

GCP Certification Series: 5.3 Viewing audit logs for the project and managed services



Prashanta Paudel

Nov 14, 2018 · 34 min read

Basic Concepts

Stackdriver Logging is part of the Stackdriver suite of products in Google Cloud Platform (GCP). It includes storage for logs, a user interface called the Logs Viewer, and an API to manage logs programmatically. Logging lets you read and write log entries, search and filter your logs, export your logs, and create logs-based metrics.

Projects

Logs are associated primarily with GCP projects, although other resources, such as organizations, folders, and billing accounts, can also have logs. The Logs Viewer shows only the logs from one project, but using the API, you can read log entries across multiple resources.

Log entries

A log entry records status or an event. The entry might be created by GCP services, AWS services, third-party applications, or your own applications. The “message” the log entry carries is called the payload, and it can be a simple string or structured data.

Your project receives log entries when you begin to use the services that routinely produce log entries, like Compute Engine or BigQuery. You also get log entries when you connect Stackdriver to AWS, when you install the Logging agent on your VM instances, and when you call the `entries.write` method in the API.

Logs

A log is a named collection of log entries within a GCP resource. Each log entry includes the name of its log. A log name can be a simple identifier, like `syslog`, or a structured name including the log's author, like `compute.googleapis.com/activity`. Logs exist only if they have log entries.

Retention period

Log entries are held in Stackdriver Logging for a limited time known as the retention period. After that, the entries are deleted. If you want to keep your log entries longer, export them outside of Stackdriver Logging.

The retention periods for different types of logs are listed in the Logging Quota Policy.

Monitored resources

Each log entry indicates where it came from by including the name of a monitored resource. Examples are individual Compute Engine VM instances, individual Amazon EC2 VM instances, database instances, and so on. For a complete listing of monitored resource types, see Monitored Resources and Services.

Filters

An advanced logs filter is an expression in the Logging filter language. It is used in the Logs Viewer and the Stackdriver Logging API to select log entries, such as those from a particular VM instance or those arriving in a particular time period with a particular severity level.

Exporting logs using sinks

Log entries received by Logging can be exported to Cloud Storage buckets, BigQuery datasets, and Cloud Pub/Sub topics. You export logs by configuring log sinks, which then continue to export log entries as they arrive in Logging. A sink includes a destination and a filter that selects the log entries to export.

Logs-based metrics

Metrics are a feature of Stackdriver Monitoring. A logs-based metric is a metric whose value is the number of log entries that match a filter that you specify.

Audit logs

An audit log is a permanent log written by a GCP service to record administrative or user actions. Audit logs appear in the Logs Viewer alongside other logs. For more information, see [Audit Logs](#).

Access control

The ability to read Logging logs is controlled by granting Cloud Identity and Access Management permissions to members.

Most logs can be read by any member with the Cloud IAM **Viewer** role. Data Access audit logs, except BigQuery Data Access audit logs, are the only “private logs”; to read these, the member requires either the Cloud IAM Owner role or other special permissions

- [illegible]

Access Control Guide

Stackdriver Logging uses Cloud Identity and Access Management to control access to logging data in projects, organizations, folders, and billing accounts.

Overview

Cloud IAM permissions and roles determine how you can use the Logging API and the Logs Viewer.

A Cloud IAM role is a collection of permissions. You assign these roles to members. You cannot assign a permission to a member directly; instead, you grant them a role, which gives them all the permissions that the role contains.

Log data resides in these resource types: projects, organizations, folders, and billing accounts. Each of these resources can have its own set of members with their own sets of Logging roles.

To create or use log data from Stackdriver Logging within a resource, a member must have a Cloud IAM role that includes the appropriate permissions. A summary list of those Cloud IAM roles and permissions is shown below:

- **roles/logging.viewer** (Logs Viewer) gives members read-only access to all features of Logging, except the permission to read private logs.
- **roles/logging.privateLogViewer** (Private Logs Viewer) gives members the permissions found in **roles/logging.viewer**, plus the permission to read private logs.
- **roles/logging.logWriter** (Logs Writer) can be granted to members that are service accounts and gives members just enough permissions to write logs. This role does not grant access to the Logs Viewer.
- **roles/logging.configWriter** (Logs Configuration Writer) gives members the permissions to create logs-based metrics and export sinks. To use the Logs Viewer, add the **roles/logging.viewer** role.
- **roles/logging.admin** (Logging Admin) gives members all permissions related to Logging. For a full list of these permissions, see API Permissions.
- **roles/viewer** (Project Viewer) gives members the same permissions as **roles/logging.viewer** at the project level. Note that granting this role applies the permissions to most GCP services at the project level, and is not confined to the usage of Logging.
- **roles/editor** (Project Editor) gives members the same permissions as **roles/logging.viewer**, plus permissions to write log entries, delete logs, and create logs-based metrics, at the project level. The role does not let you create export sinks or read private logs. Note that granting this role applies the permissions to most GCP services at the project level, and is not confined to the usage of Logging.
- **roles/owner** (Project Owner) gives you full access to Logging, including private logs. Note that granting this role applies the permissions to most GCP services at the project level, and is not confined to usage of Logging.

Data Access audit logs, except BigQuery Data Access audit logs, are the only “private logs”. To read them, members require certain permissions that are broader than read-only permissions.

For more details about Logging roles and permissions, see Permissions and roles on this page.

API Permissions

Logging API methods require specific Cloud IAM permissions. The following table lists the permissions needed by the API methods.

Note: If you are interested in logs held in organizations, billing accounts, and folders, note that those resources have their own API methods for `logs` and `sinks`. Rather than repeating all the methods in the table, only the `projects` methods are shown individually.

Logging method	Required permission	Resource
<code>type</code>	<code>billingAccounts.logs.*logging.logs.*</code> (See <code>projects.logs.*</code>)	<code>billing</code>
<code>accounts</code>	<code>billingAccounts.sinks.*logging.sinks.*</code> (See <code>projects.sinks.*</code>)	<code>billing</code>
<code>accounts</code>	<code>entries.listlogging.logEntries.list</code> or <code>logging.privateLogEntries.list</code>	<code>projects</code> , <code>organizations</code> , <code>folders</code> , <code>billing</code>
<code>accounts</code>	<code>entries.writelogging.logEntries.create</code>	<code>projects</code> , <code>organizations</code> , <code>folders</code> , <code>billing</code>
<code>accounts</code>	<code>folders.logs.*logging.logs.*</code> (See <code>projects.logs.*</code>)	<code>folders</code>
<code>accounts</code>	<code>folders.sinks.*logging.sinks.*</code> (See <code>projects.sinks.*</code>)	<code>folders</code>
<code>accounts</code>	<code>monitoredResourceDescriptors.list</code>	(none)
<code>(none)</code>	<code>organizations.logs.*logging.logs.*</code> (See <code>projects.logs.*</code>)	<code>organizations</code>
<code>(none)</code>	<code>organizations.sinks.*logging.sinks.*</code> (See <code>projects.sinks.*</code>)	<code>organizations</code>
<code>projects</code>	<code>projects.exclusions.createlogging.exclusions.create</code>	<code>projects</code>
<code>projects</code>	<code>projects.exclusions.deletelogging.exclusions.delete</code>	<code>projects</code>
<code>projects</code>	<code>projects.exclusions.getlogging.exclusions.get</code>	<code>projects</code>
<code>projects</code>	<code>projects.exclusions.listlogging.exclusions.list</code>	<code>projects</code>
<code>projects</code>	<code>projects.exclusions.patchlogging.exclusions.update</code>	<code>projects</code>
<code>projects</code>	<code>projects.logs.listlogging.logs.list</code>	<code>projects</code>
<code>projects</code>	<code>projects.logs.deletelogging.logs.delete</code>	<code>projects</code>
<code>projects</code>	<code>projects.sinks.listlogging.sinks.list</code>	<code>projects</code>
<code>projects</code>	<code>projects.sinks.getlogging.sinks.get</code>	<code>projects</code>

```

projects.sinks.createlogging.sinks.createprojects projects.sinks.u
pdatelogging.sinks.updateprojects projects.sinks.deletelogging.sink
s.deleteprojects projects.metrics.listlogging.logMetrics.listproject
s projects.metrics.getlogging.logMetrics.getprojects projects.metric
s.createlogging.logMetrics.createprojects projects.metrics.updatelo
gging.logMetrics.updateprojects projects.metrics.deletelogging.logM
etrics.deleteprojects

```

Permissions and roles

The following table lists the Cloud IAM roles that grant access to Stackdriver Logging. Each role has a specific set of logging permissions. Roles can be assigned to members of the listed resource types.

In the table, `a.b.{x,y}` means `a.b.x` and `a.b.y`.

Role name	Role title	Logging permissions	Resource type
<code>logging.viewer</code>	Logs Viewer	<code>logging.logEntries.list</code> <code>logging.logMetrics. { list , get }</code> <code>logging.logs.list</code> <code>logging.logServiceIndexes.list</code> <code>logging.logServices.list</code> <code>logging.sinks. { list , get }</code> <code>logging.usage.get</code>	<code>roles/</code> project, organization, folder, billing account
<code>logging.privateLogViewer</code>	Private Logs Viewer		<code>roles/logging.viewer</code> permissions, plus: <code>logging.privateLogEntries.list</code> project, organization, folder, billing account
<code>logging.logWriter</code>	Logs Writer	<code>logging.logEntries.create</code>	project, organization, folder, billing account
<code>logging.configWriter</code>	Logs Configuration Writer	<code>logging.exclusions. { list , create , get , update , delete }</code> <code>logging.logMetrics. { list , create , get , update , delete }</code> <code>logging.logs.list</code> <code>logging.logServiceIndexes.list</code> <code>logging.logServices.list</code> <code>logging.sinks. { list , create , get , update , delete }</code>	<code>roles/</code>

```

resourceManager.projects.get project, organization,
folder, billing account roles/
logging.admin Logging Admin logging.exclusions. { list , create ,
get , update , delete }
logging.logEntries.create
logging.logEntries.list
logging.logMetrics. { list , create , get , update , delete }
logging.logs.delete
logging.logs.list
logging.logServiceIndexes.list
logging.logServices.list
logging.privateLogEntries.list
logging.sinks.{list , create , get , update , delete }
resourceManager.projects.get project, organization,
folder, billing account roles/viewer Viewer logging.logEntries.list
logging.logMetrics. { list , get }
logging.logs.list
logging.logServiceIndexes.list
logging.logServices.list
logging.sinks. { list , get }
resourceManager.projects.get project roles/editor Editor roles/vie
wer Logging permissions, plus:
logging.logEntries.create
logging.logMetrics. { create , update , delete }
logging.logs.delete project roles/owner Owner roles/editor
Logging permissions, plus:
logging.privateLogEntries.list
logging.sinks. { create , update , delete }project

```

Custom roles

To create a custom role with Logging permissions, do the following:

- For a role granting permissions only for the Logging API, choose from the permissions in the preceding section, API permissions.
- For a role granting permissions to use the Logs Viewer, choose from permission groups in the following section, Console permissions.

For more information on custom roles, see Understanding Cloud IAM Custom Roles.

Console permissions

The following table lists the permissions needed to use the Logs Viewer.

In the table, `a.b.{x,y}` means `a.b.x` and `a.b.y`.

Console activity	Required permissions	Minimal read-only
access	<code>logging.logEntries.list</code> <code>logging.logs.list</code> <code>logging.logServiceIndexes.list</code> <code>logging.logServices.list</code> <code>resourceManager.projects.get</code>	Add ability to view logs-based
metricsAdd	<code>logging.logMetrics. { list , get }</code>	Add ability to view
exportsAdd	<code>logging.sinks. { list , get }</code>	Add ability to view logs
usageAdd	<code>logging.usage.get</code>	Add ability to exclude logsAdd
	<code>logging.exclusions. { list , create , get , update , delete }</code>	Add
ability to export logsAdd	<code>logging.sinks. { list , create , get , update , delete }</code>	Add ability to create logs-based metricsAdd
	<code>logging.logMetrics. { list , create , get , update , delete }</code>	

Access to exported logs

To create a sink, in order to export logs, you must have the permissions of `roles/logging.configWriter` or `roles/logging.admin` or `roles/owner`.

Once a sink begins exporting logs, it has full access to all incoming log entries. Sinks can export private log entries.

Once your log entries have been exported, access to the exported copies is controlled entirely by Cloud IAM permissions and roles on the destinations: Cloud Storage, BigQuery, or Cloud Pub/Sub.

Logging access scopes

Access scopes are the legacy method of specifying permissions for your Compute Engine VM instances. The following access scopes apply to the Logging API:

Access scope	Permissions
granted	<code>https://www.googleapis.com/auth/logging.read</code> <code>role/logging</code>
<code>.viewer</code>	<code>https://www.googleapis.com/auth/logging.write</code> <code>roles/loggi</code>

`ng.logWriter` <https://www.googleapis.com/auth/logging.admin> Full access to the Logging API. <https://www.googleapis.com/auth/cloud-platform> Full access to the Logging API and to all other enabled Google Cloud APIs.

Best practices

Now that Cloud IAM roles are available, a reasonable practice is to give all your VM instances the “Full access to all enabled Google Cloud APIs” scope:

```
https://www.googleapis.com/auth/cloud-platform
```

You can grant specific Cloud IAM roles in your VM instance’s service account to restrict access to specific APIs. For details, see [Service account permissions](#).

+++++

Using Exported Logs

This page explains how you can find and use your exported log entries in Cloud Storage, BigQuery, and Cloud Pub/Sub.

For an overview of exporting logs, see [Overview of Logs Export](#).

To learn how to export your logs, see the following pages:

- To use the Logs Viewer, see [Exporting Logs](#).
- To use the Stackdriver Logging API, see [Exporting Logs in the API](#).
- To use the command-line tool, see [gcloud logging](#).

Cloud Storage

To see your exported logs in Cloud Storage, do the following:

1. Go to the Cloud Storage browser in the GCP Console:

2. CLOUD STORAGE BROWSER
3. Select the bucket you are using for logs export.

See Cloud Storage organization for details on how logs are organized in the bucket.

Exported logs availability

If you don't see any exported logs, check the export system metrics. The export system metrics can tell you how many log entries are exported and how many are dropped due to errors. If the export system metrics indicate that no log entries were exported, check your export filter to verify that log entries matching your filter have recently arrived in Stackdriver Logging:

GO TO THE LOGS EXPORTS PAGE

Log entries are saved to Cloud Storage buckets in hourly batches. It might take from 2 to 3 hours before the first entries begin to appear.

Exported logs organization

When you export logs to a Cloud Storage bucket, Stackdriver Logging writes a set of files to the bucket. The files are organized in directory hierarchies by log type and date. The log type can be a simple name like `syslog` or a compound name like `appengine.googleapis.com/request_log`. If these logs were stored in a bucket named `my-gcs-bucket`, then the directories would be named as in the following example:

```
my-gcs-bucket/syslog/YYYY/MM/DD/  
my-gcs-  
bucket/appengine.googleapis.com/request_log/YYYY/MM/DD/
```

A single bucket can contain logs from multiple resource types.

Stackdriver Logging does not guarantee deduplication of log entries from sinks containing identical or overlapping filters; log entries from those sinks might be written multiple times to a Cloud Storage bucket.

The leaf directories (`DD/`) contain multiple files, each of which holds the exported log entries for a time period specified in the file name. The files are *sharded* and their names end in a shard number, `Sn` or `An` ($n=0, 1, 2, \dots$). For example, here are two files that might be stored within the directory `my-gcs-bucket/syslog/2015/01/13/` :

```
08:00:00_08:59:59_S0.json
08:00:00_08:59:59_S1.json
```

These two files together contain the `syslog` log entries for all instances during the hour beginning 0800 UTC. The log entry timestamps are expressed in UTC (Coordinated Universal Time).

To get all the log entries, you must read all the shards for each time period—in this case, file shards 0 and 1. The number of file shards written can change for every time period depending on the volume of log entries.

Within the individual sharded files, log entries are stored as a list of `LogEntry` objects. For an example `syslog` entry, see `LogEntry` type on this page.

Note that sort order of log entries within the files is not uniform or otherwise guaranteed.

BigQuery

To see your exported logs in BigQuery, do the following:

1. Go to the BigQuery UI in the GCP Console:
2. GO TO THE BIGQUERY UI
3. Select the dataset used as your sink's destination.
4. Select one of the dataset's tables. The log entries are visible on the **Details** tab, or you can query the table to return your data.

For more information, see [Table organization](#) to learn how the tables are organized, and [Exporting Logs and the BigQuery schema](#) to learn

how the exported log entry fields are named.

BigQuery availability

If you don't see any exported logs, check the export system metrics. The export system metrics can tell you how many log entries are exported and how many are dropped due to errors. If the export system metrics indicate that no log entries were exported, check your export filter to verify that log entries matching your filter have recently arrived in Stackdriver Logging:

GO TO THE LOGS EXPORTS PAGE

Log entries are saved to BigQuery in batches. It might take several minutes before the first entries begin to appear.

Table organization

When you export logs to a BigQuery dataset, Stackdriver Logging creates dated tables to hold the exported log entries. Log entries are placed in tables whose names are based on the entries' log names and timestamps¹. The following table shows examples of how log names and timestamps are mapped to table names:

Log name	Log entry timestamp	BigQuery table name
syslog	2017-05-23T18:19:22.135Z	syslog_20170523
apache-access	2017-01-01T00:00:00.000Z	apache-access_20170101
compute.googleapis.com/activity_log	2017-12-31T23:59:59.999Z	compute.googleapis.com_activity_log_20171231

1: The log entry timestamps are expressed in UTC (Coordinated Universal Time).

Schemas and fields

BigQuery table schemas for exported logs are based on the structure of the LogEntry type and the contents of the log payloads. You can see the table schema by selecting a table with exported log entries in the BigQuery Web UI.

The BigQuery table schema used to represent complex log entry payloads can be confusing and, in the case of exported audit logs, some

special naming rules are used. For more information, see [BigQuery Schema of Exported Logs](#).

Queries

Note: Audit logs exports to BigQuery now feature a compact format.

On March 1, 2019, the older schema will be removed. If you do not export audit logs to BigQuery, you will not be affected by this change. Users exporting audit logs to BigQuery should examine the changed fields and must update queries that consume them. For details, see [Migration to updated schema](#).

For examples of queries involving exported audit logs in BigQuery, see [BigQuery audit log queries](#).

See the [Query Reference](#) for more information on BigQuery queries. Especially useful are Table wildcard functions, which allow making queries across multiple tables and the Flatten operator, which allows to display data from repeated fields.

A sample Compute Engine logs query

The following BigQuery query retrieves log entries from multiple days and multiple log types:

- The query searches the last three days of the logs `syslog` and `apache-access`. The query was made on 23-Feb-2015 and it covers all log entries received on 21-Feb and 22-Feb, plus log entries received on 23-Feb up to the time the query was issued.
- The query retrieves results for a single Compute Engine instance, `15543007000000000000`.
- The query ignores traffic from the Stackdriver Monitoring endpoint health checker, `Stackdriver_terminus_bot`.

```
SELECT
  timestamp AS Time,
  logName as Log,
  textPayload AS Message
FROM
  (TABLE_DATE_RANGE(my_bq_dataset.syslog_,
    DATE_ADD(CURRENT_TIMESTAMP(), -2, 'DAY'),
    CURRENT_TIMESTAMP()),
```

```
(TABLE_DATE_RANGE(my_bq_dataset.apache_access_,
  DATE_ADD(CURRENT_TIMESTAMP(), -2, 'DAY'),
  CURRENT_TIMESTAMP()))
WHERE
  resource.type == 'gce_instance'
  AND resource.labels.instance_id == '1554300700000000000'
  AND NOT (textPayload CONTAINS 'Stackdriver_terminus_bot')
ORDER BY time;
```

Here are some example output rows:

```
Row | Time | Log
| Message
--- | --- | ---
-----
5 | 2015-02-21 03:40:14 UTC | projects/project-
id/logs/syslog | Feb 21 03:40:14 my-gce-instance
collectd[24281]: uc_update: Value too old: name =
15543007601548826368/df-tmpfs/df_complex-used; value time =
1424490014.269; last cache update = 1424490014.269;
6 | 2015-02-21 04:17:01 UTC | projects/project-
id/logs/syslog | Feb 21 04:17:01 my-gce-instance
/USR/SBIN/CRON[8082]: (root) CMD ( cd / && run-parts --
report /etc/cron.hourly)
7 | 2015-02-21 04:49:58 UTC | projects/project-
id/logs/apache-access | 128.61.240.66 - -
[21/Feb/2015:04:49:58 +0000] "GET / HTTP/1.0" 200 536 "-"
"masscan/1.0 (https://github.com/robertdavidgraham/masscan)"
8 | 2015-02-21 05:17:01 UTC | projects/project-
id/logs/syslog | Feb 21 05:17:01 my-gce-instance
/USR/SBIN/CRON[9104]: (root) CMD ( cd / && run-parts --
report /etc/cron.hourly)
9 | 2015-02-21 05:30:50 UTC | projects/project-
id/log/syslogapache-access | 92.254.50.61 - -
[21/Feb/2015:05:30:50 +0000] "GET /tmUnblock.cgi HTTP/1.1"
400 541 "-" "-"
```

A sample App Engine logs query

The following BigQuery query retrieves unsuccessful App Engine requests from the last month:

```
SELECT
  timestamp AS Time,
  protoPayload.host AS Host,
  protoPayload.status AS Status,
  protoPayload.resource AS Path
```

```
FROM

(TABLE_DATE_RANGE(my_bq_dataset.appspotengine_googleapis_com_request_log,
    DATE_ADD(CURRENT_TIMESTAMP(), -1, 'MONTH'),
    CURRENT_TIMESTAMP()))
WHERE
    protoPayload.status != 200
ORDER BY time
```

Here are some of the results:

```
Row | Time | Host
| Status | Path
--- | -
----- | -----
6 | 2015-02-12 19:35:02 UTC | default.my-gcp-project-id.appspot.com | 404 | /foo?thud=3
7 | 2015-02-12 19:35:21 UTC | default.my-gcp-project-id.appspot.com | 404 | /foo
8 | 2015-02-16 20:17:19 UTC | my-gcp-project-id.appspot.com | 404 | /favicon.ico
9 | 2015-02-16 20:17:34 UTC | my-gcp-project-id.appspot.com | 404 | /foo?thud=%22what???%22
```

Cloud Pub/Sub

To see your exported logs as they are streamed through Cloud Pub/Sub, do the following:

1. Go to the Cloud Pub/Sub page in the GCP Console:
2. GO TO CLOUD PUB/SUB
3. Find or create a subscription to the topic used for logs export, and pull a log entry from it. You might have to wait for a new log entry to be published.

See Exported logs organization for details on how logs are organized.

Exported logs availability

If you don't see any exported logs, check the export system metrics. The export system metrics can tell you how many log entries are exported

and how many are dropped due to errors. If the export system metrics indicate that no log entries were exported, check your export filter to verify that log entries matching your filter have recently arrived in Stackdriver Logging:

GO TO THE LOGS EXPORTS PAGE

When you export logs to a Cloud Pub/Sub topic, Stackdriver Logging publishes each log entry as a Cloud Pub/Sub message as soon as Stackdriver Logging receives that log entry.

Exported logs organization

The `data` field of each message is a base64-encoded `LogEntry` object. As an example, a Cloud Pub/Sub subscriber might pull the following object from a topic that is receiving log entries. The object shown contains a list with a single message, although Cloud Pub/Sub might return several messages if several log entries are available. The `data` value (about 600 characters) and the `ackId` value (about 200 characters) have been shortened to make the example easier to read:

```
{
  "receivedMessages": [
    {
      "ackId":
        "dRlJHlAbEGEIBERNK0EPKVgUWQYyODM...QlVWBwY9HFELH3cOAjYYF1cGI
        CIjIg",
      "message": {
        "data":
          "eyJtZXRhZGF0YSI6eyJzZXZ0eSI6Il...Dk0OTU2G9nIjoiaGVsbG93b3Js
          ZC5sb2cifQ==",
        "attributes": {
          "compute.googleapis.com/resource_type": "instance",
          "compute.googleapis.com/resource_id": "123456"
        },
        "messageId": "43913662360"
      }
    }
  ]
}
```

If you decode the `data` field and format it, you get the following `LogEntry` object:


```
{
  "log": "helloworld.log",
  "insertId": "2015-04-15|11:41:00.577447-07|10.52.166.198|-1694494956",
  "textPayload": "Wed Apr 15 20:40:51 CEST 2015 Hello, world!",
  "timestamp": "2015-04-15T18:40:56Z",
  "labels": {
    "compute.googleapis.com/resource_type": "instance",
    "compute.googleapis.com/resource_id": "123456"
  },
  "severity": "WARNING"
}
```

Log entry objects

Stackdriver Logging log entries are objects of type `LogEntry`. The most important fields of the log entry are shown in the following table:

It is customary for all the log entries with a particular [LOG_ID] to have the same format. Each log type documents the contents of its payload field. See the Stackdriver Logging logs index for examples. Following are some sample log entries:

Compute Engine's `syslog` is a custom log type produced by the logging agent, `google-fluentd`, which runs on virtual machine instances:

```
{
  logName: "projects/my-gcp-project-id/logs/syslog",
  timestamp: "2015-01-13T19:17:01Z",
  resource: {
    type: "gce_instance",
    labels: {
      instance_id: "12345",
      zone: "us-central1-a",
      project_id: "my-gcp-project-id"
    }
  },
  insertId: "abcde12345",
  textPayload: "Jan 13 19:17:01 my-gce-instance
/USR/SBIN/CRON[29980]: (root) CMD (  cd / && run-parts --
report /etc/cron.hourly)"
}
```

Late-arriving log entries

Exported log entries are saved to Cloud Storage buckets in hourly batches. It might take from 2 to 3 hours before the first entries begin to appear. Exported log file shards with the suffix `An` ("Append") hold log entries that arrived late.

Also, App Engine combines multiple sub-entries of type

`google.appengine.logging.v1.LogLine` (also called AppLog or

AppLogLine) under a primary log entry of

type `google.appengine.logging.v1.RequestLog` for the request that causes the log activity. The log lines each have a "request ID" that identifies the primary entry. The Logs Viewer displays the log lines with the request log entry. Stackdriver Logging attempts to put all the log lines into the batch with the original request, even if their timestamps would place them in the next batch. If that is not possible, the request log entry might be missing some log lines, and there might be "orphan" log lines without a request in the next batch. If this possibility is important to you, be prepared to reconnect the pieces of the request when you process your logs.

Third party integration with Cloud Pub/Sub

Stackdriver Logging supports logging integration with third parties. See Stackdriver Integrations for a current list of integrations.

You export your logs through a Cloud Pub/Sub topic and the third party receives your logs by subscribing to the same topic.

To perform the integration, expect to do something like the following:

1. Obtain from the third party a Google Cloud Platform (GCP) service account name created from their GCP project. For example, `12345-xyz@developer.gserviceaccount.com`. You use this name to give the third party permission to receive your logs.
2. In your project containing the logs,
3. ENABLE THE API

4. Create a Pub/Sub topic. You can do this when you configure a log sink, or by following these steps:
5. Go to the Pub/Sub topic list.
6. Select **Create topic** and enter a topic name. For example, `projects/my-project-id/topics/my-pubsub-topic`. You will export your logs to this topic.
7. Select **Create**.
8. Authorize Stackdriver Logging to export logs to the topic. See [Setting permissions for Cloud Pub/Sub](#).
9. Authorize the third party to subscribe to your topic:
10. Stay in the Pub/Sub topic list for your project in the GCP Console.
11. Select your new topic.
12. Select **Permissions**.
13. Enter the third party's service account name.
14. In the **Select a role** menu, select **Pub/Sub Subscriber**.
15. Select **Add**.
16. Give the third party the name of your Cloud Pub/Sub topic. For example, `projects/my-project-number/topics/my-pubsub-topic`. They should subscribe to the topic before you start exporting.

Start exporting the logs once your third party has subscribed to the topic:

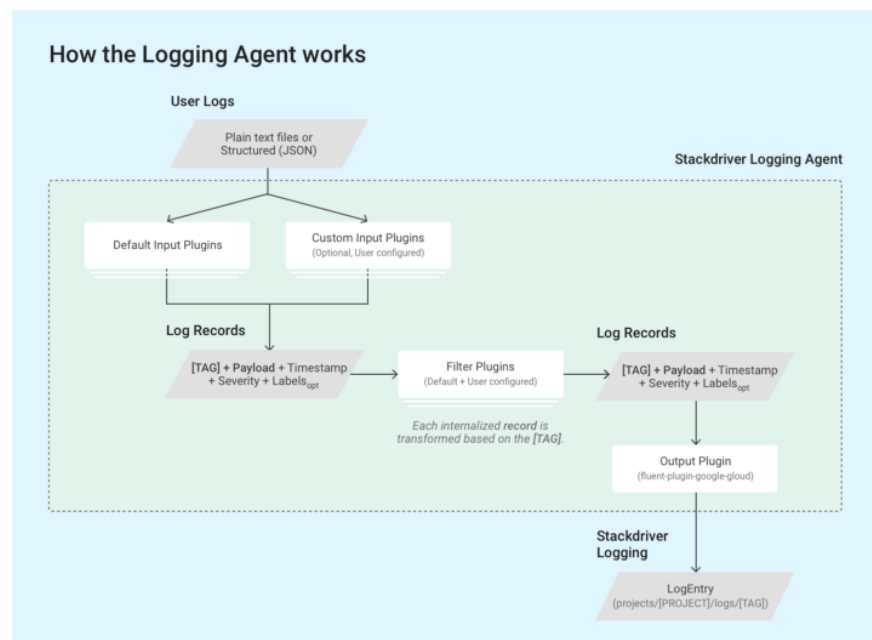
1. In your project containing the logs you want to export, click on **Create Export** above the search-filter box. This opens the **Edit Export** panel:
2. Enter a **Sink Name**.
3. In the **Sink Service** menu, select **Cloud Pub/Sub**.
4. In the **Sink Destination** menu, select the Cloud Pub/Sub topic to which the third party is subscribed.
5. Select **Create Sink** to begin the export.

6. A dialogue **Sink created** appears. This indicates that your export sink was successfully created with permissions to write future matching logs to the destination you selected.

Your third party should begin receiving the log entries right away.

About the Logging Agent

This guide provides basic information about the Stackdriver Logging agent, an application based on fluentd that runs on your virtual machine (VM) instances.



In its default configuration, the Stackdriver Logging agent streams logs from common third-party applications and system software to Stackdriver Logging; see the list of default logs. You can configure the agent to stream additional logs; see [Configuring the Stackdriver Logging agent](#) for details on agent configuration and operation.

It is a best practice to run the Stackdriver Logging agent on all your VM instances. The agent runs under both Linux and Windows. To install the Stackdriver Logging agent, see [Installing the Logging Agent](#).

Supported operating systems

You can run the Stackdriver Logging agent on the following operating systems on compatible virtual machine (VM) instances:

- CentOS 6 and 7
- Debian 7 “Wheezy”, Debian-7-backports, Debian 8 “Jessie”, and Debian 9 “Stretch”
- Red Hat Enterprise Linux 6 and 7
- Ubuntu LTS 14.04 “Trusty”, 15.04 “Vivid”, LTS 16.04 “Xenial”, 17.10 “Artful”, and LTS 18.04 “Bionic”
- SUSE Linux Enterprise Server 12 SP3, 12 SP2 for SAP, and 12 SP3 for SAP
- Windows Server 2008 R2, 2012 R2, and 2016
- Amazon Linux AMI (except Amazon Linux 2.0 AMI)
- Container-Optimized OS (supported for Kubernetes Engine nodes only)

Supported environments

The Stackdriver Logging agent is compatible with the following environments:

- Compute Engine instances. The Stackdriver Logging agent sends the logs to the project associated with each VM instance.
- For instances without external IP addresses, you must enable Private Google Access to allow the Stackdriver Logging agent to send logs.
- Amazon Web Services Elastic Compute Cloud (AWS EC2) instances. The Stackdriver Logging agent sends the logs to the AWS connector project for your Workspace. Stackdriver creates this project for you when you connect your AWS account to a Workspace.
- For the Stackdriver Logging agent to function correctly, the Amazon EC2 instance it runs on must be able to communicate

with Google Cloud APIs, particularly the Stackdriver Logging API. This requires either an external IP address or a VPC internet gateway.

For the VM instances above, a minimum of 250 MiB of resident (RSS) memory is required to run the Stackdriver Logging agent, but 1 GiB is recommended. For example, at a rate of 100 1-KB-sized log entries per second, the Stackdriver Logging agent with default configurations consumes 5% CPU on one core and 150 MiB memory. At a peak rate of 3,000 1-KB-sized log entries per second, the Stackdriver Logging agent, uses 80% CPU on one core and 250 MiB memory.

The following VM instances support Stackdriver Logging using their own software, possibly including custom versions or configurations of the Stackdriver Logging agent. Manually installing the Stackdriver Logging agent on them is not supported:

- App Engine standard environment VM instances. App Engine includes built-in support for Stackdriver Logging. For more information, see Stackdriver Logging in App Engine Apps.
- App Engine flexible environment VM instances. Apps running in the App Engine flexible environment can write logs that are in addition to what is included in the App Engine standard environment. For more information, see Stackdriver Logging and the App Engine flexible environment.
- Kubernetes Engine node instances. You can enable support for Stackdriver Logging on your new or existing container clusters. For more information, see Enabling Stackdriver Logging for Kubernetes Engine.

Stackdriver Logging agent source code

You do not need the information in this section unless you want to understand the source code or you have other special needs. The Stackdriver Logging agent is installed by the script described in the installation instructions.

The Stackdriver Logging agent, `google-fluentd`, is a modified version of the fluentd log data collector. `google-fluentd` is distributed in two

separate packages. The source code is available from the associated GitHub repositories:

- The GitHub repository named `google-fluentd` which includes the core `fluentd` program, the custom packaging scripts, and the output plugin for the Stackdriver Logging API.
 - The output plugin is packaged as a Ruby gem and is included in the `google-fluentd` package. It is also available separately at the Ruby gem hosting service at `fluent-plugin-google-cloud`.
 - The GitHub repository named `google-fluentd-catch-all-config` which includes the configuration files for the Stackdriver Logging agent for ingesting the logs from various third-party software packages.
-

Available Logs

This page provides information about the logs in Stackdriver Logging and about their structure.

Stackdriver Logging receives, indexes, and stores log entries from many sources, including Google Cloud Platform, Amazon Web Services, VM instances running the Stackdriver Logging fluentd agent, and user applications. All log entries in Stackdriver Logging are represented using a single data type, `LogEntry`, which defines certain common data for all log entries as well as carrying individual payloads. Stackdriver Logging can also export log entries to Google Cloud Storage, Google Cloud Pub/Sub, and Google BigQuery.

The LogEntry type

Every log entry in Stackdriver Logging is an object of type `LogEntry` that is characterized by the following information:

- The **project** or **organization** that owns the log entry.
- The **resource** to which the log entry applies. This consists of a **resource type** from the Monitored Resource List and additional values that denote a specific **instance**.

- A **log name**.
- A **timestamp**.
- A **payload**, which can be a **textPayload**, a **jsonPayload**, or (for GCP services) a **protoPayload**.

For more information about log entry contents, see the `LogEntry` type in the Stackdriver Logging API.

Audit logs

Google Cloud Audit Logging has three logs:

- **Admin Activity**, `cloudaudit.googleapis.com/activity`, called `activity` in the Logs Viewer
- **Data Access**, `cloudaudit.googleapis.com/data_access`, called `data_access` in the Logs Viewer
- **System Event**, `cloudaudit.googleapis.com/system_event`, called `system_event` in the Logs Viewer

Google Cloud Platform services write these logs to help you answer the question of “who did what, where, and when?” within your Google Cloud Platform projects.

Following are some characteristics of the audit log entries:

- Audit log entries have type `LogEntry`, as do all other Stackdriver Logging log entries.
- Each audit log entry includes the monitored resource to which it applies. You can find audit logs in the Logs Viewer’s resource selector menu under multiple names: **BigQuery**, **GCE instance**, etc.
- The payload of each audit log entry is an object of type `AuditLog`, a protocol buffer.
- The audit log entry’s payload has a field, `serviceData`, that some services use to hold additional information.

- All Admin Activity audit log entries are written to the log `cloudaudit.googleapis.com/activity` . Each log entry contains a monitored resource that identifies the resource whose activity is audited.

Audit logs cannot be deleted and are not subject to the same retention policy as other logs. For more information, see Audit Logging.

Agent logs

The Stackdriver Logging agent is a fluentd-based process that can run on supported VM instances. The agent sends system and third-party logs on the VM instance to Stackdriver Logging, where they appear as separate logs. For more information, see Default Logging Agent logs.

Logs available in Stackdriver Logging

The logs available in Stackdriver Logging can vary depending on which Google Cloud Platform resources you are using in your project. To learn more, visit the Google Cloud Platform Documentation Home and select the appropriate product or service.

Stackdriver Workspaces and Logging

You need a Workspace to send logs from an Amazon Web Service (AWS) account.

You cannot use a Workspace to view logs from multiple projects and AWS accounts at the same time. You must view or export the logs from each project and AWS account individually.

For an explanation of what Workspaces are and how they work, see Stackdriver Workspaces. For a step-by-step guide to creating and using Workspaces see Managing Workspaces.

Accessing AWS logs

To get logs from an AWS account, follow the instructions for Monitoring a single AWS account or Monitoring multiple projects. Each Amazon account is connected to GCP using an AWS connector project.

You will find the logs from your Amazon account in the connector project, not in the Workspace that contains the monitoring information.

You can find out the names of AWS connector projects in a Workspace by visiting the **Workspace settings** page for the Workspace:

GO TO THE STACKDRIVER WORKSPACE SETTINGS PAGE

Accessing GCP logs from a Workspace

If a Workspace is already monitoring multiple GCP projects, then you can find the *monitoring* information from those projects in the Workspace. However, you must look in the individual projects for their logs—the monitored projects or hosting project in the Workspace.

You can find out which GCP projects are being monitored by a Workspace by visiting the **Workspace settings** page for the Workspace:

GO TO THE STACKDRIVER WORKSPACE SETTINGS PAGE

Viewing Logs

This guide shows you how to search logs and view log entries with the Logs Viewer. To export your log entries, see [Exporting Logs](#). To read log entries through the Stackdriver Logging API, see [entries.list](#). To read log entries using the Google Cloud SDK, see [Reading log entries](#).

Getting started

1. Go to the **Stackdriver > Logging** page in the GCP Console:
2. GO TO THE LOGS VIEWER PAGE
3. Select an existing GCP project at the top of the page, or create a new project.
4. Using the drop-down menus, select the resource whose logs you want to view.

If you cannot see any logs, see the [Troubleshooting](#) section below.

Workspaces and Logging

You do not need a Workspace to use Stackdriver Logging, unless you want the ability to send logs from Amazon Web Services (AWS).

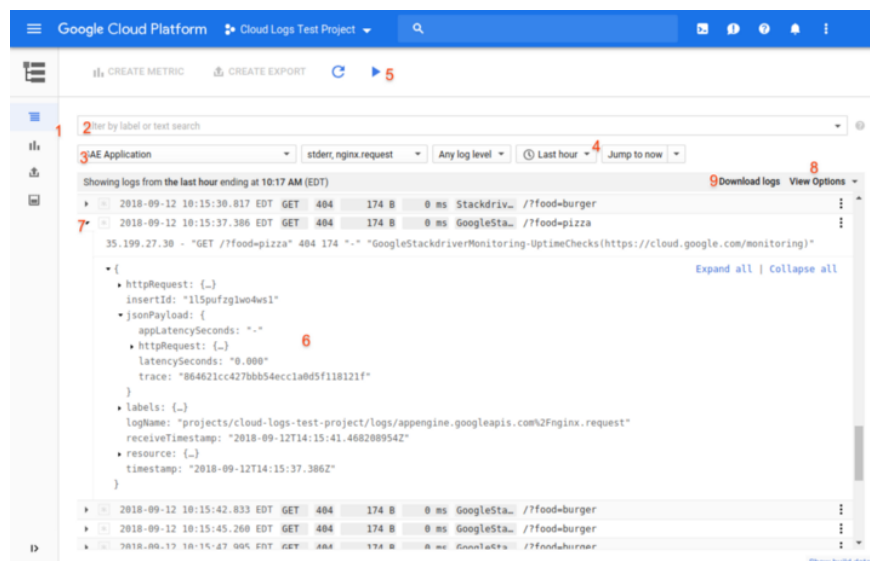
If you use Stackdriver Monitoring and a Workspace, note that Stackdriver Logging does not combine logs from monitored projects into the account project. You must select the project whose logs you want to see. For logs from an Amazon Web Services account, you must select the AWS connector project to see the AWS logs.

Logs Viewer filter interfaces

There are two filtering interfaces in the Logs Viewer:

- The default basic filter interface lets you select logs from menus and has a simple search capability.
- The advanced filter interface replaces the basic filter interface menus with a more powerful search capability that you can use to view log entries from multiple logs.

You can switch between the interfaces using the ▾ menu at the right side of the search-filter box in either interface. The following screenshot shows the Logs Viewer's layout. Four log entries from Compute Engine VM instances are shown. The second entry has been expanded by clicking its expander arrow (▶):



The basic filter interface has the following major components—indicated by red numbers in the screenshot above—some of which are shared with the advanced filter interface:

1. The *window tabs* let you choose **Logs**, **Metrics** (see Logs-based metrics), **Exports** (see Exporting logs), and **Logs ingestion**.
2. The *search-filter box* in the basic filter interface lets you filter log entries by label or text search. The basic filter is shown, and the menu at the end (▼) switches to the advanced filter interface.
3. The *basic selector menus* lets you choose resources, logs, and severity levels to display.
4. The *time-range selector* drop-down menus let you filter for specific dates and times in the logs.
5. The *streaming selector*, at the top of the page, controls whether new log entries are displayed as they arrive.
6. The *log-entry table* contains the log entries available according to your current filters and custom fields.
7. The *expander arrow* (▸) in front of each log entry lets you look at the full contents of the entry. For more information, see Expand log entries.
8. The **View Options** menu, at the far right, has additional display options.
9. The **Download logs** menu, at the far right, lets you download a set of log entries. For details, see Download log entries.

Scroll and stream logs

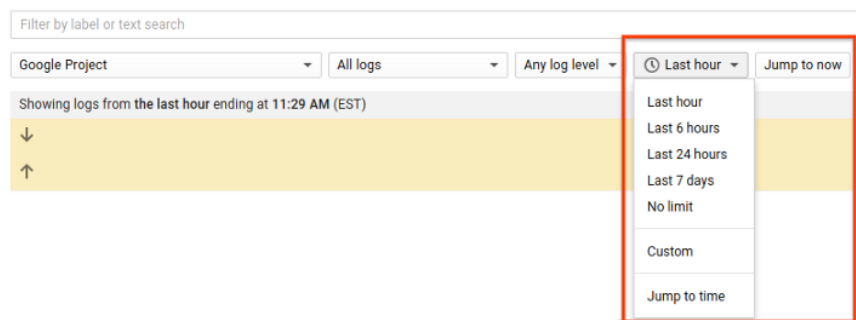
When you first look at the Logs Viewer, you see enough recent log entries to fill the screen. When you scroll through your log entries, the Logs Viewer tries to fetch additional entries. The yellow bar above and below the logs lets you know if more log entries might be available. Using the **View Options** menu, you can select the order in which to display log entries.

Icons at the top of the screen control when the logs are refreshed:

- The “refresh” icon (“Jump to newest logs”) will retrieve the latest logs and scroll the display to them.
- The “play” icon (▶) will start streaming the latest logs. It stops if you select a log entry or scroll the logs display.
- The “pause” icon (⏸) will stop streaming the latest logs.

Scroll to a time

You can filter your log entries by time and date using the *time-range selector* menus below the search filter box.



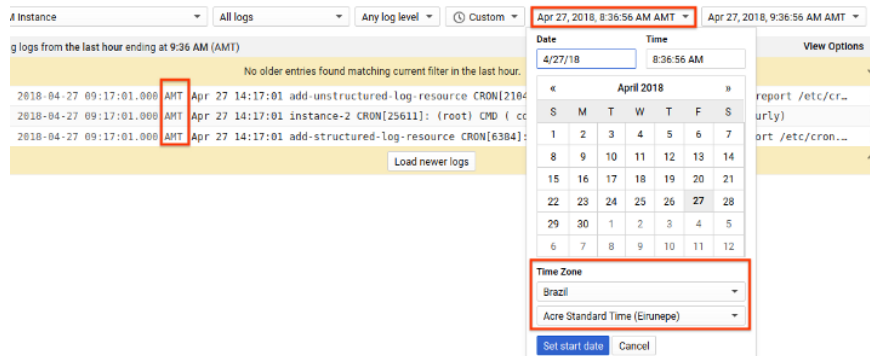
The default selection is **Last hour**. You can use the drop-down menu to select other time ranges or define a **Custom** range. Select **Jump to time** to filter the logs to a particular date and time, or use the **Jump to now** menu to see current log entries. Once you have made your selection, you can scroll the logs to inspect entries around that time. Clicking the **Refresh** or **Play** icons at the top of the page will reset the date and time in that menu to the most recently received log entry.

Change time zone

You can select a time zone by which to filter your logs:

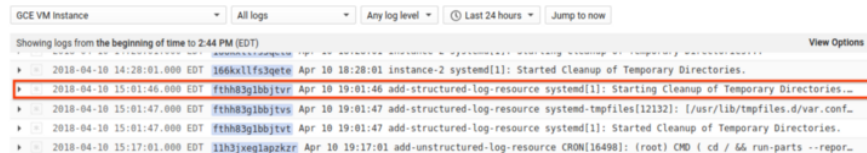
1. As shown in the preceding screenshot of the drop-down menu, under **Last hour**, select **Custom**. Two new drop-down menus appear to the right of **Custom**.
2. Click the expander arrow (▼) in either one of the new menus. A drop-down calendar menu appears.

3. In the **Time Zone** section of the calendar menu, select your preferred country and time zone.
4. Your Logs Viewer displays your updated time zone preference:



Expand log entries

The log-entry table displays a summary line for each log entry by default. Following is a screenshot with one summary line called out in red:



The Logs Viewer highlights certain fields of the log entry in the summary line. Certain fields are shown by default if they meet these criteria:

- The log-entry has a well-known type such as an App Engine request log.
- Otherwise, when the log entry contains the `httpRequest` field.
- Otherwise, when the log entry has a payload containing a field named `message`.

You can also add other fields to the summary line; see Add custom fields for details.

To see the full details for one log entry, click the expander arrow (▸) at the front of the summary line. To see the full details in a structured view for all the log entries available with your current filter, click the **View Options** menu at the far right and then select **Expand All**:



You can select **Collapse All** to collapse any expanded log entry details.

When you expand a summary line for a log entry, you will see a structured (JSON) view for that log entry:

```
{
  "textPayload": "Dec 6 20:28:53 your-gce-instance collectd[4778]: match_throttle_metadata_keys: 269 history entries, 233 distinct keys, 21483 bytes server memory.",
  "insertId": "1dearxwf7j44nL",
  "resource": {
    "type": "gce_instance",
    "labels": {
      "project_id": "my-gcp-project-id",
      "instance_id": "1428064241541024269",
      "zone": "us-central1-a"
    }
  },
  "timestamp": "2016-12-06T20:28:53Z",
  "labels": {
    "logName": "projects/my-gcp-project-id/logs/syslog"
  }
}
```

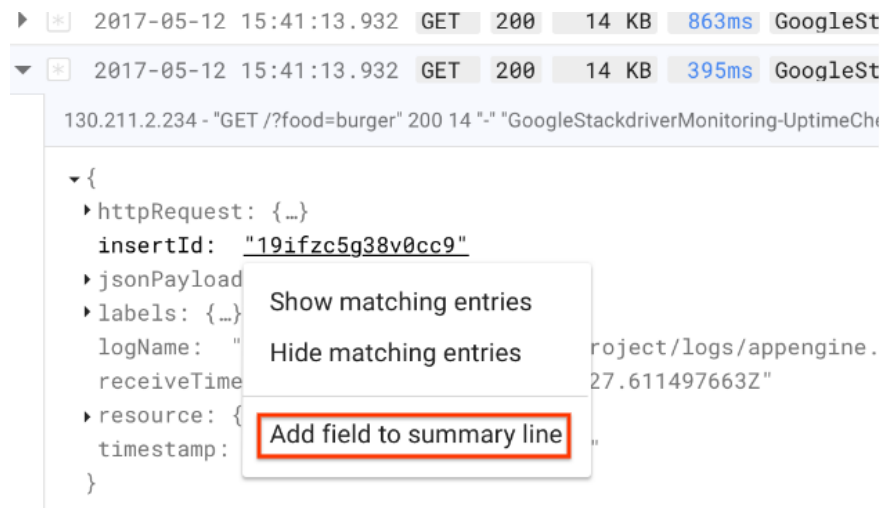
For a description of the fields in the entry, see the LogEntry type.

Add custom fields

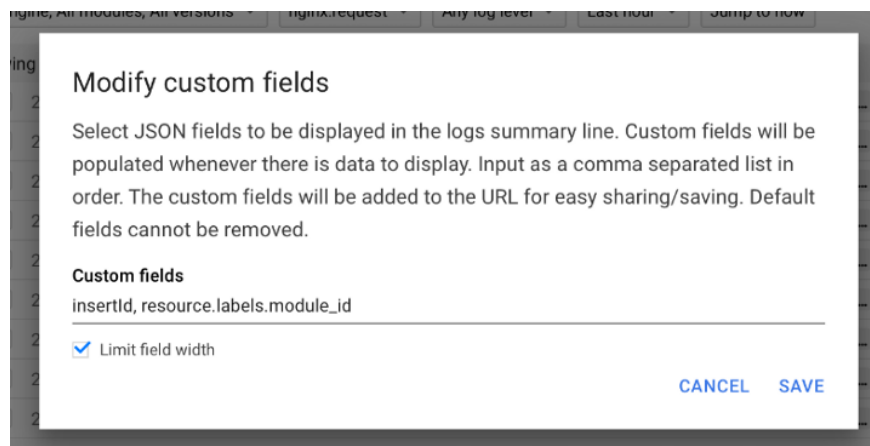
Custom fields are fields within log entries that you can bring into to a summary line for the main log entry summary.

There are 2 ways you can add custom fields to your log-entry table summary lines:

- In an expanded log entry, click on a field within the JSON representation. In the resulting panel, select **Add field to summary line**:



- From the **Viewing Options** menu at the top far right of the Logs Viewer, select **Add custom fields** (if you have existing custom fields in this project, this option will be **Modify custom fields**). In the resulting panel, add the desired JSON key and click **Save**. You can add multiple keys by separating them with a comma. To reorder the appearance of your custom fields in your summary lines, reorder the text in this panel and click **Save**.



Custom fields are populated whenever they are available in your log entries. Custom fields are added to your current URL and remain as long as you are using that URL or are within the same browser session. You cannot set them at a global level and they cannot be saved per user or per GCP project.

Custom fields are highlighted in blue in your log-entry table:

GCE VM Instance	All logs	Any log level	Last 24 hours	Jump to now
Showing logs from the beginning of time to 2:44 PM (EDT)				
2018-04-11 11:51:18.000 EDT	projects/my-sample-p...	xl8mhcglpj46p9	Apr 11 15:51:18	add-unstructured-log
2018-04-11 11:51:18.000 EDT	projects/my-sample-p...	xl8mhcglpj46pa	Apr 11 15:51:18	add-unstructured-log
2018-04-11 11:51:18.000 EDT	projects/my-sample-p...	xl8mhcglpj46pb	Apr 11 15:51:18	add-unstructured-log
2018-04-11 11:51:18.000 EDT	projects/my-sample-p...	xl8mhcglpj46pc	Apr 11 15:51:18	add-unstructured-log
2018-04-11 11:51:18.000 EDT	projects/my-sample-p...	xl8mhcglpj46pd	Apr 11 15:51:18	add-unstructured-log
2018-04-11 11:51:18.000 EDT	projects/my-sample-p...	xl8mhcglpj46pe	Apr 11 15:51:18	add-unstructured-log

There are 2 ways you can remove custom fields from your log-entry table summary lines:

- From any summary line that features the custom field that you wish to remove, click on the field and select **Remove field from summary line**.
- From the **Viewing Options** menu at the top far right of the Logs Viewer, select **Modify custom fields**. In the resulting panel, delete the JSON keys that you wish to remove and click **Save**.

Default fields cannot be removed from the log-entry table.

Show similar logs

You can click the value of an individual field in the expanded log entry view and then either show or hide all log entries with the same value:

GCE VM Instance
All logs

Showing logs from the last hour ending at 1:42 PM (EDT)

2018-04-11 13:17:01.000 EDT
Apr 11 17:17:01 in

{
insertId: "1l64ygfgl2zj9m1"
labels: {
logName: "t-12345/lo
receiveTim: :06.245400
resource:
textPayload: "Apr 11 17:17:01 instance-2 CRON
timestamp: "2018-04-11T17:17:01Z"
}

Show matching entries
Hide matching entries
Add field to summary line

2018-04-11 13:17:01.000 EDT
Apr 11 17:17:01 ad

2018-04-11 13:17:01.000 EDT
Apr 11 17:17:01 ad

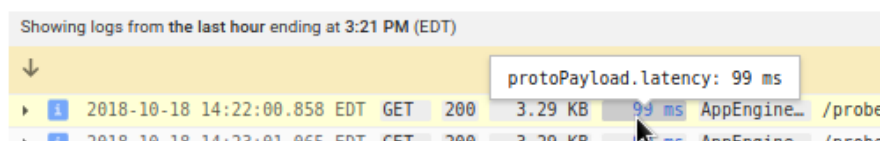
When you do this, the Logs Viewer changes to the advanced filter interface. To modify the search, edit the filter and click **Submit Filter**. For more information, see the Advanced filter interface.

Additionally, you can correlate App Engine request log entries and then view them in a nested structure. For details, see Viewing related request log entries and select your runtime language.

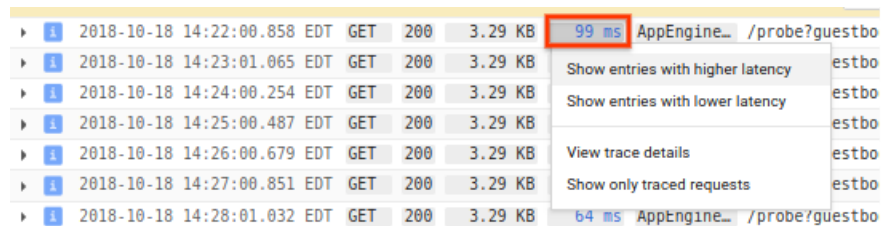
Show latency details

New! For **App Engine request logs**, the Logs Viewer provides a link to Stackdriver Trace for easy viewing of the log entry's latency details.

To show the menu of latency-related options for a log entry, identify the `protoPayload.latency` field:



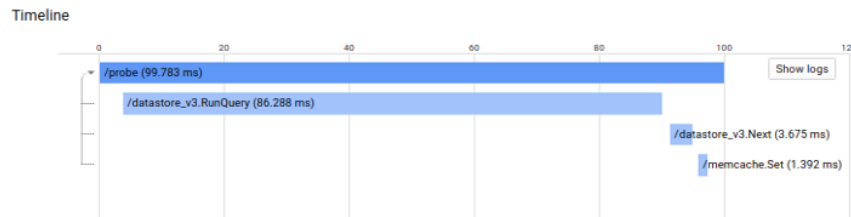
Click on the latency value:



The first two options in the menu restrict the log entries shown to those with higher or lower latencies. The last option in the menu restricts the log entries to those that contain trace details viewable by Stackdriver Trace. Specifically, the last option restricts the log entries to those where **View trace details** is enabled.

Viewing latency details in Stackdriver Trace

For certain App Engine request logs, the option **View trace details** is enabled. When enabled, click this option to open Stackdriver Trace and display the latency details of the log entry:



Selecting logs

Use the menus and filter box to find the logs you want to see:

- **Select a resource type and instance** whose logs you want to see. You can look at all instances of this resource type, or select a particular instance. In the screenshot above, **GCE VM instance**(all instances) is selected. For a list of resource types, see Monitored Resource List.
- **Select the named logs** you want to see from the second menu, or select **All logs**. The menu shows the logs that are in use by the selected resource instances.
- **Select the lowest severity level** you want to see in the third menu. Selecting **Any log level** will also show log entries that have no assigned severity.
- **Select the time range** you want to see from the fourth menu, or select **Jump to now** from the fifth menu.

As you change your menu selections, you will see the matching log entries.

Menu notes:

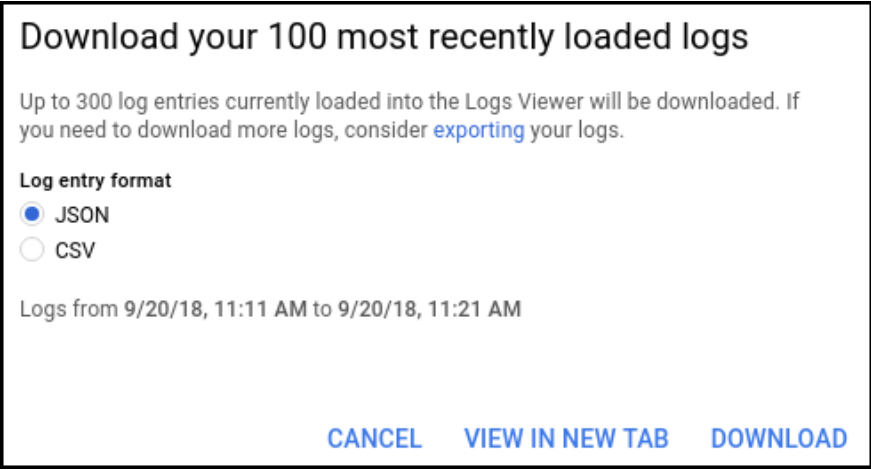
- Only the resource types, instances, and log names that are present in your project are shown in the menus. It may take a short time for the menus to be updated after adding a new resource type or instance, or writing to a new log.
- In the basic filter interface, you can look at only log entries from one resource type at a time. The advanced filter interface permits multiple resource types.

- You will not see any logs if you browse to a time before your current retention window. For more information, see [Logs retention periods](#).

Download log entries

With a few clicks, you can download, in JSON or CSV format, all of the log entries stored in the Logs Viewer's working memory. For performance reasons, the Logs Viewer attempts to load 100 log entries at a time and retains no more than 300 log entries in its working memory. These values are not configurable.

To download log entries, click the **Download logs** menu, which is located at the top far right of the Logs Viewer. In the download dialog, select JSON or CSV for the **Log entry format**, then click **Download**:



Download your 100 most recently loaded logs

Up to 300 log entries currently loaded into the Logs Viewer will be downloaded. If you need to download more logs, consider [exporting](#) your logs.

Log entry format

☒ JSON

☐ CSV

Logs from 9/20/18, 11:11 AM to 9/20/18, 11:21 AM

[CANCEL](#) [VIEW IN NEW TAB](#) [DOWNLOAD](#)

To view JSON- or CSV-formatted log entries in a web page, follow the same steps as for a download but select **View In New Tab**.

Searching with the Logs Viewer

You can further narrow your searches using filters in both the basic and advanced filter interfaces. The advanced filter interface contains most of the same features as the basic filter interface, but allows for more powerful search capabilities.

For more information on searching with the basic filter interface, see [Basic Logs Filters](#). For more information on advanced searches, see [Advanced Logs Filters](#).

Differences between basic and advanced filters

If you are familiar with the basic filter interface's text and field searches, here are some hints to get you going with the advanced logs filters.

Don't use "text:"

The Logs Viewer shows text searches in the basic filter by prefixing the text with the label `text:`. The `text:` label must not be used with advanced filters. The following table shows equivalent text searches:

Logs Viewer basic filter	Advanced logs filter with the same meaning
<code>text:"one two"</code>	<code>"one two"</code>
<code>text:three</code>	<code>three</code>
<code>text:n=5</code>	<code>n=5</code>

(quotations required)

If you accidentally use `text:` in the advanced filter, you will be searching for a match in a field called `text`, which doesn't exist.

Check field names

The basic filter interface has built-in field names for certain logs, including the App Engine request log. Those field names do not exist in advanced filters. For example, the following table shows an equivalent field search for an App Engine request log:

Basic filter	Advanced
filter <code>querystring:var=3</code>	<code>protoPayload.resource:"var=3" status:400..405</code>
	<code>protoPayload.status >= 400 AND protoPayload.status <= 405</code>

If the first sample, if you accidentally use the basic filter field name, you will be searching for a field named `querystring`, which doesn't exist, so the Logs Viewer will find no logs.

Substring matches

In the basic filter interface, all searches are case-insensitive substring matches. That is, the searches `text:abc` or `somefieldname:abc` will match log entries containing `abc`, `xyabcyx`, and `ABc`. In advanced

log filters, you must use the "has" search operator (`:`) for the same behavior.

For an exact match, use the equals operator (`=`). The comparison `field=abc` requires that `field` contain exactly `abc`, in any letter case. That search cannot be expressed in the basic filter interface.

AND and OR

In the basic filter interface, two comparisons using the same field name (or `text:`) are implicitly joined with `OR`, whereas comparisons with different labels are joined by `AND`. In advanced logs filters, all comparisons are joined by `AND` unless `OR` is specified explicitly. You can also use parentheses to group comparisons. The following table shows equivalent searches in the two filter interfaces:

Basic filter search	Advanced filter search
<code>text:xyzprotoPayload.resource:"def" AND ("abc" OR "xyz")</code>	<code>text:abc querystring:def</code>

Search performance

Here are some tips to increase your search performance:

- Search for specific values of indexed fields, like the log entry's name, resource type, and resource labels. In the basic filter interface, you do this with menu selections. In the advanced filter interface, use conditions like the following:


```
resource.type = "gce_instance"
logName =
"project/[PROJECT_ID]/logs/cloudaudit.googleapis.com%2Factivity"
resource.labels.module_id="default"
resource.labels.instance_id="1234567890"
```
- Choose exact matches over substring searches. Especially on index fields, partial matches are slower. In the basic filter interface, all text searches are partial matches. In the advanced filter interface, favor tests using equality operator (`=`) rather than using "has" (`:`).

- Shorten the time period searched. You cannot do this in the basic filter interface, but in the advanced filter interface you can specify a time range:
- ```
timestamp >= "2016-11-29T23:00:00Z" AND timestamp <= "2016-11-29T23:30:00Z"
```

For more information on performance, see [Finding log entries quickly](#).

## Troubleshooting

This section provides instructions for troubleshooting common issues found when interacting or searching with the Logs Viewer.

### There aren't any logs!

If you don't see any logs, then check the following:

- **Is the correct project selected at the top of the page?** If not, use the drop-down menu at the top of the page to select a project. You must select the project whose logs you want to see.
- **Does your project have any activity?** Even if the project is new, it should have activity or audit logs recording the fact that it was created. You can get more logs by going through the Quickstart.
- **Is the time range too narrow?** You can use the drop-down menus below the search filter box to select other time ranges or define a **Custom** range. Select **Jump to time** to filter the logs to a particular date and time, or use the **Jump to now** menu to see current log entries.

### My search isn't working!

If you are not sure why your search is not working in the basic filter interface, briefly switch to the advanced filter interface:

1. Select **Convert to advanced filter** in the ▾ menu at the end of the search box.
2. Look at the advanced filter to see if it is what you intended.

3. Return to the basic filter interface by using the browser's **Back** button.

Here are some other reasons you might not see all the log entries you expect:

- You cannot see log entries that are older than the Stackdriver Logging retention period. See Quota Policy for the logs retention period in effect.
  - During periods of heavy load there could be delays in sending logs to Stackdriver Logging or in receiving and displaying the logs.
  - The Logs Viewer does not show log entries that have timestamps in the future until the current time has “caught up” with them. This is an unusual situation, probably caused by a time skew in the application sending the logs.
- 
-



