Operating system support

Exponential 6.0.x runs on almost any operating system with PHP 8.0 or later (PHP 7.4 usable without support) and a web server. This page sorts them honestly into tiers (tested here, known to work, expected to work, not...

Source file: `doc/eng-us/help/os-support.md` Last changed 2026-11-07
https://doc.exponential.earth/exphelp/viewer/help/os-support.md

> **Supported: PHP 8.0 to 8.5. Usable without support: PHP 7.4 (Exponential 6.0.15 and later).** Why Exponential supports PHP 7.4

This page is for administrators choosing a machine for Exponential and for developers who want to know whether their own computer can run it. The short answer: **Exponential 6.0.x runs on almost any operating system that has PHP 7.4 or later and a web server.** It is plain PHP files plus a database; it has no compiled code of its own, no architecture of its own and no dependency on one C library.

> **Warning**
>
> `.htaccess` **slows down every request.** When `.htaccess` files are enabled (`AllowOverride` anything but `None`), let alone used, Apache looks for them and reads and evaluates them on every request, in every directory of the path, and page loading times go up drastically (measured). The only real fix is to switch the feature off for the virtual host: `AllowOverride None` in the HTTPS virtual host, and the rules in the virtual host's own configuration as mod_rewrite rules. That is far faster in production speed tests and well worth the time it takes to optimize a production configuration. Switch over right away: copy the rules of the example HTTPS virtual host into your virtual host (with the static cache, also `staticcache_offload.conf`), set `AllowOverride None`, run `apachectl configtest`, reload Apache and test the site. The same rules for other servers: nginx, LiteSpeed, Caddy, IIS, Velocity (built in); where virtual host rules go in a control panel: Plesk, cPanel, DirectAdmin, ISPConfig. **On LiteSpeed or OpenLiteSpeed,** the server's own page cache can also answer anonymous visitors and crawlers without PHP: Page caching with LiteSpeed and OpenLiteSpeed.

The long answer is the rest of this page, sorted into honest tiers, because "runs anywhere PHP runs" is only useful when you also know what has been watched working, what should work, and what cannot work today and why.

Contents

1. What the operating system has to provide
2. The four tiers
3. Tier 1: tested here
4. Tier 2: known to work
5. Tier 3: expected to work
6. Tier 4: not possible today (theoretical)

7. Containers, cloud and hosting panels
8. Processor architectures
9. Web servers
10. PHP SAPIs
11. Platform-specific notes
12. Check a machine in two minutes
13. Related pages

What the operating system has to provide

Everything below comes from `composer.json`, `settings/setup.ini` and chapter 2 of the installation guide; where an item is a recommendation rather than a requirement, it says so.

Need	Minimum	Why
PHP	**8.0** (`composer.json`: `"php": "^8.0 || ^8.1 || ... || ^8.8"`; the setup wizard: `[phpversion] MinimumVersion=7.4.0`)	the kernel; 8.0 is the stock PHP of RHEL 9 and its rebuilds
PHP for Velocity	**8.0**, with `sockets` (`pcntl` and `openssl` recommended); 8.1 before Velocity 0.0.4.54	Velocity, the recommended web server, is a PHP program
PHP extensions	`dom`, `libxml`, `simplexml`, `xsl`, `mbstring`, `iconv`, `intl`, `pcre`, `json`, `session`, `zlib`, one database extension, `gd` or ImageMagick	the setup wizard's critical tests and `composer.json`
A database	SQLite 3 needs only the `sqlite3` extension; the others need their server somewhere on the network	see Database support
A web server	none, if Velocity or PHP's built-in server serves the site; otherwise anything that can send a URL to `index.php`	see Web servers
64-bit PHP	recommended	the languages of an object are bits of one integer; `composer.json` suggests `php-64bit` "for support of more than 30 languages"
Composer 2	to install	downloads the expcomponents libraries (the fork of the Zeta Components) and the extensions
Writable directories	`var/`, `settings/override`, `settings/siteaccess` and the others the wizard checks	caches, uploads, logs, the SQLite database

Nothing in that list names an operating system. That is why the list of systems below is long.

The four tiers

Tier	Meaning
1. Tested here	The project runs Exponential on it every day, or its CI runs the test suites on it on every push.

Tier	Meaning
2. Known to work	The project's documentation, packages or the Velocity engine's own CI cover it, or it is the same PHP on the same kernel as a tier 1 system. Not run by the project for Exponential every day.
3. Expected to work	A maintained PHP 8.x exists for it and nothing known stands in the way, but nobody here has watched Exponential run on it. **This is a reasoned expectation, not a test result.**
4. Not possible today	No PHP 8 exists for it, or the hardware cannot run PHP 8 at all. Listed so the question gets a straight answer.

If you run Exponential on a tier 3 system, please tell the project on the community tracker or the software tracker, and the row moves up.

Tier 1: tested here

System	What runs	Evidence
AlmaLinux 9 (x86_64, RHEL 9 family)	The project's development server: Apache 2.4 with PHP-FPM 8.5 from the Plesk panel, and Exponential Velocity on the same installation; SQLite, MariaDB, MongoDB and Oracle (in a container) sites run there, and a PostgreSQL server is available	runs every day
Ubuntu (GitHub Actions `ubuntu-latest`**)**	PHPUnit on PHP 8.1, 8.2, 8.3, 8.4 and 8.5; the database tests against MySQL 8.0 and PostgreSQL 15; PHP 8.0 syntax and boot smoke test; PHPStan; the performance benchmark on PHP 8.5	`.github/workflows/phpunit.yml`, `quality.yml`, `performance.yml`, on every push

Tier 2: known to work

Linux distributions

Distribution	PHP 8 from	Notes
Red Hat Enterprise Linux 9 and 10, **Rocky Linux**, **AlmaLinux**, **Oracle Linux**, **CentOS Stream 9 and 10**	AppStream (PHP 8.0 stock in EL 9; module streams `php:8.1` to `php:8.3`), Remi (8.0 to 8.5)	the documented reference platform (chapter 2); Velocity ships `.rpm` packages for EL 9 and EL 10; the EL 9 package installs on the stock PHP 8.0
Debian 12 and 13	the distribution (8.2, 8.4), or Ondřej Surý's repository	Velocity ships `.deb` packages for both
Ubuntu 22.04 and 24.04	the distribution (8.1, 8.3), or the `ondrej/php` PPA	Velocity ships `.deb` packages for both; the CI runs on Ubuntu
Alpine Linux (musl)	apk (`php83-*` and others)	Velocity's CI proves the server on Alpine x86_64, aarch64 and i386; Velocity's extension manifest knows its packages
Plesk hosts (on any of the above)	Plesk's own PHP builds under `/opt/plesk/php/<version>/`	the project's development server; per-domain PHP-FPM pools, service `plesk-php<XY>-fpm`

Other systems

System	Notes
FreeBSD 14, **DragonFly BSD**	Velocity's CI boots and serves on real BSD kernels in a VM; PHP 8 from pkg (`php83-*`). Exponential itself is not run there by the project.
macOS on Apple Silicon (arm64) and Intel	PHP 8 from Homebrew; Velocity publishes a `macos-arm64` binary and forks workers with `pcntl` as on Linux
Windows (x64)	Official PHP for Windows builds; Apache or IIS with FastCGI. The kernel knows Windows paths (ImageMagick's `convert.exe`, server variables of IIS and nginx on Windows in its tests). See Windows for what is missing.
Docker and Podman (Linux containers)	Velocity publishes images on `ghcr.io/se7enxweb/exponential-velocity` for amd64 and arm64, one per PHP version and variant

Tier 3: expected to work

These have a maintained PHP 8 and nothing known stands in the way. None has been watched running Exponential.

System	Why it should work	What to watch
Fedora	current PHP in the distribution; the RHEL family's upstream	none known
SUSE Linux Enterprise, **openSUSE** Leap and Tumbleweed	PHP 8 packages (`php8-*`)	package names differ from RHEL and Debian
Arch Linux, **Manjaro**	always the newest PHP	the newest PHP may be newer than the newest tested (8.5)
Gentoo	PHP 8 ebuilds with USE flags	build `intl`, `xsl` and the database extension in
Slackware	PHP 8 in current releases	none known
NixOS	PHP 8 in nixpkgs, extensions declared in the configuration	the installation directory must stay writable, which the Nix store is not: keep the site outside it
Amazon Linux 2023	PHP 8 packages	none known
cPanel and **DirectAdmin** hosts	EasyApache (EA-PHP 8.x), per-account PHP	shared hosting may forbid long-running processes, so Velocity may not be allowed; use Apache or LiteSpeed with `.htaccess`
OpenBSD, **NetBSD**	PHP 8 in ports and pkgsrc	in Velocity's platform matrix, still being worked on there
illumos: OmniOS, OpenIndiana, SmartOS	PHP 8 packages	Velocity's CI boots its server on OmniOS but serving a request still fails there (an open issue in the engine); Apache or nginx with PHP-FPM is the path to try
Oracle Solaris 11	PHP 8 packages exist	no runner anywhere in the project

System	Why it should work	What to watch
IBM AIX on Power	IBM ships PHP 8 in the AIX Toolbox	no runner, no emulator
IBM i (PASE)	a maintained PHP 8 is published for IBM i	no runner; database through the usual extensions over the network
Windows Subsystem for Linux 2	a real Linux kernel; any of the distributions above	keep the site in the Linux file system, not under `/mnt/c`, for speed and for SQLite's locking
Kubernetes	it runs the containers above	see Containers: one writer for SQLite
Haiku	PHP is in HaikuPorts	Velocity's CI cannot install it unattended; PHP's built-in server or Apache are the paths to try
Android via **Termux**	Termux packages a current PHP 8	a phone is a development toy, not a server: storage, background limits, no fixed address

Systems the project could not place in a tier. HP-UX on Itanium: we know of no current PHP 8 build for it; treat it as tier 4 unless your vendor has one. **OS/2 and ArcaOS**: earlier PHP versions were ported; whether a PHP 8 port exists is unverified here.

Tier 4: not possible today (theoretical)

Stated plainly, because these questions come up and deserve a straight answer. In every case the obstacle is the same: **there is no PHP 8 for the system**, and Exponential is PHP. The day someone ports PHP 8 to one of them, the files run unchanged.

System	Why not
Commodore 64, **Apple II**, other 8-bit machines	a 6502-class processor with 64 KB of memory. PHP needs a 32- or 64-bit CPU with memory management and tens of megabytes; nothing about these machines fits, so this one is a hardware limit, not missing work.
AmigaOS 4, **MorphOS**, **AROS**	real multitasking systems with gigabytes of memory on modern hardware, but their PHP ports stop at 5.x. The gap is porting work (C library, toolchain, sockets), not hardware. The Velocity engine's README maps what that port would take.
Classic Amiga (68000, AmigaOS 3.x and older)	no PHP 7 or 8 port; the 68000 machines also lack the memory management PHP 8 expects
MS-DOS, **FreeDOS**	a single-tasking 16-bit system; no PHP 8 port, no network stack PHP could use as it is, no process model for a web server
Classic Mac OS (System 7 to Mac OS 9)	no PHP port of any current version; cooperative multitasking; no POSIX layer
BeOS	discontinued since 2001; its successor Haiku is in tier 3
Plan 9, **9front**	no PHP port known to us; PHP assumes a POSIX C library Plan 9 does not provide
RISC OS	no PHP 8 port known to us
IBM z/OS (UNIX System Services)	no current PHP 8 for USS known to us. Linux on IBM Z (s390x) is a different matter: Velocity's CI proves its server there, so Linux on the same mainframe is the path.

System	Why not
Windows XP, **Windows 7** and other Windows releases Microsoft no longer supports	current PHP 8 builds target supported Windows releases; whether a given PHP 8 build still starts on an older release is unverified here, and such a machine gets no security fixes anyway

Pages like this one are honest about tier 4 so that nobody spends a weekend on a machine that cannot work, and so that anyone who wants to make one work knows that the work is a PHP port and not an Exponential port.

Containers, cloud and hosting panels

- **Docker, Podman, Kubernetes.** Any image with PHP 8 and the extensions above runs Exponential; Velocity's images (`ghcr.io/se7enxweb/exponential-velocity`, amd64 and arm64) include the server. Keep `var/` (storage, logs, the SQLite database) and `settings/override` on a persistent volume. With SQLite, run **one** container that writes: SQLite's WAL needs shared memory on one host, so several replicas cannot share one database file over a network volume. Several replicas need a database server (PostgreSQL first; see Database support) and shared storage for `var/storage`.
- **Cloud virtual machines** (Amazon EC2 including Graviton, Google Compute Engine, Azure, Hetzner, DigitalOcean and the rest) are ordinary Linux machines: tier 2 or 3 by their distribution.
- **Platform as a service** offerings that run PHP 8 with a writable persistent disk are expected to work (tier 3). Offerings with a read-only or ephemeral file system need `var/` moved to persistent storage and a database server.
- **Hosting panels.** Plesk is tier 1 (the development server). cPanel and DirectAdmin are tier 3; on shared hosting without shell access, the setup wizard in the browser is the installer.

Processor architectures

Exponential has no machine code, so the architecture matters only through PHP and the database:

Architecture	Status
x86_64 (amd64)	tier 1
aarch64 (ARM64): Apple Silicon, AWS Graviton, Ampere, Raspberry Pi 4 and 5	tier 2: Velocity publishes aarch64 binaries and images and proves them in CI; PHP and every database here exist for it
armv7, **armv5** (Raspberry Pi 2 and 3, older boards)	tier 3: Velocity's CI proves its server there; 32-bit PHP limits a site to 30 languages
i386 (32-bit x86)	tier 3, with the same 30-language limit
ppc64le (IBM Power), **s390x** (IBM Z, big-endian)	tier 3: Velocity's CI proves its server on both, and its full suite passes on s390x, which is big-endian and where byte-order mistakes would show
riscv64	tier 3: Velocity's CI passes under emulation with PCRE's JIT turned off (`pcre.jit=0`), because the emulator, not RISC-V, crashes with it on; real hardware is untested

Web servers

Any web server that can pass a request to PHP and rewrite a URL to `index.php` serves Exponential. Setting each one up is the subject of chapter 8, Serving the site.

Server	Status	Notes
Exponential Velocity (the `qbix` engine)	recommended for every stage; tier 1	a PHP program that serves HTTP and HTTPS itself with a pool of persistent workers; PHP 8.0+ (from release 0.0.4.54); no other web server needed
Apache httpd 2.4 with PHP-FPM (or mod_php)	tier 1	the classic setup and shared hosting; rewrite rules in `.htaccess_root`
FrankenPHP (through `exp:velocity`, `Engine=frankenphp`)	production-ready alternative	Caddy with PHP built in, one binary
nginx with PHP-FPM	tier 2	rules in the server block, see 8.6
LiteSpeed, **OpenLiteSpeed**	tier 3	reads Apache's rewrite rules in most cases
Caddy with PHP-FPM	tier 3	`php_fastcgi` and a rewrite to `index.php`
Microsoft IIS with FastCGI	tier 3	the URL Rewrite module (`web.config`); the kernel's request handling has tests for IIS 7 server variables
lighttpd	tier 3	`mod_rewrite`
PHP's built-in server (`exp:velocity start --engine=php`)	development only	one process, no HTTPS

PHP SAPIs

SAPI	Use
FPM (`fpm-fcgi`)	the usual way behind Apache, nginx, Caddy and the others; one pool per site
CLI	Composer, the installers, the console (`bin/php/console`), cronjobs, and Velocity itself, whose workers are CLI processes
Apache module (`apache2handler`, mod_php)	works; ties PHP to Apache's prefork model, so PHP-FPM is preferred
CGI / FastCGI (`cgi-fcgi`)	IIS on Windows; Velocity's workers on Windows are `php-cgi` processes
LiteSpeed (`litespeed`, LSAPI)	LiteSpeed's own SAPI
FrankenPHP	its own embedded SAPI
built-in web server (`cli-server`)	development only

The web PHP and the command line PHP are configured separately and are often different builds; give both the same version and extensions (chapter 2).

Platform-specific notes

Linux

The reference platform. Package names per family are in chapter 2, and `php bin/php/console exp:velocity ext install-hint <extension>` prints the install command for the system it runs on.

PHP 8.0 and Red Hat Enterprise Linux

Exponential Velocity runs on PHP 8.0 to 8.5 from its release 0.0.4.54 (20 October 2026), with nothing extra on 8.0. On RHEL 9 and its rebuilds the default AppStream PHP 8.0 works, with the el9 rpm (it depends on `php-cli >= 8.0`, no module stream to enable) or with Composer. On RHEL 8, whose default PHP is 7.2, enable the 8.0 stream (`dnf module enable php:8.0`) and use the phar or Composer; there is no el8 rpm. On PHP 8.0, AVIF image output (needs 8.1) and the dashboard's per-request memory peak (needs 8.2) are not available, `$_FILES` has no `full_path`, and a class declared with `eval()` that has static properties makes its worker get replaced after that request. Details: chapter 2 and the engine's PHP 8.0 and Red Hat Enterprise Linux page (extensions, systemd, SELinux, verification).

BSD

PHP 8 and every database extension are in FreeBSD's `pkg` and in OpenBSD ports and NetBSD's pkgsrc. `pcntl` (for asynchronous publishing and Velocity's worker pool) is available. ImageMagick is found by the setup wizard under `/usr/local/bin`.

macOS

Homebrew's `php` formula carries the extensions Exponential needs; Velocity runs with forked workers as on Linux. For development, SQLite and Velocity (or PHP's built-in server) need nothing else.

Windows

- Use the official PHP 8 builds for Windows (x64, non-thread-safe for FastCGI) with Apache or IIS. They include the extensions Exponential needs, `intl` among them.
- `pcntl` and `posix` do not exist on Windows: asynchronous publishing (`bin/php/ezasynchronouspublisher.php`) is not available, and Velocity runs its workers as `php-cgi` subprocesses without copy-on-write memory sharing.
- Velocity's single-file Windows binaries are built **without** `intl`, which Exponential's setup requires, so run Velocity on Windows through the phar with the official PHP rather than through its binary (the phar path on Windows is expected to work but untested, according to the Velocity engine's own platform notes).
- The documentation and the shell commands in this help are written for Linux; translate paths and permissions.
- WSL2 runs the Linux tiers above and is the simpler path for development on Windows.

Containers

See Containers, cloud and hosting panels.

Check a machine in two minutes

Shell

```
php -v                # 8.0 or later, for Velocity too
php -m                # the extensions listed above
php -i | grep "Integer Size"   # 64 bits recommended
```

Then let the installer check the rest: the setup wizard's **System check** page, or `exp:install` and the kickstarter, which run the same critical tests and print the fix for each failure. With Velocity installed, `php bin/php/console exp:velocity ext check` compares the machine with the extension set Velocity expects.

Related pages

- Chapter 2, Requirements and Chapter 8, Serving the site.
- Database support: the supported databases, in order.
- Velocity packages and binaries, Velocity web server, Velocity engines.
- PHP 8.0 support, Continuous integration.
- Deploying.
- Free documentation: these pages are free to copy, translate and improve.

> **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.