Performance

Overview of the performance section for hosting companies: the levers in order of effect, the caching layers, measuring before tuning, and a one-line guide to every page of the section.

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

This section is for system administrators and product managers at hosting companies who want Exponential sites to be fast on their machines, and who must explain to customers what is and is not included in a plan. It answers: which levers matter most, which caches exist and who clears them, and how to measure before and after a change.

Back to Hosting companies: start here.

Contents

1. The levers, in order of effect
2. Measure first
3. Pages in this section
4. What hosts get wrong most often
5. Related pages

The levers, in order of effect

The installation book lists them in this order (Chapter 10, section 10.12):

1. **Serve with Velocity** and its response cache. A cached anonymous page is answered without PHP; persistent workers keep the application, the database connection and the compiled files warm.
2. **Keep the content view cache and template-block cache on** (`site.ini [ContentSettings] ViewCaching=enabled`, `[TemplateSettings] TemplateCache=enabled`, both shipped), and warm the site after a clear.
3. **OPcache.**
4. **APCu** for Velocity's response cache memory tier and the SQL query cache in `shared` mode.
5. **The SQL query cache** (`querycache.ini [QueryCacheSettings] Mode=request` or `shared`), measured on your site.
6. **The database** (Chapter 9).

The README adds a point that matters for hosts that keep Apache or nginx: "Most of the speed came from fixing Exponential and the cache, not from the server process model ... Exponential 6.0.15+ is fast under load on Apache or Nginx too. Velocity raises the ceiling further; it is not what makes the application quick."

Measure first

`exp:benchmark` measures pages over HTTP, cached and rendered, and compares two servers back to back:

Shell

```
php bin/php/console exp:benchmark --save=var/benchmark/base.json --allow-root-user
# ... make one change ...
php bin/php/console exp:benchmark --baseline=var/benchmark/base.json --allow-root-user
```

Every figure in this section that comes from the sources is a measurement on a particular machine and site. The sources say so themselves; treat them as examples of what to measure, not as promises to customers. See Benchmarking a hosted site.

Pages in this section

Page	What it answers
PHP-FPM tuning	Pool settings that matter for Exponential, reloads, timeouts, APCu
OPcache	The right OPcache settings under PHP-FPM, Velocity and FrankenPHP, and why they differ
Caching layers	Every cache from the browser to the database, who fills it and who clears it
Velocity response cache	The cache in Velocity's parent process: settings, checks, clearing
HTTP cache headers	What browsers, proxies and CDNs may keep, and the role-aware HTTP cache
Varnish and reverse proxies	Caching proxies in front: trust, purging, what not to cache
Page caching with LiteSpeed and OpenLiteSpeed	The page cache built into LiteSpeed and OpenLiteSpeed, for anonymous visitors and crawlers, without a plugin: switching it on, checking hits, purging
CDN	Putting a content delivery network in front of customer sites
Valkey and Redis	The `sevenx_valkey_cache` extension: caches in Redis or Valkey
Static cache	Pages generated as files; why it ships off and when to use it
Cache clearing and warming	Which cache to clear for which change, in which order, and how to warm
Benchmarking a hosted site	`exp:benchmark` for hosts: fair comparisons, deploy checks, plan limits
The compiled kernel container	What `var/cache/kernelservices/` is, when it is rebuilt, how to check it and what it saves per request

What hosts get wrong most often

Mistake	Effect	Page
`opcache.validate_timestamps=0` on an Exponential site	a cache file rewritten after a clear is served in its old form until PHP restarts	OPcache
clearing rendered-output caches before the new code runs	the old output is cached again and the change "did not work"	Cache clearing and warming
reloading the wrong PHP-FPM master	the site keeps running the old code	PHP-FPM tuning

Mistake	Effect	Page
judging Velocity's memory by RSS	the pool looks many times larger than it is	Velocity workers and pools
a caching proxy that stores pages with a session cookie	signed-in pages served to other visitors	Varnish and reverse proxies
a LiteSpeed page cache with `CacheEnable public /` instead of the conditions of `.htaccess_root`	which pages are stored depends on the server policy, not on the session and address conditions	Page caching with LiteSpeed and OpenLiteSpeed
switching on the static cache for speed	every publish regenerates pages inside the editor's request	Static cache

Related pages

- Hosting companies: start here
- Web servers and engines
- Slow site
- Caches
- Benchmarking Exponential
- 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.