Variants connecting Apache with PHP5-FPM

Introduction

Thanks to the PHP-FPM project and major distros picking it up (usually as `php5-fpm`) newer shared hosting setups provide a set of per-user pools of PHP-FPM processes, combining the advantages of a running PHP process (caching of compiled version of .php files) and SuExec-style execution of PHP scripts under specific low-right system user (protecting shared hosting customers' data from each other).

However, there about 10 different approaches to connect Apache with PHP5-FPM, and about 1000 HowTos ...

SetHandler approach

From Apache 2.4.10 (?) on it's simple. Actually painfully simple compared to earlier approaches:

```
# i.e. in VirtualHost
<IfModule mod_fastcgi.c>
<FilesMatch \.php$>
    SetHandler "proxy:fcgi://127.0.0.1:9000/"
    # or with socket
    # SetHandler "proxy:unix:/var/run/php5-fpm.sock|fcgi://localhost"
</FilesMatch>
</IfModule>
```

Based on http://blog.remirepo.net/post/2014/03/28/PHP-FPM-and-HTTPD-2.4-improvement

You may have to enable the proxy_fcgi module.

Advantages

Simple.

Should be evaluated late enough to allow for heavy .htaccess Rewrite orgies.

More secure? (I think FilesMatch only to existing files that really end in "`.php`", but I'm not 100% sure yet)

Disadvantages

HTTP Basic Authentication does not work for PHP scripts. Cookie auth works.

ProxyPass approach

To avoid what looks like a "subdirectory move" to Rewrite, one might use `ProxyPassMatch` (or `LocationMatch` + `ProxyPass`) instead of `FastCgiExternalServer` and `Action`, as proposed in https://wiki.apache.org/httpd/PHP-FPM

Advantages

Works before Apache 2.4.10

Disadvantages

May be evaluated too early, circumventing .htaccess Rewrite orgies.

Security risks, see bottom of https://wiki.apache.org/httpd/PHP-FPM

Not sure if HTTP Basic Authentication works for PHP script.

FastCgiExternalServer / Action approach

Advantages

Works before Apache 2.4.10

HTTP Basic Authentication works for PHP scripts.

Definitely evaluated late enough to allow for .htaccess Rewrite orgies, but works as a Rewrite itself which can lead to Rewrite loops.

Disadvantages

Complicated to configure.

Works as a Rewrite itself which can lead to Rewrite loops.

Details

This was the first almost-satisfying approach I found, but it has major disadvantages.

A typical setup may look like this:

```
# once for each DocumentRoot
<IfModule mod_fastcgi.c>
  FastCgiExternalServer /home/jdoe/www.jdoe/php5-fcgi -socket /var/run/php5-fpm-jdoe.sock -pass-header
Authorization
</IfModule>

# once for each VirtualHost, often twice for :80 and :443
<IfModule mod_suexec.c>
  SuexecUserGroup jdoe doe
</IfModule>
<IfModule mod_fastcgi.c>
  AddType application/x-httpd-fastphp5 .php
  Action application/x-httpd-fastphp5 /php5-fcgi
</IfModule>
```

Unfortunatley, the `Rewrite` engine can get confused by this, because it sees requests for say `/foobar.php` scripts with a path like `/php5-fcgi /foobar.php`.

This can trigger `Rewrite` "loops", producing a `500` response code for the client (browser) and something like this in the Apache `error.log`:

```
AH00124: Request exceeded the limit of 10 internal redirects due to probable configuration error. Use
'LimitInternalRecursion' to increase the limit if necessary. Use 'LogLevel debug' to get a backtrace., referer:
https://.../wp-admin/
```

Further Reading: Action directive, FastCgiExternalServer directive.

Circumventing Rewrite loops, general approach

Exclude `/php5-fcgi` (or whatever virtual path the LAMP setup uses) from problematic `RewriteRules`:

```
# before RewriteRule:
RewriteCond %{REQUEST_URI} !^/php5-fcgi/*
```

Circumventing Rewrite loops in typical .htaccess WordPress

A typical .htaccess for WordPress looks like this:

.htaccess vulnerable to Rewrite-Looping

```
# BEGIN WordPress
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteRule ^index\.php$ - [L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]
</IfModule>
# END WordPress
```

To avoid the rewrite loop, change it to this:

Ironized .htaccess

```
# BEGIN WordPress
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteRule ^index\.php$ - [L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_URI} !^/php5-fcgi/*
RewriteRule . /index.php [L]
</IfModule>
# END WordPress
```