FrankenPHP

FrankenPHP for hosts: what the production-ready alternative engine is, how exp:velocity installs and verifies the binary, its embedded PHP that reads no php.ini, threads, HTTPS, logs, its limits against Velocity, and...

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

This page is for system administrators at hosting companies who consider FrankenPHP for customer sites. It answers: what FrankenPHP is in Exponential, how it is installed and verified, what its embedded PHP needs, how it compares with Velocity's own engine, and what goes wrong first. The sources are Chapter 8, section 8.4 and FrankenPHP as the Exponential web engine.

Contents

1. What it is
2. FrankenPHP or Velocity's qbix engine
3. Install and start
4. The embedded PHP
5. Threads
6. HTTPS
7. Configuration, control and logs
8. Troubleshooting
9. Checklist

What it is

FrankenPHP is the Caddy web server with PHP built in, as one binary (150 to 180 MB per platform). Velocity downloads it, verifies it and drives it with the same commands as its other engines. It runs in **classic mode**: a pool of PHP threads, each request starting from a clean state as under PHP-FPM. FrankenPHP's worker mode is deliberately not offered, because the kernel keeps request state in globals and would leak one request into the next.

`exp:frankenphp <verb>` is the same as `exp:velocity <verb> --engine=frankenphp` (Server control commands).

FrankenPHP or Velocity's qbix engine

	`qbix` **(Velocity)**	`frankenphp`
Role	recommended for every stage	production-ready alternative
PHP	the machine's, 8.0 or later (from Velocity 0.0.4.54)	the binary's own (PHP 8.5 in the official 1.12 releases)
Requests	persistent workers	classic mode, clean state per request

	qbix **(Velocity)**	frankenphp
Response cache	yes	not yet ([CacheSettings] is ignored)
TLS and HTTP/2	yes	yes
graceful	re-exec, socket kept	configuration reload, no dropped connection
Configuration	the /etc/vc tree	a generated Caddyfile
Machine PHP and PHP-FPM to maintain	the machine's PHP	none for serving (the console still uses the machine's PHP)

Choose FrankenPHP when you want one verified binary with TLS and HTTP/2 and PHP-FPM-style isolation per request. Choose `qbix` when you want persistent workers and the response cache (Velocity engines).

Install and start

Shell

```
php bin/php/console exp:velocity config set --apply ServerSettings Engine frankenphp --allow-root
-user
php bin/php/console exp:velocity install --allow-root-user          # optional: start installs it
 when missing
php bin/php/console exp:velocity install --check --allow-root-user  # is the pinned binary in pla
ce?
php bin/php/console exp:velocity start --allow-root-user
php bin/php/console exp:velocity status --all --allow-root-user
```

- `install` fetches the release asset for this machine for `[FrankenPHPSettings] Version` (shipped `1.12.7`) and refuses it unless its SHA-256 matches the `Sha256[<asset>]` line pinned beside it. The binary is stored in `var/vc/frankenphp/bin/`.
- `Variant` chooses the Linux build: `gnu` (default, can load `.so` extensions), `musl` (fully static) or `mimalloc`.
- `install --from=<file>` installs a copied file on a machine without access to GitHub, checked the same way.
- If a release was re-cut upstream and the pin no longer matches, the install fails with a SHA-256 mismatch. The source page describes how to verify the published digest and set the pin in an override (FrankenPHP, section 2.4). Never switch verification off to make an install pass.

Shipped ports: `8089` for HTTP and `8444` for HTTPS (`[FrankenPHPSettings] Port`, `HTTPSPort`), so it can run beside the `qbix` engine for tests.

The embedded PHP

The binary reads no `php.ini` of the machine. Everything the machine's `php.ini` would provide has to be given as `IniOptions[]` lines, in this order (later wins): `[PHPSettings] IniOptions[]` (shared with `qbix`), then `[FrankenPHPSettings] IniOptions[]`. Or `[FrankenPHPSettings] PHPIniFile` names a `php.ini` of your own.

Shipped `[FrankenPHPSettings] IniOptions[]`: `display_errors=Off`, `log_errors=On`, `memory_limit=512M`, `opcache.enable=1`, `post_max_size=64M`, `upload_max_filesize=32M`.

The OPcache tuning comes from `[PHPSettings]`.

> **Warning**
>
> The trap hosts meet first: with `[DatabaseSettings] Server=localhost`, the MariaDB and MySQL drivers connect through a socket whose path normally comes from `php.ini`. The embedded PHP has none, the connection fails and the site shows "An unexpected error has occurred." Set the socket explicitly, or connect over TCP with `Server=127.0.0.1` (FrankenPHP, section 6.1).

INI

```
# settings/override/velocity.ini.append.php
[PHPSettings]
IniOptions[]=mysqli.default_socket=/var/lib/mysql/mysql.sock
IniOptions[]=pdo_mysql.default_socket=/var/lib/mysql/mysql.sock
```

Find the real path with `php -i | grep -i "default_socket"`. The same applies to any other value a `php.ini` provides: a timezone, `sendmail_path`, `open_basedir`, `session.save_path`, an extension's configuration. When something works under Apache but not under FrankenPHP, suspect a `php.ini` value the embedded PHP never saw.

The binary ships a broad set of extensions, but not necessarily every one a site needs. On the `gnu` variant a missing extension can be loaded as `IniOptions[]=extension=/path/ext.so` if it matches the binary's PHP version; otherwise an own build is needed.

Threads

`[FrankenPHPSettings] Workers` is the number of PHP threads; empty means FrankenPHP's default, twice the CPU cores. `SpareWorkers` adds threads under load. These do **not** fall back to `[ServerSettings]`. Measured on a 12-core test installation, more threads than that were slower under heavy load: at 64 concurrent requests, 12 to 48 threads did about 10 % more work than 96 or 590, with up to 24 % less CPU per page (FrankenPHP, Threads). Set `Workers` explicitly only after measuring your own pages.

FrankenPHP is a Go program; `GOMEMLIMIT` and `GOMAXPROCS` are environment variables of the process that runs `exp:velocity start`, not Caddy configuration.

HTTPS

On by default (`HTTPS=enabled`) on `HTTPSPort` beside HTTP on `Port`. It uses `[HTTPSSettings] Certificate` and `Key` when both are set; with both empty it makes a self-signed certificate in `var/vc/frankenphp/tls/` (valid a year, renewed a month before it runs out). A public site names its certificate; after replacing the files, run `exp:velocity graceful`. ACME is not managed for this engine: `exp:velocity ssl` is the `qbix` engine's and refuses here, and the generated Caddyfile sets `auto_https off`.

Configuration, control and logs

What	Where
Generated Caddyfile	`var/vc/frankenphp/run/Caddyfile`, regenerated and validated on every `start`, `graceful` and `restart`; never edit it

What	Where
Your own Caddy directives	a file named by `[FrankenPHPSettings] SiteInclude`
Admin API	owner-only socket `var/vc/frankenphp/run/admin.sock`; `stop` and `graceful` use it
Logs	`var/vc/frankenphp/log/access.log` and `error.log`, Caddy's JSON format
Compression	`[FrankenPHPSettings] Compression` (`zstd br gzip`), or `disabled`

Shell

```
php bin/php/console exp:velocity ctl caddyfile --engine=frankenphp --allow-root-user   # print wh
at would be generated
php bin/php/console exp:velocity ctl validate  --engine=frankenphp --allow-root-user
php bin/php/console exp:velocity ctl version   --engine=frankenphp --allow-root-user
```

Troubleshooting

Symptom	Cause	Fix
`SHA-256 mismatch ... expected ... got ...` on install	the pin is stale or the asset or variant is wrong	verify the published digest, set the pin, `install --force`
database connection fails, "An unexpected error has occurred."	no MariaDB or MySQL socket in the embedded PHP	the `IniOptions[]` socket lines, or `Server=127.0.0.1`
port not reachable from outside	firewall closed, or `Host` bound to loopback	open the port; set `Host`
browser certificate warning	the self-signed certificate is in use	name a real certificate in `[HTTPSSettings]`
`graceful` says "reload rejected"	the new Caddyfile (usually `SiteInclude`) is invalid	fix it; the running configuration was kept
works under Apache, 500 here	a `php.ini` value the embedded PHP never saw	add it as `IniOptions[]` or use `PHPIniFile`

Checklist

- ☐ Binary installed and verified (`install --check`); the pin matches.
- ☐ Database socket or TCP connection set for the embedded PHP; the site answers.
- ☐ Every `php.ini` value the site depends on given as `IniOptions[]` or through `PHPIniFile`.
- ☐ A CA certificate named in `[HTTPSSettings]`, renewed by your tool, followed by `graceful`.
- ☐ `Workers` left at the default or set from a measurement.
- ☐ Plans that rely on the response cache use `qbix` instead.

Demo mode under FrankenPHP. Set `EXP_DEMO_MODE=1` in the environment of the FrankenPHP process (its unit, or the environment Velocity starts it with) for a public demo only. Demo mode for hosts.

Related pages

- Velocity for hosts
- Caddy
- Choosing a web server
- FrankenPHP as the Exponential web engine
- 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.