Storage Products (https://cloud.google.com/products/storage/)

Documentation (https://cloud.google.com/storage/docs/) Guides

Consistency

This page explains which Cloud Storage operations are strongly consistent and which are eventually consistent. In the case of cacheable, publicly readable objects, you control the degree to which operations on the objects are consistent.

Strongly consistent operations

Cloud Storage provides strong global consistency for the following operations, including both data and metadata:

- Read-after-write

- Read-after-metadata-update

- Read-after-delete

- Bucket listing

- Object listing

- Granting access to resources using ACLs
  (https://cloud.google.com/storage/docs/access-control/lists)

When you upload an object to Cloud Storage, and you receive a success response, the object is immediately available for download and metadata operations from any location where Google offers service. This is true whether you create a new object or overwrite an existing object. Because uploads are strongly consistent, you will never receive a `404 Not Found` response or stale data for a read-after-write or read-after-metadata-update operation.

In addition, when an upload request succeeds, it means your data is replicated in multiple data centers. The latency for writing to Cloud Storage's globally consistent, replicated store might be slightly higher than for a non-replicated or non-committed store. This is because a success response is returned only when multiple writes complete, not just one.

Strong global consistency also extends to deletion operations on objects. If a deletion request succeeds, an immediate attempt to download the object or its metadata will result in a `404 Not`

`Found` status code. You get the `404` error because the object no longer exists after the delete operation succeeds.

Bucket listing is strongly consistent. For example, if you create a bucket, then immediately perform a `list buckets` operation, the new bucket appears in the returned list of buckets.

Object listing is also strongly consistent. For example, if you upload an object to a bucket and then immediately perform a `list objects` operation, the new object appears in the returned list of objects.

Important: Cached objects that are publicly readable might not exhibit strong consistency. See Cache control and consistency (#cache-control) for details.

For buckets, while metadata updates are strongly consistent for read-after-metadata-update operations, the resulting configuration changes may take time to propagate. For example, if you enable object versioning (https://cloud.google.com/storage/docs/object-versioning) on a bucket, you should wait at least 30 seconds before deleting or overwriting objects.

Eventually consistent operations

The following operations are eventually consistent:

- Revoking access from resources
- Granting access to resources using Cloud Identity and Access Management (https://cloud.google.com/storage/docs/access-control/iam)

It typically takes about a minute for these operations to take effect. In some cases, it might take several minutes longer.

As an example of behavior that can arise from eventual consistency, if you remove a user's access to a bucket, this change is immediately reflected in the metadata for the bucket; however, the user may still have access to the bucket for a short period of time.

Cache control and consistency

Cached objects that are publicly readable might not exhibit strong consistency. If you allow an object to be cached, and the object is in the cache when it is updated or deleted, the cached object is not updated or deleted until its cache lifetime expires.

The cache lifetime of an object is defined by the `Cache-Control` metadata associated with the object. The `Cache-Control` metadata can be set using a `Cache-Control` request header included in the initial upload of the object, or in a subsequent update to the metadata of the object (https://cloud.google.com/storage/docs/viewing-editing-metadata#edit). Because you control the `Cache-Control` metadata, you also control the degree to which cached objects are consistent. Moreover, while requests for the object can include their own `Cache-Control` header, these headers are ignored by Cloud Storage if they're set to avoid cached content.

Note: If you do not specify a cache lifetime, a publicly accessible object can be cached for up to 60 minutes.