



## Optimizing the Costs of Monitoring



In our final module we discuss optimizing the costs for Google Cloud's operations suite.

---

# Agenda

The Costs of Monitoring

Bill Estimation

Cost Control Best Practices



Specifically, you will learn to analyze resource utilization costs for operations related components within Google Cloud, and implement best practices for controlling the cost of operations within Google Cloud

---

# Agenda

The Costs of Monitoring

Bill Estimation

Cost Control Best Practices



Let's start with the costs.

## Currently Free



### Free Stuff

- |  |   |
|--|---|
| <ul style="list-style-type: none"><li>• Debugger and Profiler</li><li>• Cloud Audit and Transparency logs</li><li>• BigQuery Data Access log</li><li>• Log exclusions</li><li>• Exporting logs (\$\$ at resource)</li><li>• Dashboards</li></ul> | <ul style="list-style-type: none"><li>• Google Cloud metrics</li><li>• Anthos metrics and On-Prem to Google log streams</li><li>• App Engine Standard Trace spans</li><li>• Uptime checks</li></ul> |
|--|---|



At the time of this writing, there are a number of parts of the Google operations suite that are free, including:

- Debugging and using the Profiler
- Collecting and using the Cloud Audit and Transparency logs
- BigQuery Data Access logs
- Anything excluded from logs
- Exporting, though what you export to might incur spend
- Creating and using dashboards
- The metrics themselves
- Anthos metrics and log streams
- App Engine standard trace spans
- Uptime checks

## Free Allotments



### Per Month

- Logging: 50GiB/project
- Monitoring: 150MiB/billing account
- Monitoring API calls: 1 mil/project
- Trace ingestion: First 2.5 mil spans/project
- Trace spans scanned: 25 mil spans/project



Other parts of the operations suite have a free tier to get you started. Examples include:

- The first 50Gib of logging per project
- The first 150MiB of monitoring data per billing account
- The first million monitoring API calls per project
- The first 2.5 million trace spans ingested per project
- The first 25 million trace spans scanned per project

## Pricing



### After Free Allotment

- |   |   |
|---|---|
| <ul style="list-style-type: none"><li>• Logging/project: \$0.50/GiB</li><li>• Monitoring/billing account<ul style="list-style-type: none"><li>◦ \$0.285/MiB: &lt;100TiB</li><li>◦ \$0.151/MiB: 100-250TiB</li><li>◦ \$0.061 &lt;250TiB</li><li>◦ API calls: .01/1,000</li></ul></li></ul> | <ul style="list-style-type: none"><li>• Trace ingestion/billing account: \$.20/million spans</li><li>• Trace spans scanned: .02/million spans</li><li>• Log storage will go to \$.01/GiB in April, 2021</li></ul> |
|---|---|



After the free tier, the prices for the various operation related fees you can see on the slide. For the latest pricing information, please always check the [Google Documentation](#).

## Network Telemetry Pricing

- Network telemetry logs generate spend
  - VPC Flow, Firewall Rules, and Cloud NAT logging
- Logs stored in Cloud Logging are free to generate, but billed for storage
- Exported logs are billed as follows:

Log Generation	Price
0–10 TB per month	0.50/GB
10–30 TB per month	0.25/GB
30–50 TB per month	0.10/GB
>50 TB per month	0.05/GB



The Networking logs including, VPC Flow, Firewall Rules, and Cloud NAT, will cost you the standard log storage fees (\$0.50/GiB), but they won't cost you anything extra to generate, provided you are storing them in Cloud Logging.

If you export the network telemetry logs to an external service, then the table on the bottom of the slide shows how much the logs will cost to generate, above and beyond any destination or networking fees.

---

# Agenda

The Costs of Monitoring

Bill Estimation

Cost Control Best Practices

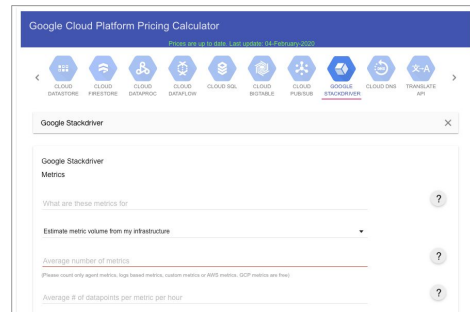


Ok, we've covered some of the pure costs, now let's talk about bill estimation.



## Bill Estimation

- Google Cloud Pricing Calculator is accurate
  - But requires good data
- Start with current usages
- Several ways to extract the data

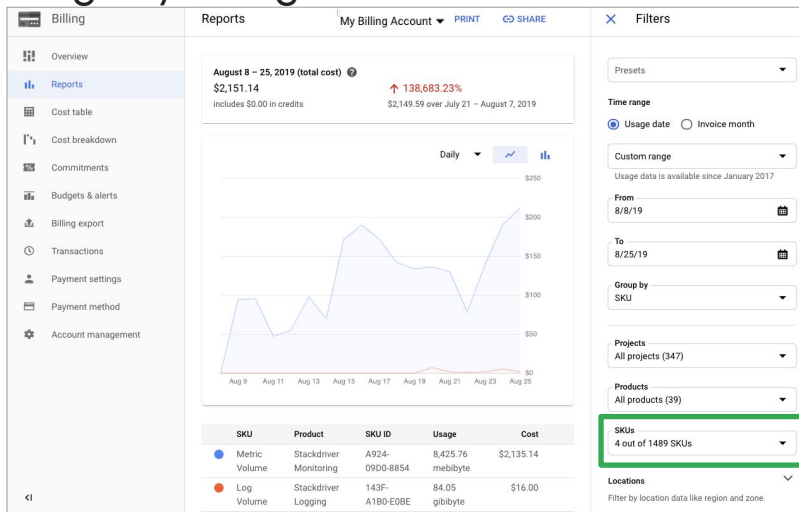


The goto page, when trying to estimate prices in Google Cloud, should always be the Pricing Calculator. The pricing calculator is accurate, but it's only as accurate as the data you provide it.

If you have operation services already running the Google Cloud, start by pulling the prices of what you're spending on them currently.

There are several places you'll need to look for the requisite data

## Usage by Billing Account



Filter to SKUs

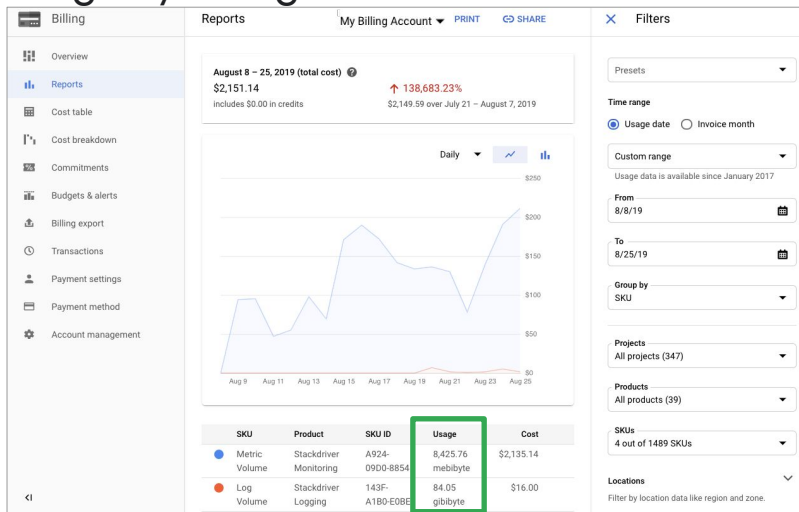
- Log Volume
- Spans Ingested
- Metric Volume
- Monitoring API Requests



Start by going to your billing account reports page. Set your date range and then filter by SKU. The SKUs you want are:

- **Log Volume**
- **Spans Ingested**
- **Metric Volume**
- **Monitoring API Requests**

## Usage by Billing Account



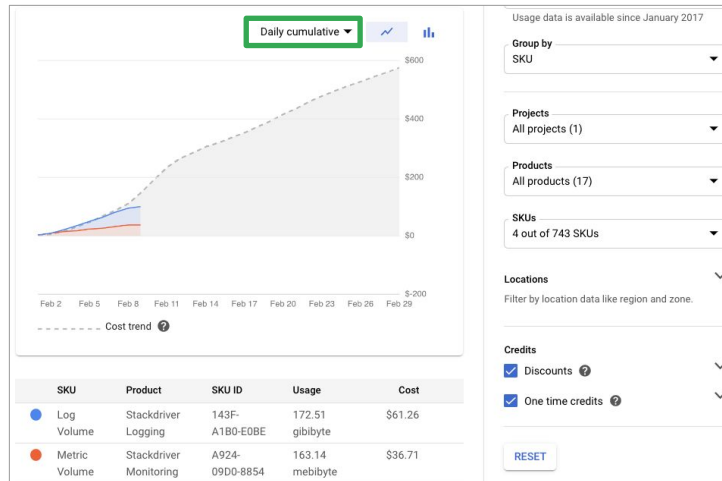
Filter to SKUs

- Log Volume
- Spans Ingested
- Metric Volume
- Monitoring API Requests

Then check your usages and costs.

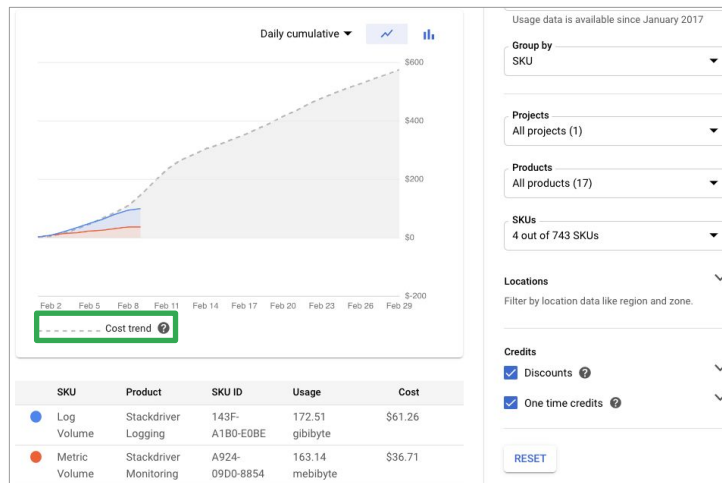
Note, if SKU usage is 0 for a metric, then it won't appear in list.

## Usage by Billing Account, Cost Trend



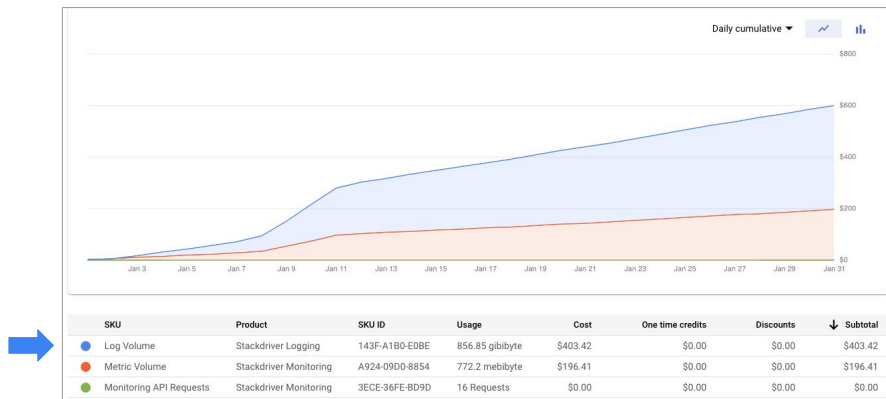
Here's another view of the same page. In the chart, you can change the view to display the data Daily, Monthly, Daily Cumulative, or Monthly Cumulative.

## Usage by Billing Account, Cost Trend



When you put it in Daily cumulative, you can see the cost trend line does a nice job showing where we're headed based on current spend trends. If you recently added logging, this might be a good way to estimate what your bill might do.

## Last Month's Spending (What's the Log Vol. from?)

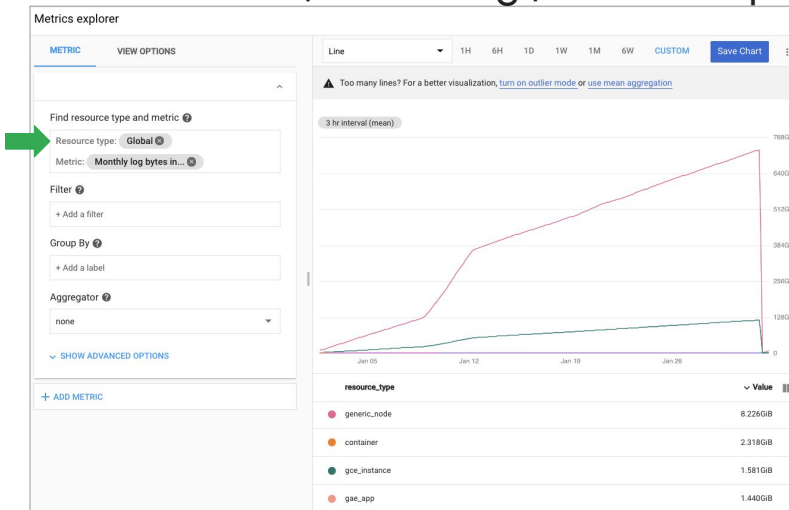


You can do the same sort of thing to see how money was spent in the past month. In this case, the biggest line item is log volume.

It begs the question. If I'm spending money on logging data, where exactly is that data coming from?

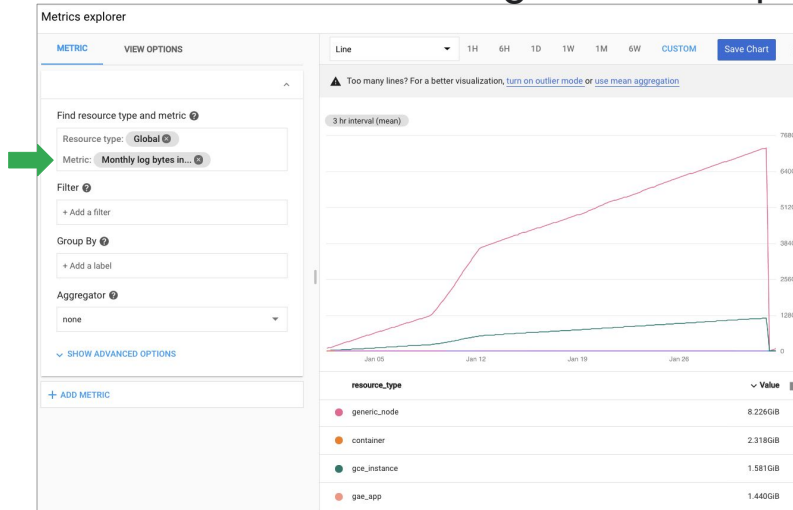
To find the answer, check the Metrics Explorer.

## Metric Details (Monitoring | Metrics Explorer)



Open the Monitoring | Metrics Explorer, and set the Resource type to Global.

## Metric Details (Monitoring | Metrics Explorer)



### Helpful Metrics

- Log bytes ingested
- Monthly log bytes ingested
- Metric bytes ingested
- Trace spans ingested
- Monthly trace spans ingested

Then, depending on what you're looking for, set the Metric to one of the following:

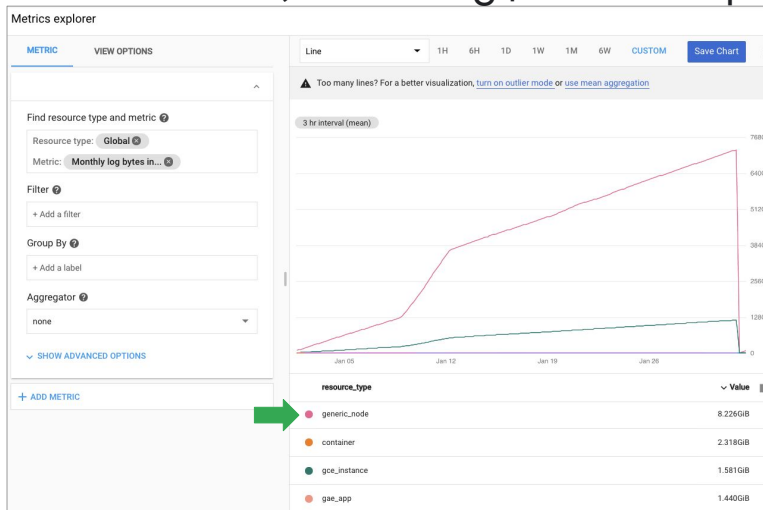
- [Log bytes ingested](#): Log bytes ingested in Logging.
- [Monthly log bytes ingested](#): Each point represents the month-to-date sum of log bytes ingested in Logging. The monthly total is available on the last day of the month, when it also resets.
- [Metric bytes ingested](#): Chargeable number of bytes of metric data ingested in Monitoring.
- [Trace spans ingested](#): Chargeable trace spans ingested in Trace.

and

- [Monthly trace spans ingested](#): Each point represents the month-to-date sum of trace spans ingested in Trace. It resets on the last day of the month; the monthly total is found on the last day of the month.



## Metric Details (Monitoring | Metrics Explorer)



### Helpful Metrics

- Log bytes ingested
- Monthly log bytes ingested
- Metric bytes ingested
- Trace spans ingested
- Monthly trace spans ingested

This example was built on a project doing a lot of VM Migration work. The `generic_node` is all the logging data coming from Migrate for Compute Engine (formerly Velostrata). You can find more info on Migrate for GCE [here](#).

## Logging Usage by Project (Logging | Resource Usage)

Resource usage

LOGS ENABLED

CREATE USAGE ALERT

Last month's ingested log volume  
856.85 GiB  
Total for the month of January. [See bill](#)

This month's ingested log volume  
176.91 GiB  
since the first of the month.

Excluded log volume  
0 B  
since the first of the month.

Projected ingestion log volume ⓘ  
596.09 GiB  
by end of month

Ingestions

Exclusions

Logs Ingestion

Resource ⓘ

Previous Month Usage ⓘ

▼ Ingested (MTD) ⓘ

Excluded (MTD) ⓘ

Projected (EOM) ⓘ

Ingestion Status ⓘ

Generic Node731.94 GiB148.79 GiB0 B501.35 GiB✔ All ingested

GCE VM Instance119.04 GiB27.39 GiB0 B92.3 GiB✔ All ingested

Cloud VPN Gateway1.64 GiB466.45 MiB0 B1.53 GiB✔ All ingested

GKE Container2.33 GiB219.83 MiB0 B740.72 MiB✔ All ingested

Service Account150.06 MiB27.43 MiB0 B92.44 MiB✔ All ingested

Cloud NAT Gateway95.66 MiB22.05 MiB0 B74.29 MiB✔ All ingested



To view details of the logging usage in a Google Project, visit the Logging | Resource Usage page.

The Resource usage page displays a summary of your logs ingestion data for the selected Google Cloud project. This data includes the previous month's total usage, the current month's to-date usage, and projected usage for the current month.

To review your project-level billing data in detail, click [See Bill](#) under Last month's ingested logs volume. This takes you to the Cloud Billing Reports page we saw earlier.

Below that, the Logs ingestion table displays a summary of your logs ingestion data by resource type. This data includes the previous month's total usage, the current month's to-date usage, and projected usage for the current month.

## Logging | Logs-Based Metrics

User-defined Metrics					
User defined logs-based metrics that count the number of log entries that match a given filter.					
<input type="text" value="name:"/> <span>×</span>					
Name ^	Type	Description	Previous Month Usage	Usage (MTD)	Filter
user/200Responses	Counter	Test - 200 responses from Uptimesss	0 B	0 B	resource.type="http_load_balancer" httpRequest.status=200 httpRequest.userAgent="GoogleStackdriverMonitoring-UptimeChecks(https://cloud.google.com/monitoring)"
user/404Errors	Counter		225.23 KB	217.17 KB	resource.type="gae_app" logName=("projects/cloud-logs-test-project/logs/appengine.googleapis.com%2Fstderr" OR "projects/project/logs/appengine.googleapis.com%2Fnginx.request" OR ")



To see your logs-based metrics usage, head over to [Logging | Logs-Based Metrics](#).

Previous Month Usage represents the sum of bytes ingested in the logs-based metric in the previous calendar month.

Usage (MTD) represents the sum of bytes ingested in the logs-based metric in the current calendar month.

Clicking any of the column names lets you sort data in ascending or descending order. This is helpful, for example, if you want to review which metrics ingest the most data.

## Monitoring | Settings

[SUMMARY](#) [AGENTS](#) [EMAIL REPORTS](#) [KUBERNETES MIGRATION STATUS](#)

### Overall usage

<b>Overall metrics ingested</b> 154.38MB Month to date	554.01MB Previous month	<b>Monthly projection</b> 700.34MB By end of month	↑ 26.41% Versus previous month <a href="#">View Bill</a>
--	----------------------------	--	---

### GCP Projects

Project name ↑	Project ID	Type	Previous month	Month to date	Monthly projection	
▶ VelosSandbox	velossandbox	Host project	554.01MB	154.38MB	700.34MB	⋮

[ADD GCP PROJECTS](#)



Early on in this series of modules, we learned that a Workspace is used in Cloud Monitoring to monitor the resources you care about, whether they are in a Google Cloud project, an AWS account, or multiple Google Cloud projects and AWS accounts. To view your Monitoring usage by workspace, go to Monitoring | Settings.

On the Summary tab, the **Metrics Ingested** table displays a summary of your metrics ingestion data by resource. This data includes the previous month's total usage, the current month's to-date usage, and projected usage for the current month.

To get your project-level usage in detail, click **View Bill** in the **Metrics Ingested** table. This takes you to the Cloud Billing Reports page.

---

# Agenda

The Costs of Monitoring

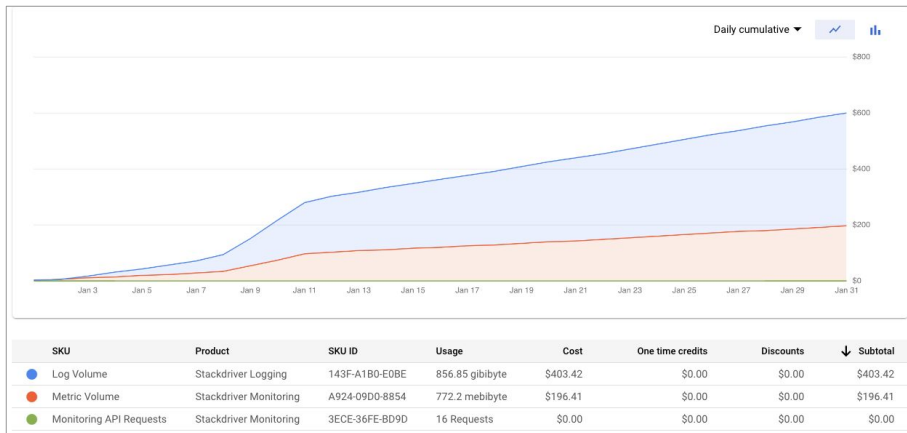
Bill Estimation

Cost Control Best Practices



Finally, let's discuss some billing best practices.

## Know Your Costs!



Start with what we've already discussed in this module, mostly in the last section. Know your costs. Know your operations related spend, and exactly where it goes. Know what the big-ticket items are.

## Know What You're Paying For



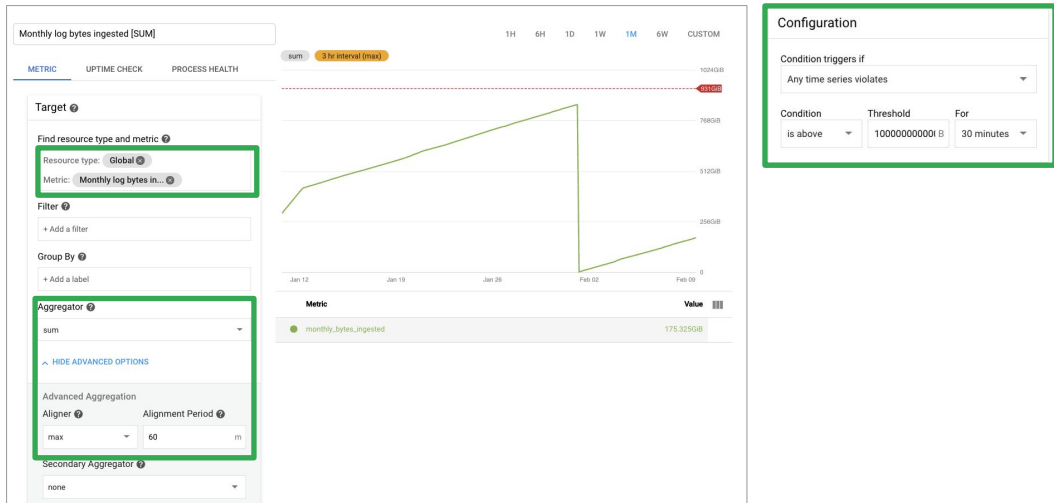
### After Free Allotment

- |   |   |
|---|---|
| <ul style="list-style-type: none"><li>• Logging/project: \$0.50/GiB</li><li>• Monitoring/billing account<ul style="list-style-type: none"><li>◦ \$0.285/MiB: &lt;100TiB</li><li>◦ \$0.151/MiB: 100-250TiB</li><li>◦ \$0.061 &lt;250TiB</li><li>◦ API calls: .01/1,000</li></ul></li></ul> | <ul style="list-style-type: none"><li>• Trace ingestion/billing account: \$.20/million spans</li><li>• Trace spans scanned: .02/million spans</li></ul> |
|---|---|



Know what exactly you're paying for.

# Use Alerts



If you're adding a new type of logging, create alerts to spot runaway spend.

In the example you see on the current slide.

The Resource type is Global, and the metric is Monthly log bytes ingested. So this is our logging data which, thanks to the last slide, we know will be billed at \$0.50/GiB.

To get the total amount, the Aggregator is set to sum. No reason for this to be super live information, so we've set the aligner function to max, and the alignment Period to 60min.

The alert will trigger if Any time series is above some threshold for some period.



---

## Products Which Can Generate Big Logs

- Cloud Load Balancing
- The Logging agent on Compute Engine (or AWS)
- The write operation in the Cloud Logging API



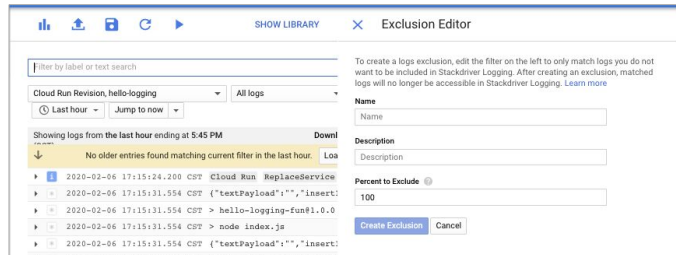
Some products which might generate substantial logs include: Cloud Load Balancing, the Compute Engine logging agents, and write operations in the Cloud Logging API.

Exclude what you don't need.

## Exclude What You Don't Need

Common exclusions:

- Load Balancer (90%)
- VPC Flow Logs
  - Percentage
  - Message type (CIDR)
- HTTP 200 OK from requests



If you Exclude log entries, you don't pay for them, but, they're also gone forever, so be careful what you leave in and what you leave out.

Also, remember you can exclude a percentage of entries, so another good trick is to do exclusions, but leave enough that you still get a taste.

Common exclusions:

- Google has several different Load Balancers, and they all support various forms of monitoring and/or logging. For load balancers that support logging, frequently, all you need is a small percentage. Exclude 90% or more.
- VPC Flow Logs is another good example. Again, you can often exclude 95%+ of the entries and still get enough to monitor the VPC. Also, remember that your exclusion can filter on entry contents. Perhaps you only want to retain logs from sources with a CIDR outside your network.
- Another common exclusion is web applications and services kicking out HTTP 200 OK from requests. Frequently, OK messages don't provide much insight, and they can generate a lot of entries.

## Logging Charges from Metrics

- Logging agents installed in a VM or on AWS
- Custom metrics in OpenCensus, OpenTelemetry, or custom code
  - OpenCensus and OpenTelemetry support sampling to reduce volume
- External metrics like Prometheus or NetApp
- Logs-based metrics



To reduce metric related charges, watch your agents. Virtual Machine monitoring and logging agents both can generate a lot of data. Though installing them is undoubtedly a best practice, there are absolutely going to be exceptions. Weigh the pros and cons, and remember that both the agents are customizable through configuration files.

Like the agents, custom metrics can generate spend. Newer metrics created using OpenCensus or OpenTelemetry support sampling to help reduce volume.

Metrics also might come from installed, third-party libraries like Prometheus or NetApp. They help provide visibility, but they also bump the bill.

And don't forget Logs based metrics we discussed previously.

---

## Trace Costs

- Pay by spans ingested and later scanned (count)
  - Except for App Engine Standard
- Surprise costs
  - Manual spans added to any code, even App Engine
  - Load Balancing spans and logs
- Spans created in high frequency loops



[Trace](#) charges are based on the number of trace spans ingested and scanned. When latency data is sent to Trace, it's packaged as a *trace* that is composed of *spans*, and the spans are *ingested* by the Cloud Trace backend. When you view trace data, the stored spans are *scanned* by Cloud Trace.

[Trace prices](#) are based on the number of [spans](#) ingested and eventually scanned. Some Google Cloud services, such as the App Engine standard environment, automatically produce non-chargeable spans, and trace has a free allotment per month, mentioned previously.

Don't be surprised by costs related to:

- Manual spans added to App Engine or custom application.
- Load balancing spans and logs.

Since spans are charged by count, watch code that generates lots of spans by looping, repeated operations.

## Logging Exports

- Exports themselves are free, but not target Resource
  - Cloud Storage
  - BigQuery storage, streaming, and queries
  - Pub/Sub message and network egress fees
- Review usage (resource, metric)
  - gcs\_bucket, total\_bytes
  - bigquery\_dataset, stored\_bytes
  - pubsub\_topic, byte\_cost
- Be specific with your filter
- **Trick: Export and Exclude**



Log exports themselves are free, but not the target Resources.

- Storage fees in Cloud Storage
- Storage, streaming, and query fees in BigQuery
- Pub/Sub message and networking egress fees

Review your usage. The following values are listed resource, metric

- gcs\_bucket, total\_bytes
- bigquery\_dataset, stored\_bytes
- pubsub\_topic, byte\_cost

Be specific with your export filters.

Also, remember you can Exclude and export simultaneously. That way, the data isn't gone, but it's stored outside of logging.

As a point of fact:

- 2 TiBs of data access log data stored in Logging would cost about \$1,000
- The same 2 TiBs stored in a regional, standard class bucket, would cost about \$40. With archival, it would be much cheaper.
-

## Be Selective with the Logging Agent

- Logs from third-party apps may be extensive
  - Apache, NGINX, MySQL
- May omit agent installation on selective machines
  - Standard logs still exist
  - Dev machines?
- Can install but configure on the machine
  - Disable integration with selected third-party apps



As mentioned earlier, be selective with the logging agent on your virtual machines.

Some third-party apps may generate a lot of logging and/or monitoring data (Apache, NGINX, MySQL).

You may want to omit agent installation on selective machines. Most of those apps are still generating local logs, which could be accessed if needed. Also, perhaps the logging is enabled on the dev machines, but not in production?

The logging and monitoring agents are also customizable, to exclude or include particular metrics or logs.

# Learned how to...

---

Analyze resource utilization cost for monitoring related components within Google Cloud

Implement best practices for controlling the cost of monitoring within Google Cloud



In this module you learned how to:

Analyze resource utilization costs for operations related components within Google Cloud

and

Implement best practices for controlling the cost of operations within Google Cloud

You did it! Great job.