PHP

Running PHP on Apache httpd

The scope of this document is to enumerate all possible configuration scenarios to allow httpd to serve php content.

Using proxy_fcgi and php-fpm (for 2.4)

This method is preferred above all other recipes, and is suitable for versions 2.4 and newer. Not only it allows you to discern and diagnose problems related to PHP more quickly, but also you will be able to use a faster and more scalable threaded mpm such as event or worker, which will significantly reduce the memory footprint of your httpd server.

Please see the following article

Using php with fastcgi (for 2.2 and older)

This method is suitable for versions 2.2 and older. It allows for a threaded mpm such as worker, which will significantly reduce the RAM requirements on your server.

Please see the following article

Using php with fcgid (for 2.2 and older)

This method is suitable for versions 2.0 or 2.2. It allows for a threaded mpm such as worker, which will significantly reduce the RAM requirements on your server. And mod_fcgid is an official Apache module.

Please see the following article

Setting up wordpress with php-fpm

Please see the following article

Using mod_php as a DSO with a threaded mpm (2.0 and newer)

This approach is identical to the next recipe, with the exception that a threaded mpm such as event or worker can be used. The main requirement is that the php system libraries and DSO must be (re)compiled with the thread-safe flags.

If apache httpd 2.0 or older is used, it must be recompiled to change the mpm. For 2.4, loading the appropriate mpm module suffices.

Special care must be taken to ensure that workers are restarted often enough (MaxConnectionsPerChild > 0) since child processes are still prone to php memory leaks and processes are likely to consume large amounts of RAM and deplete the available system resources.

This is probably the least used approach of all, due to the headaches of maintaining a thread-safe php library, and since most linux distributions do not ship those packages.

Using mod_php as a DSO (legacy)

This method is the oldest and slowest possible configuration. It was suitable for version 2.2 and older, and requires the use of the prefork mpm.

Why you shouldn't use mod_php with the prefork mpm anymore

- mod_php is loaded into every httpd process all the time. Even when httpd is serving static/non php content, that memory is in use.
- mod_php is not thread safe and forces you to stick with the prefork mpm (multi process, no threads), which is the slowest possible configuration

How to use it anyway

First, the module must be loaded:

```
LoadModule php5_module lib/httpd/modules/libphp5.so
```

Then, add the handler for the dso:

```
# Then, configure the handler for all files that end with .php

# A regexp such as \.(php|php4|php5)$ can also be used to support more extensions

<FilesMatch \.php$>
```

```
SetHandler application/x-httpd-php
</FilesMatch>
```

References:

Official php installation and configuration instructions

Common issues:

Why is my php file offered as a download with mod_php?