CDN

Putting a content delivery network in front of hosted Exponential sites: what is safe to cache at the edge, trusting the CDN's addresses for HTTPS and the visitor's address, the CDN headers Velocity understands, purge…

Source file: `doc/eng-us/help/hosting/performance/cdn.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/hosting/performance/cdn.md

This page is for system administrators and product managers at hosting companies who offer a content delivery network (CDN) in front of customer sites, their own or a third party's. It answers: what Exponential sends that a CDN can cache, what the CDN must be trusted for, which CDN headers Exponential and Velocity understand, and how to keep the edge from serving stale or personal pages. Provider dashboards and rule languages are not described here; check the provider's documentation.

Contents

1. What a CDN can do for an Exponential site
2. Static files at the edge
3. Pages at the edge
4. Trusting the CDN
5. Purging
6. Certificates
7. Checks before go-live
8. Checklist

What a CDN can do for an Exponential site

Content	Cacheable at the edge?	What decides it
Design and extension stylesheets, scripts, fonts, images	yes	the lifetime the server sends (`StaticMaxAge` under Velocity)
Content images (`var/*/storage/images`)	yes	Exponential changes an image's URL when the image is updated
Anonymous pages	only if you switch on cacheable headers	`[HTTPHeaderSettings]` with `OnlyForAnonymous=enabled`
Signed-in pages, the administration, forms	no	`no-cache` by default; pages with a form token are `private`

The note about image URLs is in the shipped Apache example: "Exponential appends the version number to image URL (ezimage datatype) so when an image is updated, its URL changes" (doc/eng-us/examples/ezpublish.conf).

Static files at the edge

Velocity sends a one-year lifetime for static files (`[ServerSettings] StaticMaxAge=31536000`); under Apache or nginx you add the headers yourself (HTTP cache headers). A stylesheet replaced under the same name is not fetched again until its lifetime ends, at the edge as in the browser. Packed scripts and stylesheets of the ezjscore packer carry their modification time in the name when `ezjscore.ini [Packer] AppendLastModifiedTime=enabled` (shipped so), which keeps them safe to cache long.

Pages at the edge

Pages ship `Cache-Control: no-cache, must-revalidate`, so a CDN that honors the origin's headers stores none. To let the edge keep anonymous pages, switch on cacheable headers for anonymous visitors only, with a short `max-age` (HTTP cache headers).

> **Warning**
>
> Never let the edge store a response for a request that carried the session cookie or `is_logged_in`, and never configure the CDN to ignore the origin's `private` or `no-cache`. A signed-in page cached at the edge is served to strangers.

If the customer's site runs on Velocity, its response cache already answers anonymous pages at the origin; the CDN mostly saves the network distance and offloads static files.

Trusting the CDN

Behind a CDN, every request comes from the CDN's addresses. Without trust, "the kernel treats requests as plain HTTP from the proxy's address", which shows as `http://` URLs, redirects to `http://` and every visitor having the CDN's address (Forwarded headers).

INI

```ini
# settings/override/site.ini.append.php
[HTTPHeaderSettings]
TrustedProxies[]
TrustedProxies[]=127.0.0.1
TrustedProxies[]=::1
# the CDN's published ranges, one per line, copied from the provider's own list
TrustedProxies[]=192.0.2.0/24
ClientIpByCustomHTTPHeader=X-Forwarded-For
```

- List only the ranges the provider publishes, and keep them current: ranges change from time to time.
- Cloudflare appends to `X-Forwarded-For` and also sends `CF-Connecting-IP` (one address); either works as `ClientIpByCustomHTTPHeader`.
- A CDN in front of a load balancer: list both; the walk from the right passes through all of them.
- Make sure the origin cannot be reached around the CDN (firewall, security group), or a visitor talks to PHP directly.

Under Velocity, the engine's own list `Q.webserver.proxy.trusted` decides instead, and from a trusted proxy it treats a plain connection as HTTPS on `X-Forwarded-Proto: https`, `CloudFront-Forwarded-Proto: https` or Cloudflare's `CF-Visitor: {"scheme":"https"}` (

Chapter 8, section 8.3.9). Put the ranges in a snippet as shown on Velocity beside Apache or nginx.

Purging

Exponential sends no purge requests to a CDN. Options:

- keep anonymous page lifetimes short, so the edge refreshes on its own;
- `httpcache.ini` `[HttpCacheSettings]` `TagHeader=Cache-Tag` (Cloudflare) or `Surrogate-Key` (Fastly) names each page's purge tags for the CDN to purge by; off by default because the tags name internal ids;
- a listener on the kernel's `content/cache` event, emitted with node and object ids on every publish, can call the provider's purge API (Varnish and reverse proxies).

Certificates

When the CDN terminates TLS for the visitor, the origin still needs a certificate if the CDN connects over HTTPS. Velocity's ACME HTTP-01 challenge needs `/.well-known/acme-challenge/...` on port 80 of each name to reach the server; with a CDN in front, check that path reaches the origin, or use the documented alternatives (`dns-01` with a hook program, or your own certificate files) (Velocity HTTPS certificates). `HSTSMaxAge` is sent by the origin and passed on by the CDN; raise it only when every name is on HTTPS for good.

Checks before go-live

Shell

```
# through the CDN: static files cacheable, pages not, unless configured
curl -sI https://example.com/robots.txt | grep -i cache-control
curl -sI https://example.com/ | grep -i -E 'cache-control|x-cache|x-exp-cache'
# directly at the origin, bypassing the CDN: a forged header must change nothing
curl -s -o /dev/null -w '%{redirect_url}\n' -H 'X-Forwarded-Proto: https' http://origin.example.com/user/logout
```

Then sign in with a test account through the CDN and check that the signed-in page is answered `no-cache` and is never served to a second, anonymous browser.

Checklist

- ☐ The CDN's ranges are in `TrustedProxies[]` (or `Q.webserver.proxy.trusted`), kept current.
- ☐ The origin is not reachable around the CDN.
- ☐ Static files have long lifetimes; anonymous pages short ones, only with `OnlyForAnonymous=enabled`.
- ☐ Requests with the session cookie or `is_logged_in` bypass the edge cache.
- ☐ A purge strategy is chosen.
- ☐ Certificate renewal works with the CDN in front.

Related pages

- Varnish and reverse proxies
- HTTP cache headers
- TLS and HTTPS
- Forwarded headers are trusted only from configured proxies
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

> **Become an Exponential Partner**
>
> Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.
>
> Partners: start here | How to apply

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.