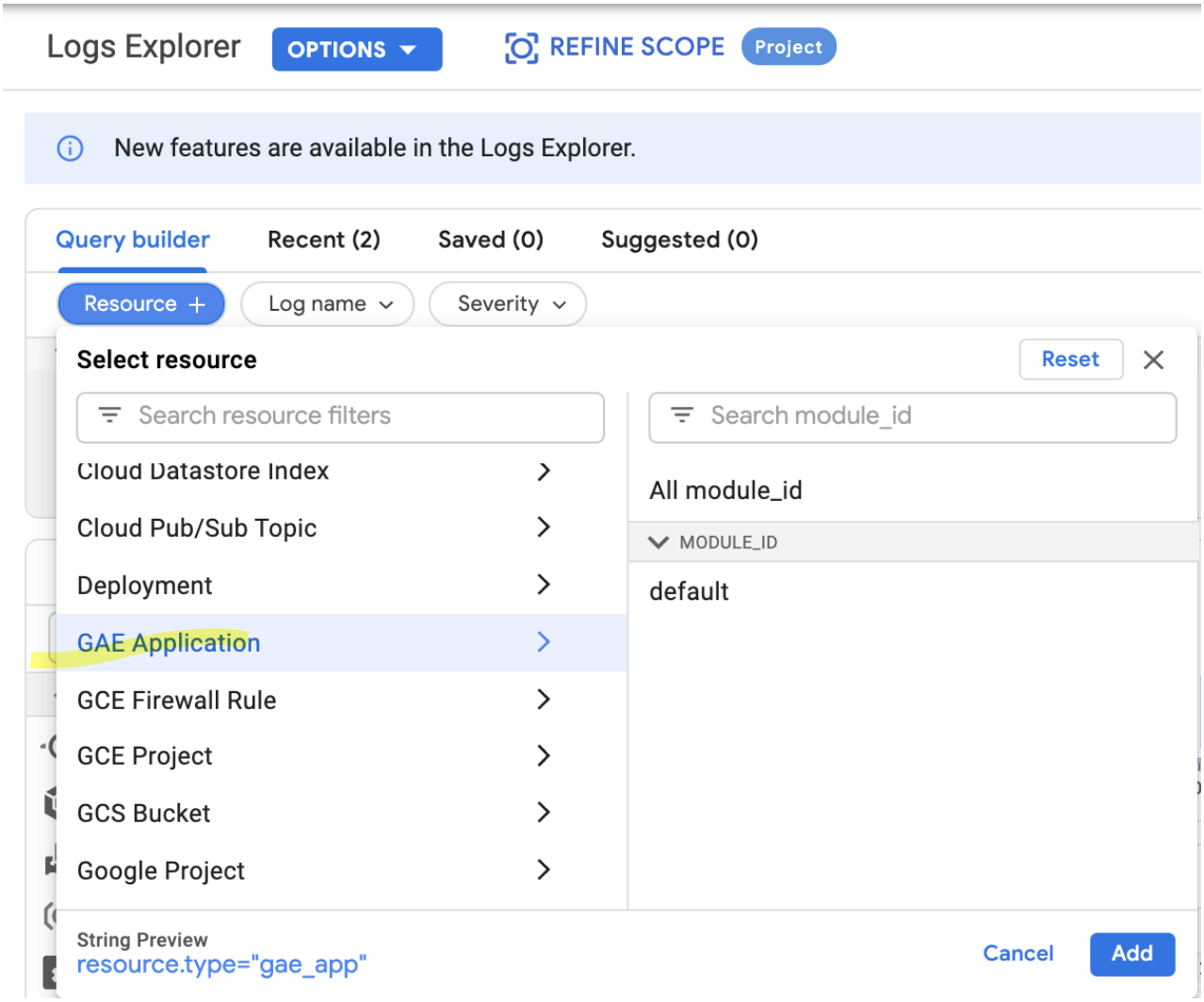
Create user defined logs-based metrics

Labels and user defined metrics

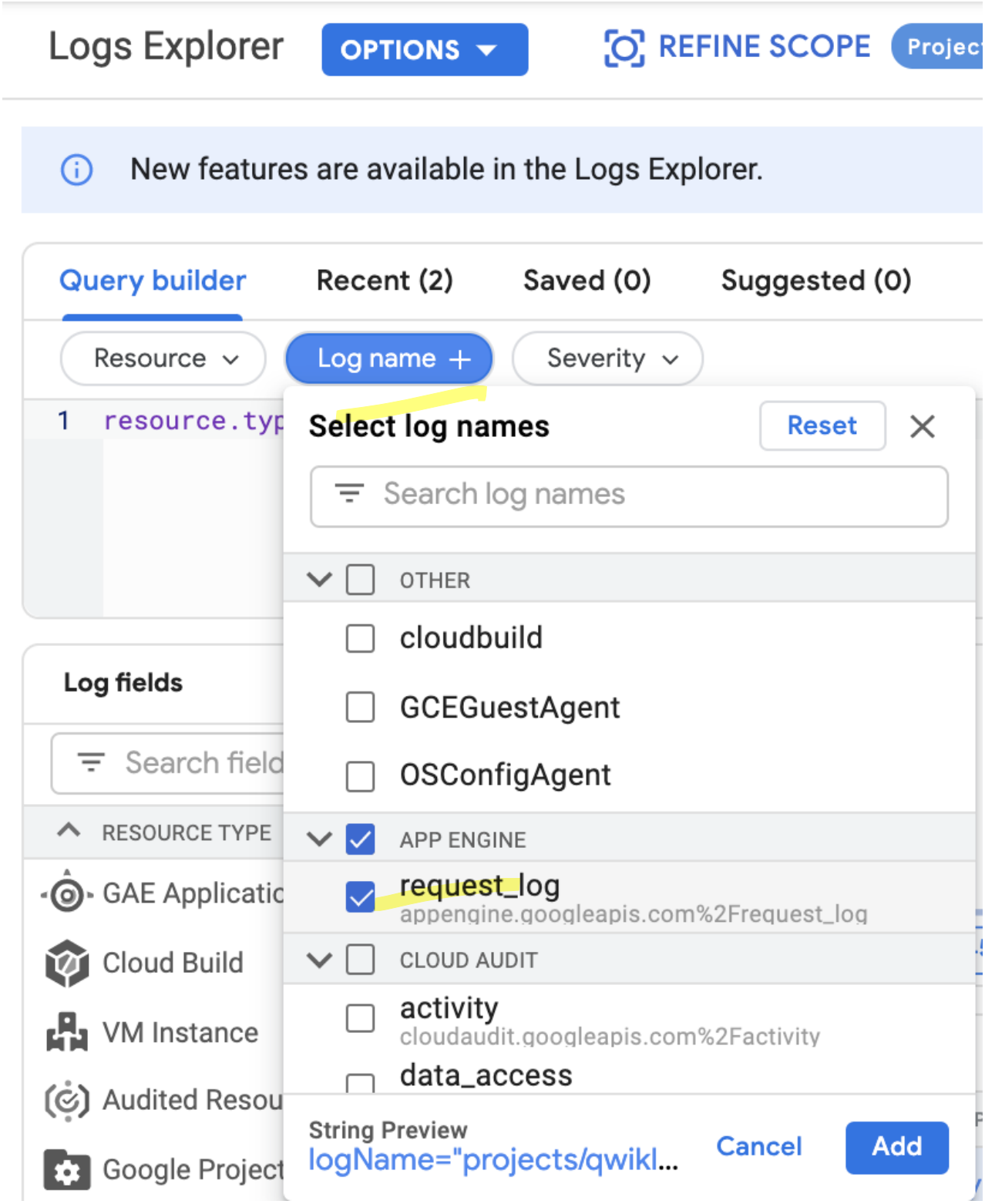
User defined labels can be created when you create a metric. An extractor expression is required for each configured label to tell Cloud Logging how to extract values from logs and place them as the labels' value. You cannot add labels to system logs-based metrics.

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**.



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

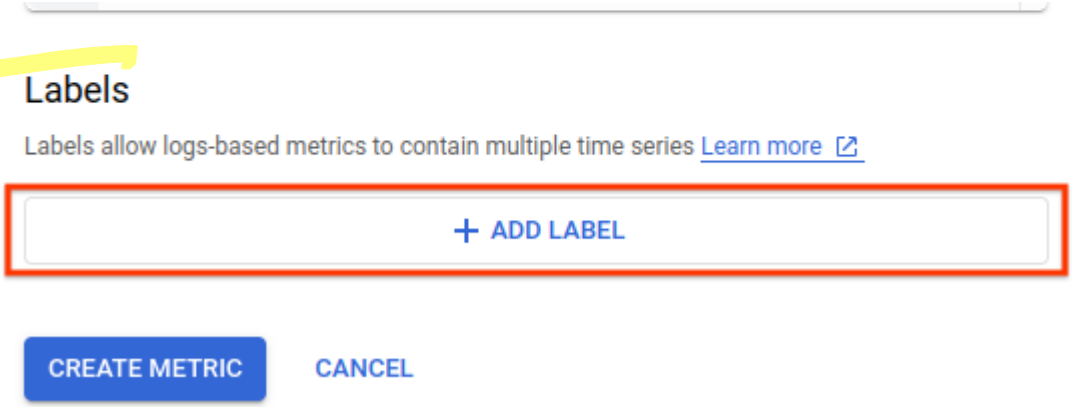
Filter selection

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

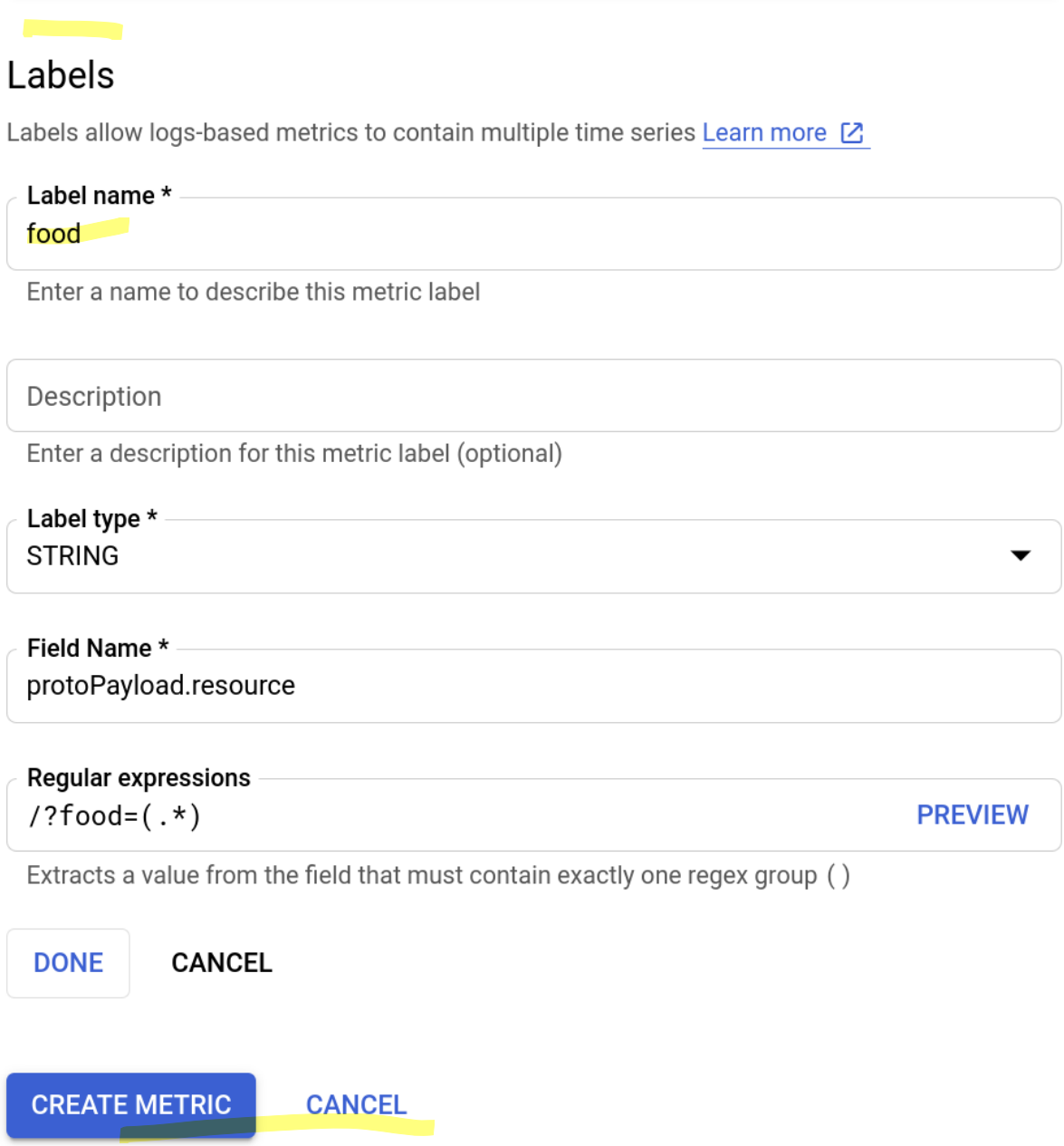
Labels allow logs-based metrics to contain multiple time series — one for each label value. All logs-based metrics come with some default labels.

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:

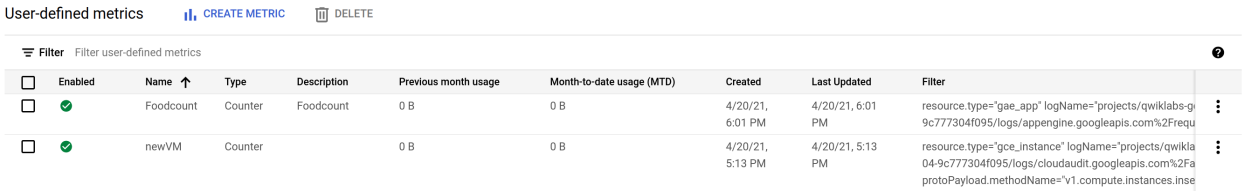
- Label name: food
- Label type: String
- Field Name: search for "resource" and choose **protoPayload.resource**.
- Regular expressions: `/?food=(.*)`
- Click **Done**



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**.

If you click on **Logs-based Metrics** on the left side you'll see your user-defined metric added to the logs-based metrics screen.



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

Create labels and user defined metrics

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:

- In the Find resource type and metric field, start typing "logging/user/" and select **logging/user/Foodcount**. If this metric doesn't appear, close the window and try again.
- For Configution, 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.

Create the Foodcount alerting policy

Custom dashboard with heatmap

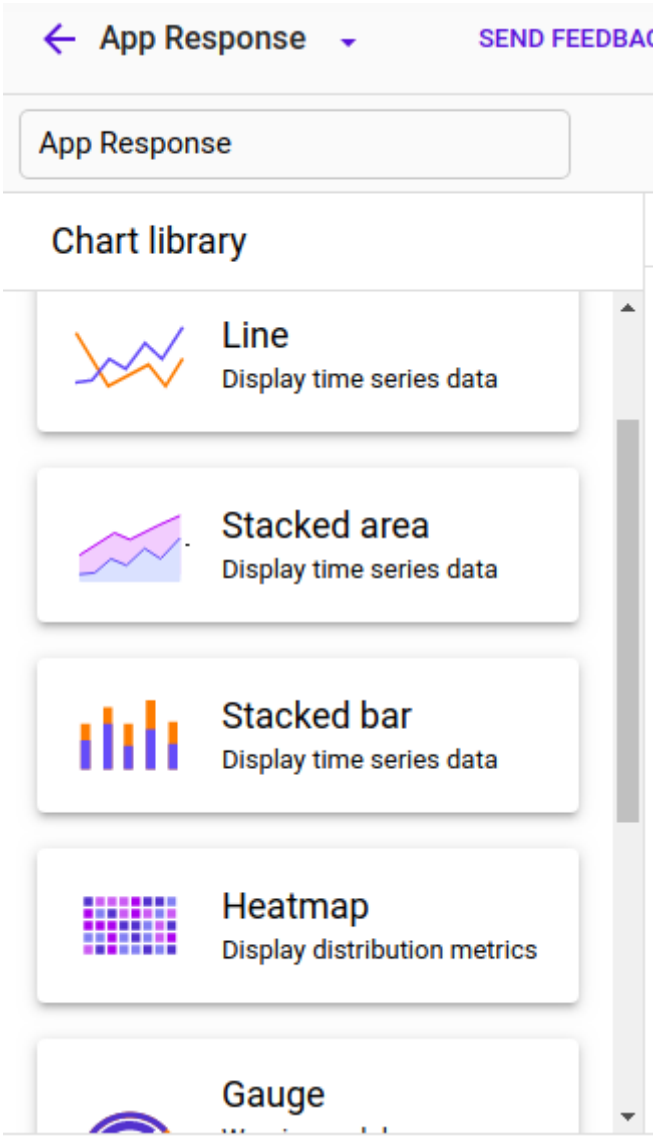
Creating a custom dashboard to show you a specific metric is a great way to see your data. Next you'll create a custom dashboard to show the same latency metric you just used.

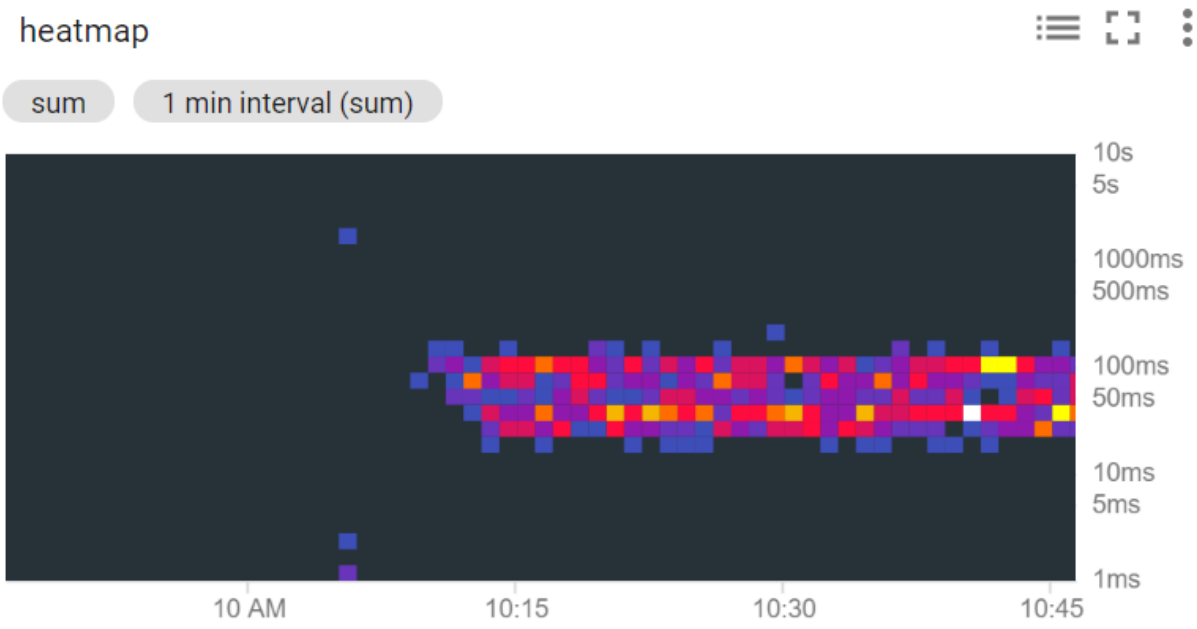
1. Still on the Cloud Monitoring window, in the left menu, click **Dashboards** > **Create Dashboard**.

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".





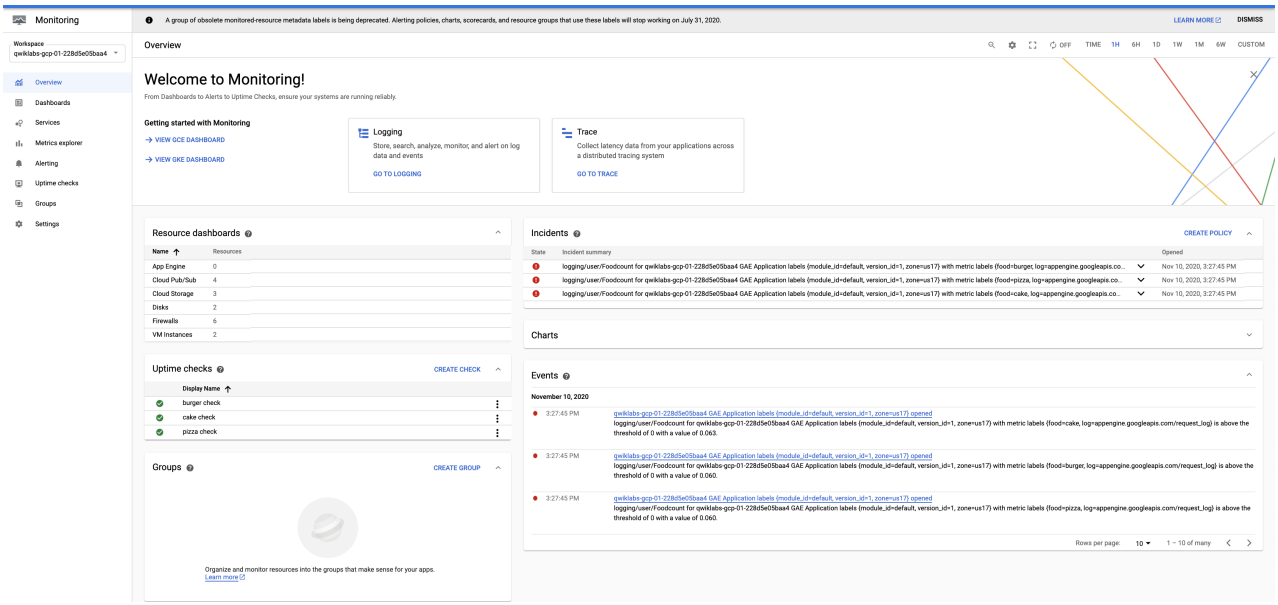
Check for the alert

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 newVM 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.



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 May 3, 2021

Lab Last Tested May 3, 2021

Continue questing

Lab

Cloud Monitoring: Qwik Start