This document discusses how to download and review access logs and storage information for your Cloud Storage buckets, and analyze the logs using Google BigQuery (/bigquery/).

Cloud Storage offers access logs and storage logs in the form of CSV files that you can download and view. *Access logs* provide information for all of the requests made on a specified bucket and are created hourly. *Storage logs* provide information about the storage consumption of that bucket for the last day and are created daily. Once set up, access logs and storage logs are automatically created as new objects in a bucket that you specify.

Timeliness and completeness of access logs delivery is not guaranteed.

In most cases, Cloud Audit Logs (/storage/docs/audit-logs) is the recommended method for generating logs that track API operations performed in Cloud Storage:

- Cloud Audit Logs tracks access on a continuous basis.
- Cloud Audit Logs produces logs that are easier to work with.
- Cloud Audit Logs can monitor many of your Google Cloud services, not just Cloud Storage.

In some cases, you may want to use access logs instead. You most likely want to use access logs if:

- You want to track access to public objects, such as assets in a bucket that you've configured to be a static website.
- You want to track access to objects when the access is exclusively granted because of the Access Control Lists (ACLs) (/storage/docs/access-control/lists) set on the objects.
- You want to track changes made by the Object Lifecycle Management (/storage/docs/lifecycle) feature.
- You intend to use authenticated browser downloads (/storage/docs/request-endpoints#cookieauth) to access objects in the bucket.

- You want your logs to include latency information, or the request and response size of individual HTTP requests.

Generally, you should not use storage logs. The recommended tool for measuring storage consumption is Stackdriver, which provides visualization tools as well as additional metrics related to storage consumption that storage logs do not. See Determining a bucket's size (/storage/docs/getting-bucket-information#bucket-size) for step-by-step instructions on using Stackdriver.

ɔn: If you use or enable VPC Service Controls (/vpc-service-controls/docs/overview), the logged bucket and the bucke
ɡ the log data must reside within the same security perimeter, or else logging fails.

The following steps describe how to set up logs delivery for a specific bucket using the gsutil tool (/storage/docs/gsutil), the XML API (/storage/docs/xml-api/overview), and the JSON API (/storage/docs/json_api/). If you don't have the gsutil tool, you can download and install it as part of the Google Cloud SDK (/sdk/docs/) package or as a stand-alone product (/storage/docs/gsutil_install#install).

Storage logs are generated once a day and contain the storage usage for the previous day. They are typically created before 10:00 am PST.

Usage logs are generated hourly when there is activity to report in the monitored bucket. Usage logs are typically created 15 minutes after the end of the hour.

Any log processing of usage logs should take into account the possibility that they may be delivered later than 15 mi after the end of an hour.

Usually, hourly usage log object(s) contain records for all usage that occurred during that hour. Occasionally, an hou usage log object contains records for an earlier hour, but never for a later hour.

Cloud Storage may write multiple log objects for the same hour.

Occasionally, a single record may appear twice in the usage logs. While we make our best effort to remove duplicate records, your log processing should be able to remove them if it is critical to your log analysis. You can use the **`s_request_id`** field to detect duplicates.

Access to your logs is controlled by the ACL on the log objects. Log objects have the default object acl (/storage/docs/access-control/lists#defaultobjects) of the log bucket.

The easiest way to download your access logs and storage logs is either through the Google Cloud Console (/storage/docs/cloud-console) or using the gsutil tool (/storage/docs/gsutil). Your access logs are in CSV format and have the following naming convention:

For example, the following is an access logs object for a bucket named gs://example-bucket, created on June 18, 2013 at 14:00 UTC and stored in the bucket gs://example-logs-bucket:

Storage logs are named using the following convention:

For example, the following is a storage log object for the same bucket on June 18, 2013:

To download logs:

To query your Cloud Storage usage and storage logs, you can use Google BigQuery (/bigquery/) which enables fast, SQL-like queries against append-only tables. The BigQuery Command-Line Tool, **bq**, is a Python-based tool that allows you to access BigQuery from the command line. For information about downloading and using bq, see the bq Command-Line Tool (/bigquery/bq-command-line-tool) reference page.

1. Select a default project.

For details about selecting a project, see Working With Projects (/bigquery/bq-command-line-tool#projects).

2. Create a new dataset.

3. List the datasets in the project:

4. Save the usage and storage schemas to your local computer for use in the load command.

   You can find the schemas to use at these locations: cloud_storage_usage_schema_v0 (http://storage.googleapis.com/pub/cloud_storage_usage_schema_v0.json) and cloud_storage_storage_schema_v0 (http://storage.googleapis.com/pub/cloud_storage_storage_schema_v0.json). The schemas are also described in the section Access and Storage Logs Format (#format).

5. Load the access logs into the dataset.

   These commands do the following:

   - Load usage and storage logs from the bucket `example-logs-bucket`.
   - Create tables `usage` and `storage` in the dataset `storageanalysis`.
   - Read schema data (.json file) from the same directory where the bq command runs.

- Skip the first row of each log file because it contains column descriptions.

Because this was the first time you ran the load command in the example here, the tables `usage` and `storage` were created. You could continue to append to these tables with subsequent load commands with different access log file names or using wildcards. For example, the following command appends data from all logs that start with "bucket_usuage_2014", to the `storage` table:

When using wildcards, you might want to move logs already uploaded to BigQuery to another directory (e.g., `gs://example-logs-bucket/processed`) to avoid uploading data from a log more than once.

BigQuery functionality can also be accessed through the BigQuery Browser Tool (/bigquery/bigquery-browser-tool). With the browser tool, you can load data through the create table process.

For additional information about loading data from Cloud Storage, including programmatically loading data, see Loading data from Cloud Storage (/bigquery/loading-data-into-bigquery#loaddatagcs).

In some scenarios, you may find it useful to pre-process access logs before loading into BigQuery. For example, you can add additional information to the access logs to make your query analysis easier in BigQuery. In this section, we'll show how you can add the file name of each storage access log to the log. This requires modifying the existing schema and each log file.

1. Modify the existing schema, cloud_storage_storage_schema_v0 (http://storage.googleapis.com/pub/cloud_storage_storage_schema_v0.json), to add file name as shown below. Give the new schema a new name, for example, **cloud_storage_storage_schema_custom.json**, to distinguish from the original.

2. Pre-process storage access log files based on the new schema, before loading them into BigQuery.

   For example, the following commands can be used in a Linux, macOS, or Windows (Cygwin) environment:

   The gsutil command copies the files into your working directory. The second command loops through the log files and adds "filename" to the description row (first row) and the actual file name to the data row (second row). Here's an example of a modified log file:

3. When you load the storage access logs into BigQuery, load your locally modified logs and use the customized schema.

Once your logs are loaded into BigQuery, you can query your access logs to return information about your logged bucket(s). The following example shows you how to use the bq tool in a scenario where you have access logs for a bucket over several days and you have loaded the logs as shown in Loading access logs into BigQuery (#loadBigQuery). You can also execute the queries below using the BigQuery Browser Tool (/bigquery/bigquery-browser-tool).

1. In the bq tool, enter the interactive mode.

2. Run a query against the storage log table.

For example, the following query shows how the storage of a logged bucket changes in time. It assumes that you modified the storage access logs as described in Modifying the Access Log Schema (#customSchema) and that the log files are named "log*storage**".

Example output from the query:

If you did not modify the schema and are using the default schema, you can run the following query:

3. Run a query against the usage log table.

   For example, the following query shows how to summarize the request methods that clients use to access resources in the logged bucket.

   Example output from the query:

4. Quit the interactive shell of the bq tool.

The access logs and storage logs can provide an overwhelming amount of information. You can use the following tables to help you identify all the information provided in these logs.

Access log fields:

Field	Type	Description
`time_micros`	integer	The time that the request was completed, in microseconds since the Unix epoch (https://en.wikipedia.org/wiki/Unix_epoch).
`c_ip`	string	The IP address from which the request was made. The "c" prefix indicates that this is information about the client.
`c_ip_type`	integer	The type of IP in the c_ip field: • A value of **1** indicates an IPV4 address. • A value of **2** indicates an IPV6 address.

`c_ip_region`	string	Reserved for future use.
`cs_method`	string	The HTTP method of this request. The "cs" prefix indicates that this information was sent from the client to the server.
`cs_uri`	string	The URI of the request.
`sc_status`	integer	The HTTP status code the server sent in response. The "sc" prefix indicates that this information was sent from the server to the client.
`cs_bytes`	integer	The number of bytes sent in the request.
`sc_bytes`	integer	The number of bytes sent in the response.
`time_taken_micros`	integer	The time it took to serve the request in microseconds, measured from when the first byte is received to when the response is sent. Note that for resumable uploads, the ending point is determined by the response to the final upload request that was part of the resumable upload.
`cs_host`	string	The host in the original request.
`cs_referer`	string	The HTTP referrer (https://en.wikipedia.org/wiki/HTTP_referrer) for the request.
`cs_user_agent`	string	The User-Agent (http://tools.ietf.org/html/rfc1945#section-10.15) of the request. The value is `GCS Lifecycle Management` for requests made by lifecycle management (/storage/docs/lifecycle).
`s_request_id`	string	The request identifier.
`cs_operation`	string	The Cloud Storage operation e.g. `GET_Object`.
`cs_bucket`	string	The bucket specified in the request. If this is a list buckets request, this can be null.
`cs_object`	string	The object specified in this request. This can be null.

Storage log fields:

Field	Type	Description
`bucket`	string	The name of the bucket.
`storage_byte_hours`	integer	Average size in byte-hours over a 24 hour period of the bucket. To get the total size of the bucket, divide byte-hours by 24.