Sizing memory, CPU and disk

The memory, CPU and disk figures the Exponential repository actually states (PHP memory limits, Velocity workers, OPcache, APCu, SQLite page cache, MongoDB, disk health thresholds, log rotation), what grows on disk, and...

Source file: `doc/eng-us/help/hosting/basics/sizing-memory-cpu-and-disk.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/hosting/basics/sizing-memory-cpu-and-disk.md

This page is for system administrators and product managers at a hosting company who size servers and plans for Exponential sites. It answers: which figures the project's documentation and settings actually give, what consumes memory, CPU and disk on a running site, and how to measure a real site. It gives no figures the repository does not state; chapter 2.10 says the same: "The documentation gives few hard numbers, and this book does not invent others."

Contents

1. The figures the repository gives
2. Memory
3. CPU
4. Disk
5. Measure, do not guess
6. Turning measurements into plans
7. Checklist
8. Related pages

The figures the repository gives

Item	Value	Source
PHP `memory_limit`, installer floor	64M	`settings/setup.ini`, `[memory_limit] MinMemoryLimit`
PHP `memory_limit`, health check	below 128M a failure, below 256M a warning	System information
PHP `memory_limit` under FrankenPHP	512M shipped	`settings/velocity.ini`, `[FrankenPHPSettings] IniOptions[]`
One Velocity worker	about 10 MB of private memory (measured)	`settings/velocity.ini`; Velocity web server
Velocity pool	`Workers=4` shipped; the engine's own default is what fits in RAM, at most 8 per core and 64, never fewer than 4	`settings/velocity.ini`, chapter 2.10
OPcache	`memory_consumption=256`, `max_accelerated_files=20000`, `interned_strings_buffer=32`	chapter 10.12 and `velocity.ini` `[PHPSettings]`

Item	Value	Source
APCu for Velocity	`apc.shm_size=256M` (reserved, not used, until it fills)	`velocity.ini [PHPSettings]`
SQLite page cache	up to 64 MB per process that opened the database; 256 MB memory map	Database support
SQLite `VACUUM`	about twice the database's size in free disk	chapter 9.2
A MongoDB installation	RAM 512 MB minimum, 2 GB recommended; disk 2 GB minimum, 10 GB recommended	porting reference, section 24.1
Free disk, health check	under 1 GB or 5 % a failure; under 5 GB or 10 % a warning	System information
Log rotation	each log rotated at `LogFileMaxSize=2097152` (2 MB), `LogFileRotateCount=3` kept	`settings/site.ini`, `[FileSettings]`
A kernel clone with history	about 370 MB	chapter 3.2

Memory

What uses memory on a server running Exponential sites:

- **PHP processes.** Each PHP-FPM child or Velocity worker up to its `memory_limit` at peak; package installs, imports and image work need the most. Plan pool sizes so that the busiest moment fits (PHP-FPM tuning).
- **Velocity workers** hold about 10 MB of private memory each from the moment they are forked, idle or not. With `SpareWorkers` the pool keeps only that many when quiet and grows to `Workers` under load, retiring idle workers after `IdleWorkerTimeout` seconds.
- **Shared caches.** OPcache (`opcache.memory_consumption`) and APCu (`apc.shm_size`) reserve shared memory. PHP-FPM takes them from its `php.ini`; Velocity sets its own in `velocity.ini [PHPSettings] IniOptions[]`, because the command-line default for APCu is 32 MB, too small for its response cache and the SQL query cache together.
- **The database.** SQLite's page cache counts per process that used it: `SQLitePragmas[]=cache_size=-16384` lowers it to 16 MB, which helps on servers with many small SQLite sites. Database servers need their own sizing.
- **Optional services.** Solr, Valkey or Redis, if a plan includes them.

> **Note**
>
> Read Velocity's memory as its control panel shows it: the proportional set size (PSS) of the parent and its workers. The resident set size (RSS) of each forked worker counts the memory they share again and again, and adds up to far more than the machine really uses (chapter 2.10).

CPU

- **Cached pages cost little.** Velocity's response cache answers anonymous visitors in the server process without a worker. `settings/velocity.ini` records a measurement: one server answers cached pages at about 0.3 ms of CPU each, about 3,500 a second on a

12-core machine; with `Instances` above 1 the kernel spreads connections across servers (two measured 6,600 to 6,900, four 12,200 to 13,100). Renders are bounded by the workers, not by this.

- **Rendered pages** (first views after a clear, signed-in editors, search) are where CPU goes. Keep the content view cache and template-block cache on (both shipped), and leave the static cache off (chapter 10.12).
- **OPcache matters.** Under Velocity, `opcache.file_update_protection` above 0 made a render take about four times the CPU it takes under PHP-FPM; the shipped `velocity.ini` sets it to 0 (Velocity opcode cache).
- **FrankenPHP** threads default to twice the CPU cores when `[FrankenPHPSettings] Workers` is empty.
- **Cron and image work** add bursts: image variations are generated on first use, and the cronjobs run on schedule.

Disk

What grows	Where	How to keep it in check
Uploaded files and image variations	`var/storage` (per site `var/<site>/storage`)	grows with content; the main factor of a plan's disk
Caches	`var/cache` (per site `var/<site>/cache`)	cleared and rebuilt; can live on a memory file system (multi-site paths)
Logs	`var/log`, `var/vc/` for Velocity	rotation (`LogFileMaxSize`, `LogFileRotateCount`, `LogFileCompress`), plus the system's logrotate for cron logs
The SQLite database	`var/storage/sqlite3/`	room for the database twice over, for `VACUUM` and backups
Backups	off the machine, ideally	never only on the same disk
Code	the installation, `vendor/`, extension clones	Git clones with history take more than archives

The disk check of the administration (**Setup > System information**) fails under 1 GB or 5 % free and warns under 5 GB or 10 %; use the same thresholds in your monitoring. See also the disk full runbook.

Measure, do not guess

Shell

```
php bin/php/console exp:benchmark --save=var/benchmark/base.json --allow-root-user
php bin/php/console exp:benchmark kernel --repeat=30 --allow-root-user
php bin/php/console exp:system:info --json --allow-root-user
du -sh var/storage var/cache var/log
```

`exp:benchmark` requests the front page, menu pages, the search page and the admin login page and prints median, p90 and p99; it is polite by default against other hosts (Benchmarking). `exp:system:info` prints the facts and health checks of **Setup > System information** (memory limit, OPcache and APCu, disk, machine memory and load). Velocity's control panel shows the memory of the pool.

> **Tip**
>
> Build a reference site from your most common plan, measure it with these commands on your hardware, and size from that. A figure measured on your own servers is worth more than any rule of thumb.

Turning measurements into plans

The repository gives no plan sizes and no prices; these are the dimensions a host can choose to put limits on, each measurable with the commands above:

Dimension	Measured with
Disk for `var/storage`	`du -sh var/storage`
PHP memory per process	`memory_limit` and the pool's peak
Number of PHP workers	the pool size or Velocity's `Workers`
Database size	**Setup > System information**, Database card
Upload size	`upload_max_filesize`, `post_max_size`

Checklist

- ☐ `memory_limit` 256M or more for production pools.
- ☐ Velocity's `Workers` set explicitly; `SpareWorkers` considered on busy multi-site servers.
- ☐ OPcache and APCu sizes set for PHP-FPM and for Velocity separately.
- ☐ Velocity memory read as PSS.
- ☐ Disk monitoring at the System information thresholds.
- ☐ A reference site benchmarked on your own hardware.

Related pages

- Requirements at a glance
- Caching layers and Benchmarking a hosted site
- Velocity workers and pools
- Designing hosting plans
- Chapter 2.10: Memory, CPU and disk
- 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.