

Best practices for content delivery

This page provides best practices for optimizing and accelerating content delivery with [Cloud CDN \(/cdn\)](#).

Cloud CDN uses [HTTP\(S\) Load Balancing \(/load-balancing/docs/https\)](#) as the origin for cacheable content. An external HTTP(S) load balancer can deliver a mix of static and dynamically-created content to users through one global IP address from [Compute Engine virtual machine \(VM\) instance groups \(/compute/docs/instance-groups\)](#), [Cloud Storage buckets \(/storage/docs/creating-buckets\)](#), or both.

Because of the seamless integration with Google Cloud, you have several options for deploying Cloud CDN and managing content. Use the best practices listed here to plan and refine your deployment. For more information, see [Using Cloud CDN \(/cdn/docs/using-cdn\)](#).

Setting the expiration for time-sensitive content

Each piece of content in a Cloud CDN cache has an associated expiration time, and it's important to set an expiration that is appropriate for your use case. Because origin servers must resend content that expires on cache servers, you need to choose the expiration carefully.

One method for choosing the expiration is to categorize content based on how often you update the content; for example:

- **Near real-time updates** such as live feeds for sporting events or traffic
- **Frequent updates** such as weekly, daily, or hourly weather information or front-page news images
- **Infrequent updates** such as a website logo or CSS or JavaScript files

Next, choose the expiration by content category. For example, a five-second expiration might be appropriate for near real-time sports scores, and a one-hour expiration could be used for weather updates. For content stored in Cloud Storage, set the expiration times by using [Cache-Control metadata \(/storage/docs/goutil/addlhelp/WorkingWithObjectMetadata#cache-control\)](#). When content is served by Compute Engine, you control expiration times by configuring your web server software.

Expiration times are specified by the `max-age` and `s-maxage` values in the `Cache-Control` header. This header is defined by the [HTTP specification](https://tools.ietf.org/html/rfc7234#section-5.2) (<https://tools.ietf.org/html/rfc7234#section-5.2>). For example, the following `Cache-Control` header makes the associated content publicly readable and cacheable with a cache expiration of 72 hours (259200 seconds):

```
Cache-Control: public, max-age=259200
```

To maximize caching, follow the guidelines in the [Caching overview](/cdn/docs/caching) (</cdn/docs/caching>). Remember that `max-age` and `s-maxage` values in the `Cache-Control` metadata field work together in the following ways:

- The `max-age` and `s-maxage` values are measured in seconds.
- The `s-maxage` value applies only to shared caches, not browser caches.
- The `max-age` value applies to all caches unless `s-maxage` overrides it.

For content that changes infrequently or that must change along with related content, it's often appropriate to use a long expiration time in combination with [versioned URLs](#) (`#versioned-urls`).

Using custom cache keys to improve cache hit ratio

By default, Cloud CDN uses the complete request URL to build the cache key. For performance and scalability, it's important to optimize cache hit ratio. To help optimize your cache hit ratio, you can use [custom cache keys](/cdn/docs/caching#cache-keys) (</cdn/docs/caching#cache-keys>).



(</cdn/images/cache-key.svg>)
Cloud CDN key-value store

You can customize cache keys to include or omit any combination of protocol, host, and query string. For example, suppose you have two websites on different domains that use the same logo. To show the logo, use custom cache keys, as follows:

- The website content is different, but you use the same company logo on both domains. When you turn on Cloud CDN and customize the cache keys (/cdn/docs/using-cache-keys#turn-on) for the backend service that holds the logo, clear the **Host** checkbox so that the cache ignores the domain but caches the logo.
- The logo needs to be cached whether displayed through HTTP or HTTPS. When you customize the cache keys for the backend service that holds the logo, clear the **Protocol** checkbox so that requests through HTTP and HTTPS count as matches for the logo's cache entry.

To learn how to customize cache keys, see Using cache keys (/cdn/docs/using-cache-keys).

Using versioned URLs to update content

Versioning content serves a different version of the same content, effectively removing it by showing users new content before the cache entry expires. Because versioning is free and easy to use, we recommend that you use versioning as the default approach for updating cacheable content.

To version content, add a parameter to the URL, such as a version number. There are various ways to include parameters in URLs, such as:

- Add a query string: `file.ext?v=100`
- Alter the file name: `file.1.0.0.ext` or `file_v100.ext`
- Alter the file path: `/v100/file.ext`

When you add the parameter, you change the name of the file and the URL. This change forces the cache to ignore any existing cache entry.

Using invalidation sparingly to remove content

Invalidation (/cdn/docs/cache-invalidation-overview) removes content from the Cloud CDN distributed cache servers before the cache entry expires. Invalidation is eventually consistent.

We recommend that you use invalidation sparingly and only as a last resort. For example, invalidation is useful when you must remove content for legal reasons or because of an

accidental upload. Otherwise, we recommend that you use versioning whenever possible or wait until the content expires normally. Cloud CDN cache servers routinely evict infrequently accessed content to make room for new content. Content that is not accessed for 30 days is removed unconditionally.

In contrast to [setting expiration times](#) (#expiration) and [using versioned URLs](#) (#versioned-urls), you are [charged for each cache invalidation that you initiate](#) (/cdn/pricing#overview), and cache invalidations are [rate limited](#) (/cdn/docs/troubleshooting-steps#known-issues).

To learn more about invalidation, see the [Cache invalidation overview](#) (/cdn/docs/cache-invalidation-overview).

Using signed URLs

If you're using [signed URLs](#) (/cdn/docs/private-content), note the following:

- Keep public and private content in [separate Cloud Storage buckets](#) (/storage/docs/access-control#recommended_bucket_architecture).
- Follow [security best practices](#) (/cdn/docs/using-signed-urls#security_considerations).

Reviewing third-party performance tests

Review reports from third-party providers, such as the availability, latency, and throughput [reports](#) (https://itm.cloud.com/google-reports/) provided by [Citrix Radar](#) (https://docs.citrix.com/en-us/citrix-intelligent-traffic-management/radar.html).

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](#) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](#) (https://www.apache.org/licenses/LICENSE-2.0). For details, see the [Google Developers Site Policies](#) (https://developers.google.com/site-policies). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2020-07-13 UTC.