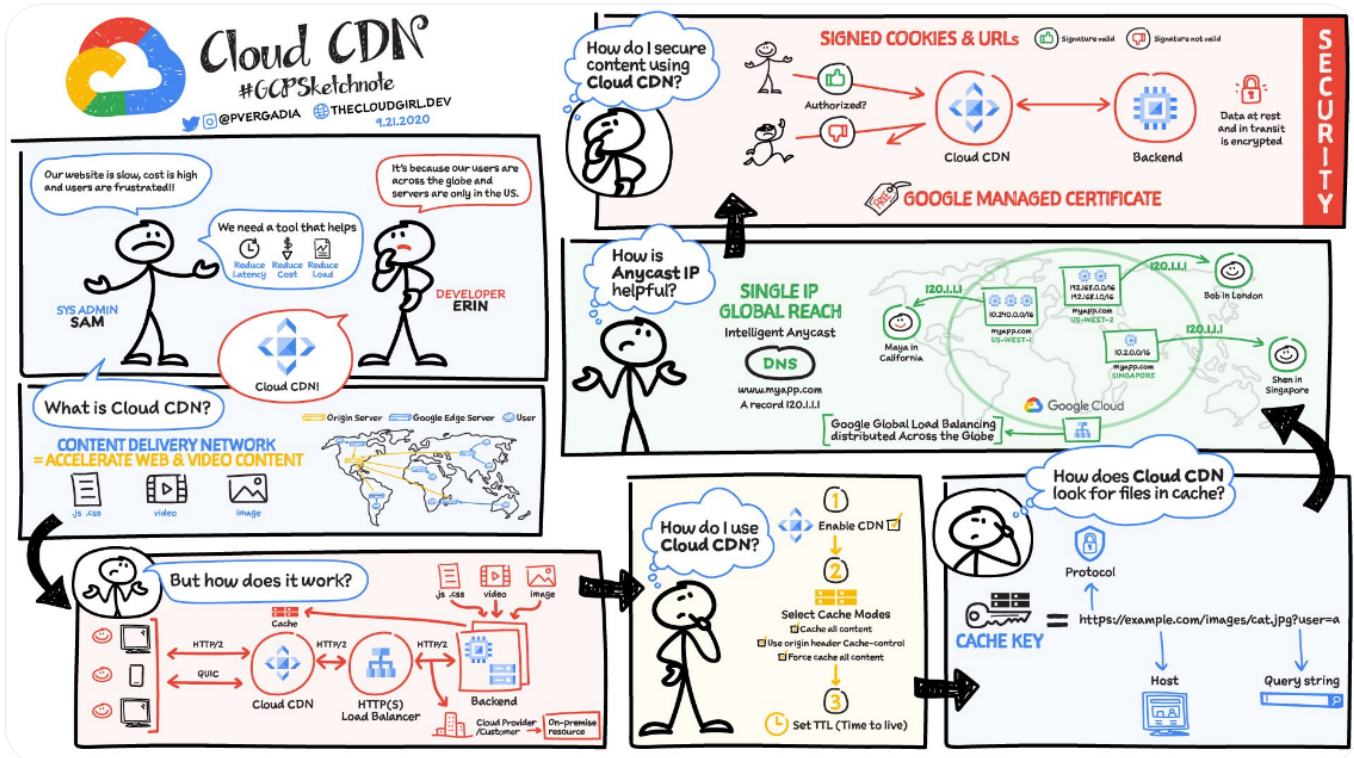




Developers & Practitioners

What is Cloud CDN and how does it work?

November 17, 2021

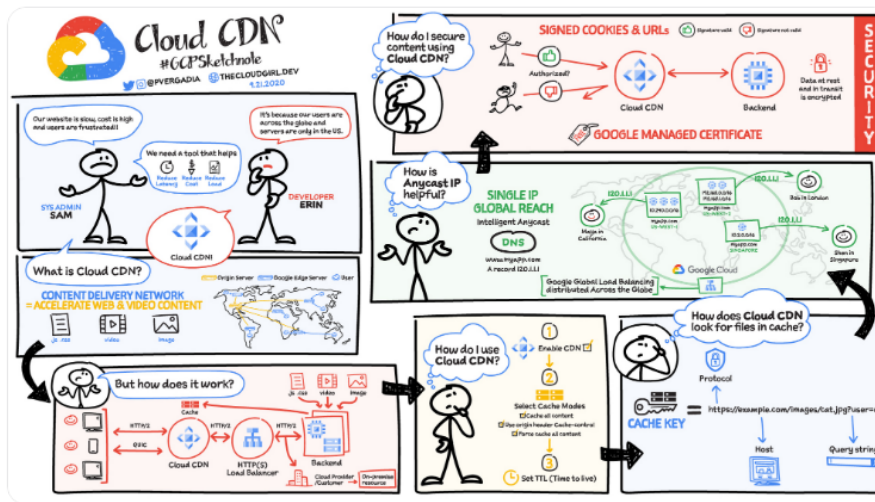


Priyanka Vergadia

Staff Developer Advocate, Google Cloud

No matter what your app or website does, chances are that your users are distributed across various locations and are not necessarily close to your servers. This means the requests travel long distances across the public internet, leading to inconsistent and sometimes frustrating user experiences. That's where [Cloud CDN](#) comes in!





Click to enlarge

What is Cloud CDN?

Cloud CDN is a content delivery network that accelerates your web and video content delivery by using Google's global edge network to bring content as close to your users as possible. As a result latency, cost, and load on your backend servers is reduced, making it easier to scale to millions of users. Global anycast IP provides a single IP for global reach. It enables Google Cloud to route users to the nearest edge cache automatically and avoid DNS propagation delays that can impact availability. It supports HTTP/2 end-to-end and the QUIC protocol from client to cache. QUIC is a multiplexed stream transport over UDP, which reduces latency and makes it ideal for lossy mobile networks.

How does Cloud CDN work?

Let's consider an example to understand how Cloud CDN works:

- When a user makes a request to your website or app, the request is routed to the closest [Google edge node](#) (we have over 120 of these!) for fast and reliable traffic flow. From there the request gets routed to the global HTTPS Load Balancer to the backend or origin.
- With Cloud CDN enabled, the content gets directly served from the cache — a group of servers that store and manage cacheable content so that future requests for that content can be served faster.
- The cached content is a copy of cacheable web assets (JavaScript, CSS), images, video, and other content that is stored on your origin servers.
- Cloud CDN automatically caches this content when you use the recommended “cache

The cached content is a copy of cacheable web assets (JavaScript, CSS), images, video, and other content that is stored on your origin servers.

-

Cloud CDN automatically caches this content when you use the recommended “cache mode” to cache all static content. If you need more control, you can direct Cloud CDN by setting HTTP headers on your responses. You can also force all content to be cached; just know that this ignores the “private”, “no-store”, or “no-cache” directives in Cache-Control response headers.

-

When the request is received by Cloud CDN it looks for the cached content using a [cache key](#). This is typically the URI, but you can customize the cache key to remove protocol, hosts, or query strings.

-

If a cached response is found in the Cloud CDN cache, the response is retrieved from the cache and sent to the user. This is called a cache hit. When a cache hit occurs, Cloud CDN looks up the content by its cache key and responds directly to the user, shortening the round-trip time and reducing the load on the origin server.

-

The first time that a piece of content is requested, Cloud CDN can't fulfill the request from the cache because it does not have it in cache. This is called a cache miss. When a cache miss occurs, Cloud CDN might attempt to get the content from a nearby cache. If the nearby cache has the content, it sends it to the first cache by using cache-to-cache fill. Otherwise, it just sends the request to the origin server.

-

The maximum lifetime of the object in a cache is defined by the TTLs, or time to live values, set by the cache directives for each HTTP response or cache mode. When the TTL expires, the content is evicted from cache.

How to use Cloud CDN

You can set up Cloud CDN through gCloud CLI, Cloud Console, or the APIs. Since Cloud CDN uses [Cloud Load Balancing](#) to provide routing, health checking, and anycast IP support, it can be enabled by easily selecting a checkbox while setting up your backends or origins.

Cloud CDN makes it easy to serve web and media content using Google Cloud Storage. You just upload your content to a Cloud Storage bucket, set up your load balancer, and enable caching. To enable hybrid architectures spanning across clouds and on-premises, Cloud CDN and HTTP(S) Load Balancing also support external backends.

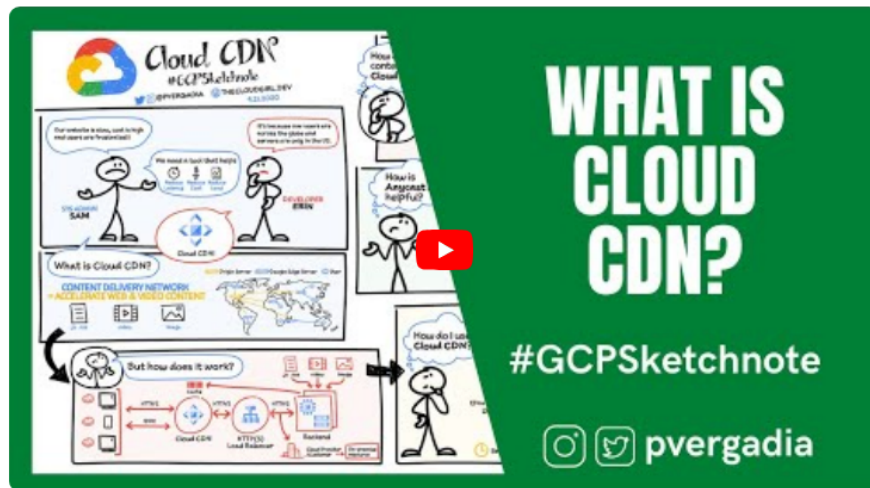
Security

-



Security

- Data is encrypted at rest and in transit from Cloud Load Balancing to the backend for end-to-end encryption.
 - You can programmatically sign URLs and cookies to limit video segment access to authorized users only. The signature is validated at the CDN edge and unauthorized requests are blocked right there!
 - On a broader level, you can enable SSL for free using Google managed certs!
- For a more in-depth look into Cloud CDN check out the [documentation](#).



For more #GCPSketchnote, follow the [GitHub repo](#). For similar cloud content follow me on Twitter [@pvergadia](#) and keep an eye out on [thecloudgirl.dev](#).

Developers & Practitioners

Traffic Director explained!

If your application is deployed in a microservices architecture then you are likely familiar with the networking challenges that come with it. Traffic Director helps you run microservices in a global service mesh.

By Priyanka Vergadia • 4-minute read



Posted in [Developers & Practitioners](#)—[Google Cloud](#)

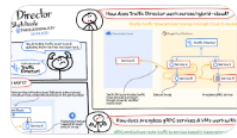


For more #GCPSketchnote, follow the [GitHub repo](#). For similar cloud content follow me on Twitter [@pvergadia](#) and keep an eye out on [thecloudgirl.dev](#).

Developers & Practitioners

Traffic Director explained!

If your application is deployed in a microservices architecture then you are likely familiar with the networking challenges that come with it. Traffic Director helps you run microservices in a global service mesh.



By Priyanka Vergadia • 4-minute read

Posted in [Developers & Practitioners](#)—[Google Cloud](#)

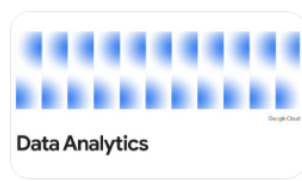
Related articles



Developers & Practitioners

Five Best Practices for Using AI Coding Assistants

By Kari Loftesness • 6-minute read



Data Analytics

Connect Spark data pipelines to Gemini and other AI models with Dataproc ML library

By Brad Miro • 3-minute read



Developers & Practitioners

Accelerate AI with Agents: Event Series for Developers in EMEA

By Abdel Sghiouar • 4-minute read



Databases

Gemini CLI extension for PostgreSQL in action: Build a fuzzy search feature in minutes

By Bala Narasimhan • 2-minute read

Follow us

