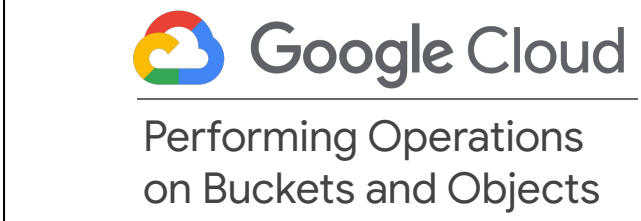


Course name: Developing Applications with Google Cloud
Module name: Performing Operations on Buckets and Objects

Featured products: Cloud Storage

Google Cloud Storage concepts

Resources are entities in Google Cloud including:

- Projects
- Buckets - the basic Cloud Storage container
- Objects - the individual pieces of data that you store in Cloud Storage

For more information, see: https://cloud.google.com/storage/docs/key-terms

Storage classes

Storage Class	Minimum duration	Availability SLA	Typical monthly availability	Use cases	Name for APIs and gcloud storage
Standard Storage	None	Multi-region 99.95% Dual-region 99.95% Region 99.9%	>99.99% availability in multi-regions and dual-regions; 99.99% in regions	Access data frequently ("hot" data) and/or store for brief periods • Serve website content • Stream videos • Interactive workloads • Mobile and gaming apps	STANDARD
Nearline Storage	30 days	Multi-region 99.9% Dual-region 99.9% Region 99.0%	99.95% availability in multi-regions and dual-regions; 99.9% in regions	Read/modify data ≤ once per month • Data backup • Serve long-tail multimedia content	NEARLINE
Coldline Storage	90 days			Read/modify data no more than once a quarter	COLDLINE
Archive Storage	365 days	None		Read/modify data < once a year • Cold data storage • Disaster recovery	ARCHIVE

Google Cloud

Google Cloud Storage has four primary storage classes, with different characteristics, use cases, and prices for your needs.

Standard Storage is best for data that is frequently accessed ("hot" data) and/or stored for only brief periods of time. When used in a region, co-locating your resources maximizes the performance for data-intensive computations and can reduce network charges. When used in a dual-region, you still get optimized performance when accessing Google Cloud products that are located in one of the associated regions, but you also get the improved availability that comes from storing data in geographically separate locations. When used in a multi-region, Standard Storage is appropriate for storing data that is accessed around the world, such as serving website content, streaming videos, executing interactive workloads, or serving data supporting mobile and gaming applications.

Nearline Storage is a low-cost, highly durable storage service for storing infrequently accessed data. Nearline Storage is a better choice than Standard Storage in scenarios where slightly lower availability, a 30-day minimum storage duration, and costs for data access are acceptable trade-offs for lowered at-rest storage costs. Nearline Storage is ideal for data you plan to read or modify on average once per month or less. Nearline Storage is appropriate for data backup, long-tail multimedia content, and data archiving.

Coldline Storage is a very-low-cost, highly durable storage service for storing infrequently accessed data. Coldline Storage is a better choice than Standard Storage

or Nearline Storage in scenarios where slightly lower availability, a 90-day minimum storage duration, and higher costs for data access are acceptable trade-offs for lowered at-rest storage costs. Coldline Storage is ideal for data you plan to read or modify at most once a quarter.

Archive Storage is the lowest-cost, highly durable storage service for data archiving, online backup, and disaster recovery. Archive Storage has higher costs for data access and operations, as well as a 365-day minimum storage duration. Archive Storage is the best choice for data that you plan to access less than once a year. For example, cold data storage, such as data stored for legal or regulatory reasons, and disaster recovery.

For more information, see: https://cloud.google.com/storage/docs/storage-classes

Characteristics applicable to all storage classes

- Unlimited storage with no minimum object size.
- Worldwide accessibility and worldwide storage locations.
- Can be used in any multi-region, dual-region, or region.
- Geo-redundancy if the data is stored in a multi-region or dual-region.
- Low latency (time to first byte typically tens of milliseconds).
- High durability (99.999999999% annual durability).
- A uniform experience with Cloud Storage features, security, tools, and APIs.

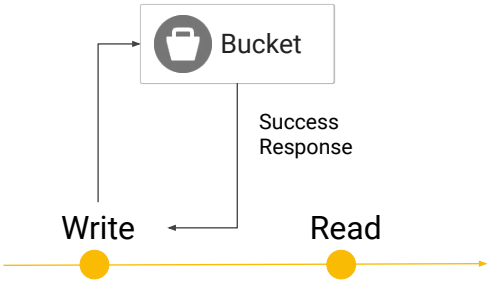
Google Cloud

We've discussed the four primary storage classes and differentiated between them in terms of characteristics, availability and use cases. It is worth noting that there are a number of characteristics that apply across all storage classes. These include:

- Unlimited storage with no minimum object size requirement,
- Worldwide accessibility and locations,
- Can be used in any multi-region, dual-region, or region,
- Geo-redundancy if data is stored in a multi-region or dual-region,
- Low latency and high durability, and
- A uniform experience, which extends to security, tools, and APIs.

The following operations are strongly consistent

- Read-after-write
- Read-after-metadata-update
- Read-after-delete
- Bucket listing
- Object listing
- Granting access to resources



Strongly consistent: when you *perform an operation* in Cloud Storage and *receive a success response*, the object is *immediately available* for download and metadata operations.

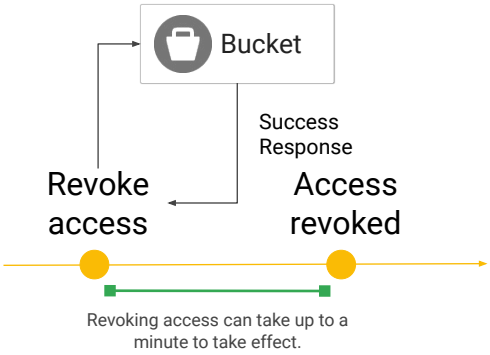
Google Cloud

Strongly consistent operations mean that when you perform an operation in 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. The following operations in Google Cloud Storage are strongly consistent: read-after-write, read-after-metadata-update, read-after-delete, bucket listing, object listing, and granting access to resources.

For more information, see: https://cloud.google.com/storage/docs/consistency

The following operations are eventually consistent

- Revoking access from objects
- Accessing publicly readable cached objects

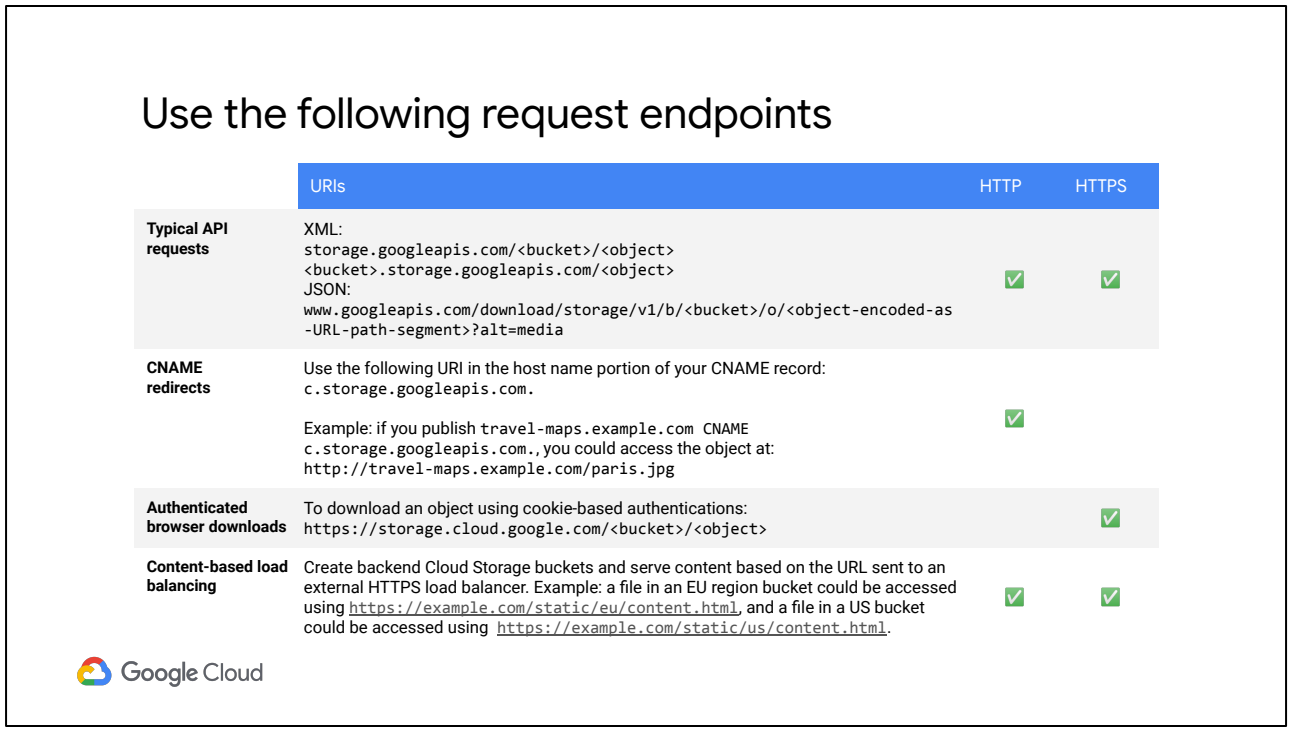


Eventually consistent: when you perform an operation, it may take some time for the operations to take effect.

Google Cloud

The following operations are eventually consistent: revoking access from objects and accessing publicly readable cached objects. Revoking access can take up to a minute for changes to take effect, so operations won't be consistent until the revocation has taken place. Publicly readable cached objects are eventually consistent if the object is in the cache when it is updated or deleted. When the object in the cache is updated or deleted, the operation doesn't take effect until its cache lifetime expires.

For more information, see:
https://cloud.google.com/storage/docs/consistency#eventually_consistent_operations



Depending on the operation you are performing and your application requirements, you can access Cloud Storage through three request endpoints (URIs).

Using the XML API, you can use the two listed URIs, and using the JSON API, you can use the listed URI. These URIs support secure sockets layer (SSL) encryption, so you can use either HTTP or HTTPS. If you authenticate to the Cloud Storage API using OAuth 2.0, you should use HTTPS.

A CNAME redirect is a special DNS record that lets you use a URI from your own domain to access a resource (bucket and object) in Cloud Storage without revealing the Cloud Storage URI. For example, if you published the following, *travel-maps.example.com CNAME c.storage.googleapis.com*, you could access an object with the following URI: http://travel-maps.example.com/paris.jpg. You can only use CNAME redirects with HTTP.

You can also access resources using authenticated browser downloads, which let you download data through your browser if you are signed in to your Google account and have been granted permission to read the data. Authenticated browser downloads use cookie-based Google account authentication in conjunction with Google account-based ACLs. To download an object using cookie-based authentication, you must use the following URI: https://storage.cloud.google.com/<bucket>/<object>. Only HTTPS is supported.

You can modify the content-based load balancing configuration to route requests for

static content to a Cloud Storage bucket. Requests to URI paths that begin with */static* can be sent to your storage bucket, and all other requests can be sent to your virtual machine instances. You can also use content-based load balancing to route to buckets in a specific region. For example, requests to a path starting with */static/eu* could route to a bucket in an EU region.

For more information, see:
Cookie-Based Authentications:
https://cloud.google.com/storage/docs/authentication.html#cookieauth
Request Endpoints: https://cloud.google.com/storage/docs/request-endpoints
Content-Based Load Balancing:
https://cloud.google.com/compute/docs/load-balancing/http/adding-a-backend-bucket-to-content-based-load-balancing

Composite objects and parallel uploads

Combine up to 32 objects into a single new object.

Use cases include:

- Dividing your data and uploading each chunk to a distinct object, composing your final object, and deleting any temporary objects.
- Uploading data to a temporary new object, composing it with the object you want to append it to, and deleting the temporary object.

Compose an object of smaller chunks using `gcloud storage`:

```
gcloud storage objects compose gs://example-bucket/component-obj-1
gs://example-bucket/component-obj-2
gs://example-bucket/composite-object
```

Google Cloud

You can compose up to 32 existing objects into a new object without transferring additional object data.

Object composition can be used for uploading an object in parallel: simply divide your data into multiple chunks, upload each chunk to a distinct object in parallel, compose your final object, and delete any temporary objects.

In parallel uploads, you divide an object into multiple pieces, upload the pieces to a temporary location simultaneously, and compose the original object from these temporary pieces.

Some use cases include:

- Fully use your available bandwidth by dividing your data into chunks and uploading them separately.
- Copying many files in parallel with a single command in big data scenarios.

For more information, see:
Composite objects: https://cloud.google.com/storage/docs/composite-objects
Working with Objects: https://cloud.google.com/storage/docs/object-basics

Design your application to handle network failures with truncated exponential backoff

Truncated exponential backoff:

- Is a standard error-handling strategy for network applications.
- Periodically retries failed requests with increasing delays between requests.
- Should be used for all requests to Cloud Storage that return HTTP 5xx and 429 response codes.

```
@retry(wait_exponential_multiplier=1000, wait_exponential_max=10000)

def wait_exponential_1000():

    print "Wait 2^x * 1000 milliseconds between each retry, up to 10 seconds, then 10 seconds afterwards"
```

Google Cloud

Truncated exponential backoff is a standard error-handling strategy for network applications in which a client periodically retries a failed request with increasing delays between requests.

Clients should use truncated exponential backoff for all requests to Cloud Storage that return HTTP 5xx and 429 response codes, including uploads and downloads of data or metadata.

Understanding how truncated exponential backoff works is important if you are building client applications that use the Cloud Storage XML API or JSON API directly, accessing Cloud Storage through a client library (some client libraries, such as the Cloud Storage Client Library for Node.js, have built-in exponential backoff), or using the gcloud storage command line tool (has configurable retry handling).

The Google Cloud console sends requests to Cloud Storage on your behalf and will handle any necessary backoff.

The example shows a backoff implementation for the Python retry library when you retry distributed services and other remote endpoints.

For more information, see: https://cloud.google.com/storage/docs/exponential-backoff

Quiz

Your application serves users in a specific region and performs analysis of data that changes frequently. What Cloud Storage class would be most appropriate?

A. Standard Storage

B. Nearline Storage

C. Coldline Storage

D. Archive Storage

Google Cloud

Your application serves users in a specific region and performs analysis of data. What Cloud Storage class would be most appropriate?

a. Standard Storage
b. Nearline Storage
c. Coldline Storage
D. Archive Storage

Quiz

Your application serves users in a specific region and performs analysis of data that changes frequently. What Cloud Storage class would be most appropriate?

A. Standard Storage

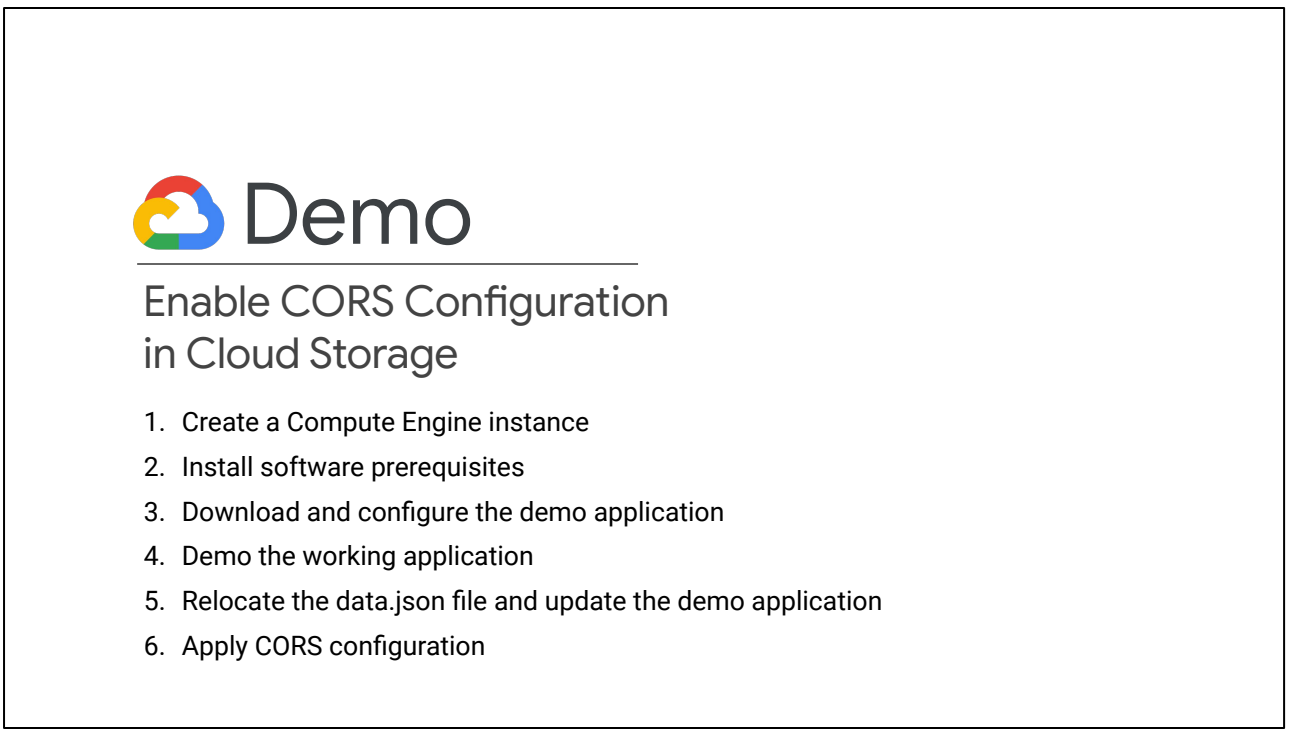
B. Nearline Storage

C. Coldline Storage

D. Archive Storage

Google Cloud

If your application serves users in a specific region and performs analysis of data, the Standard Storage class with a region location type would be most appropriate. (A)



Demo: Enabling CORS with Cloud Storage

You have a script on a page hosted from Google App Engine at example.appspot.com and want to use static resources stored in a Cloud Storage bucket at example.storage.googleapis.com. The browser won't allow a script from example.appspot.com to fetch resources from example.storage.googleapis.com using XMLHttpRequest because the resource being fetched is from a different origin.

The Cross Origin Resource Sharing (CORS) spec was developed by the World Wide Web Consortium (W3C) to get around this limitation. Cloud Storage supports this specification by allowing you to configure your buckets to return CORS-compliant responses. Because Cloud Storage supports CORS, a browser can ask example.storage.googleapis.com for permission to share its resources with scripts from example.appspot.com.

For more information, see: https://cloud.google.com/storage/docs/cross-origin


Google Cloud