Giphy CDNs

A **CDN** is like this for the internet. It's a network of servers placed strategically around the world, each storing copies of content like GIFs so they can be delivered faster from the closest server. This cuts down on loading time, improves performance, and handles high demand smoothly.

1. Bringing GIFs Closer to You with Geographically Distributed Servers

When you request a GIF on Giphy, the CDN delivers it from the server nearest to you. For example, if you're in Europe, the CDN serves your GIF from a European server rather than one in the U.S. This geographical distribution reduces **latency** (the time it takes for a GIF to load) and ensures you get your GIF quickly.

2. Multi-Layered Caching for Faster Access

CDNs also employ **multi-layered caching** — storing copies of frequently used data at different levels to reduce retrieval time. Here's how it works:

Edge Cache: The closest layer to the user, providing the fastest access to frequently requested GIFs.
Regional Cache: A second layer that serves regions or countries, keeping a larger repository of GIFs that are popular within a given area.
Origin Server Cache: The main, centralized layer, which stores all GIFs and updates the other layers. It's the final fallback if a GIF isn't available in the upper layers.

This caching strategy reduces the load on the origin servers and minimizes data requests, making sure that users get their GIFs instant

3. Load Balancing

A CDN uses load balancing in a data center to distribute incoming requests across the available server pool to ensure that spikes in traffic are handled in the most efficient manner possible. By efficiently using available resources, load balancing is able to increase processing speeds and effectively utilize server capacity.

A CDN also uses load balancing to make changes quickly and efficiently when the availability of server resources fluctuates up or down. In the event that a server fails and failover occurs, a load balancer will redirect the traffic allocated for the failed server and distribute it proportionally across the remaining servers. This provides resiliency and reliability by increasing likelihood that hardware failures will not disrupt the flow of traffic. When a new server comes online in the data center, a load balancer will proportionately remove load from other servers and increase the utilization of the new hardware

CDNS also handle failovers -

In computer systems that require a high degree of reliability and near continuous availability, failover is used to prevent traffic from being lost when a server is unavailable. When a server goes down, the traffic needs to be rerouted to a server that is still functional. By automatically offloading tasks to a standby system or another machine with available capacity, intelligent failover can prevent disruption of service to users.

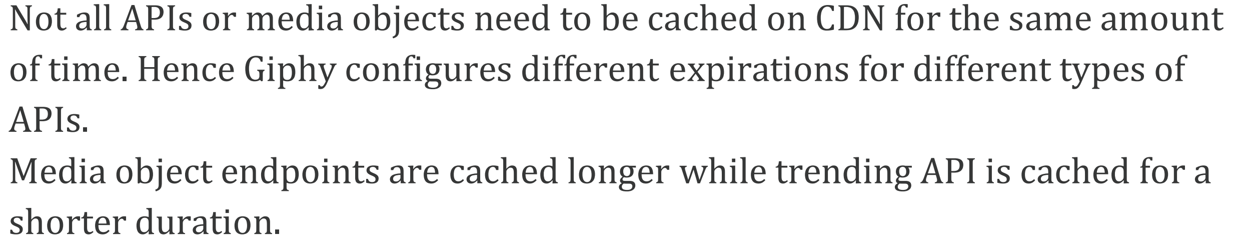
A CDN is like a GPS system coupled with express toll roads; a CDN will be able to find the optimal path to reach a distant location and will be able to use its own network to find the optimal route to get there as quickly as possible.

4. Smart Routing and Processing Closer to Users

CDNs optimize the path from the server to the user through "**smart routing**," selecting the quickest route to avoid slow or busy paths. Some CDNs also handle processing tasks — like resizing or compressing GIFs — right at the CDN server, instead of sending every request back to Giphy’s main servers. This saves time and reduces the load on the origin servers.

6. Monitoring and Adjusting Performance Constantly

To stay reliable, Giphy and CDN providers monitor how things are working in real-time. If they see delays or high demand in certain areas, they can quickly add resources, update cached files, or reroute traffic. This constant monitoring keeps GIFs available, smooth, and quick for users.



Route-specific TTL

Not all APIs or media objects need to be cached on CDN for the same amount of time. Hence Giphy configures different expirations for different types of APIs.
Media object endpoints are cached longer while trending API is cached for a shorter duration.

Response-driven TTL

Sometimes, it is the backend server that should dictate for how long the response should be cached.
Hence, Giphy, in the HTTP response from the origin server provides `max-age` headers that tell CDN the TTL for the specific response. This gives finer control over key expiration.

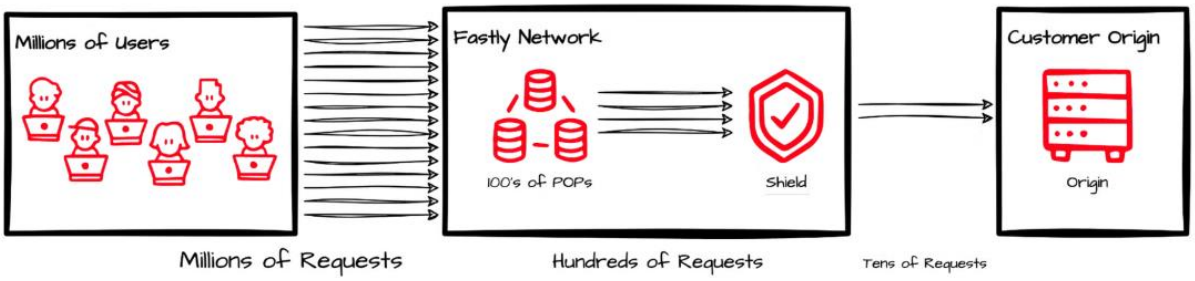
ORIGIN SHIELDING IN CDNS

The origin shield is a mid-tier caching layer between your origin server(s) and your CDN edge servers (POPs). An origin shield is a protective measure that protects your origin server(s) from overloading and allows you to unlock high availability and performance while realizing cost reduction.

For this reason, most companies implement some level of a tiered distribution caching strategy to reduce their server requirements and benefit from the performance gains that caching offers. An origin shield will always reduce the load on your origin server(s) and improve your end users' experience.

Correctly implemented, caching provides an added layer of protection and security, while shielding provides a crucial layer of midgress cache protection to ensure these outages don't happen.

An origin shield reduces the number of requests to your origin server by designating an edge node point-of-presence (POP) as the aggregation point for incoming uncached requests. The selected shield POP is an added layer of defense that protects your origin server from thousands or millions of incoming requests. Your origin server can then be locked down from the outside internet, only allowing requests from the shield POP you designate in your configuration. Directing the uncached requests through a shield POP before origin increases your cache hit ratio – enabling you to serve content faster, more efficiently, and without downtime at your origin.


Millions of Users
Fastly Network
100's of POPs
Shield
Customer Origin
Origin
Millions of Requests
Hundreds of Requests
Tens of Requests