FreeBSD Administration - Support #385

Setting Up A FreeBSD, Apache 2.2, MariaDB 5.5, PHP 5.4 (FAMP) Server

05/02/2014 10:42 PM - Daniel Curtis

Status:	Closed	**Start date:**	05/02/2014
Priority:	Normal	**Due date:**	
Assignee:	Daniel Curtis	**% Done:**	100%
Category:	Web Server	**Estimated time:**	3.00 hours
Target version:	FreeBSD 9	**Spent time:**	15.00 hours

Description

Here is a procedure to install a FAMP, FreeBSD with Apache, MariaDB and PHP, server. The following setup runs Apache 2.2, MariaDB 5.5, and PHP 5 on FreeBSD 9.2. If any version of the packages needs to be changed, replace the versions in the commands accordingly.

Pre-Installation Tasks

- Before installation of the components, make sure the ports tree and packages are up to date using the following command:

```
pkg update && pkg upgrade && portsnap fetch extract
```

Apache 2.2 & PHP – Installation and Configuration

- Install Apache 2.2 with mod_php5

```
pkg install apache22 php5 mod_php5 php5-xml php5-dom libgpg-error php5-xmlreader php5-simplexml php5-ctype php5-fileinfo php5-openssl php5-hash php5-filter openldap-client xproto xextproto libXau libXdmcp libpthread-stubs kbproto libICE freetype2 png jpeg printproto php5-exif curl php5-json php5-sqlite3 php5-pdo php5-mysql php5-mysqli php5-pdo_mysql oniguruma4 php5-iconv mp3info php5-zlib php5-zip php5-bz2 php5-session libgcrypt php5-wddx php5-ldap libxcb libSM php5-curl php5-pdo_sqlite php5-mbstring libxslt libX11 libXt php5-xsl libXext libXp libXmu libXpm libXaw t1lib php5-gd
```

NOTE: This installs a few PHP extensions, add or remove as needed.

- Edit the apache configuration file, i.e. /usr/local/etc/apache22/httpd.conf, and make the following changes:

```
ServerRoot "/usr/local"
ServerAdmin you@your.address
ServerName www.example.com:80
DocumentRoot "/usr/local/www"
Listen :80
NameVirtualHost *:80
...
<Directory "/usr/local/www">
    Options Indexes FollowSymLinks

    AllowOverride None

    Order allow,deny
    Allow from all
</Directory>
...
<IfModule dir_module>
    DirectoryIndex index.php index.html
</IfModule>
```

```
    <FilesMatch "\.php$">
        SetHandler application/x-httpd-php
    </FilesMatch>
    <FilesMatch "\.phps$">
        SetHandler application/x-httpd-php-source
    </FilesMatch>
    ...
    Include etc/apache22/Includes/*.conf
```

Now create a vhost website in /usr/local/etc/apache22/Includes/www.example.com.conf:

```
<VirtualHost *:80>
    ServerAdmin root@example.com
    DocumentRoot "/usr/local/www/www.example.com"
    ServerName www.example.com
    ErrorLog "/var/log/www.example.com-error_log"
    CustomLog "/var/log/www.example.com-access_log" common
</VirtualHost>
```

- Edit the /etc/hosts file and add the following line:

  <ip-address> <hostname>.<domain>

- eg:

  192.168.1.1 www.example.com

- (Optional) Create a file named /boot/loader.conf or edit it if it is already present and add the following line:

  accf_http_load="YES"

- Add the following line to /etc/rc.conf:

  echo 'apache22_enable="YES"' >> /etc/rc.conf

- Test the apache server installation using the following command:

  ```
  /usr/local/sbin/apachectl start
  ```

MariaDB – Installation and Configuration

Install MariaDB 5.5

```
cd /usr/ports/databases/mariadb55-server/
make install
make clean
```

- Add the following line to the file /etc/rc.conf:

  ```
  echo 'mysql_enable="YES"' >> /etc/rc.conf
  ```

- Start MariaDB

  ```
  service mysql-server start
  ```

- Set password for mysql using the following command

```
rehash
mysqladmin -uroot password
```

Configure MariaDB

- Use the following command:

```
cp /usr/local/share/mysql/my-small.cnf /etc/my.cnf
```

Restart mysql using the following commands:

```
service mysql-server restart
```

PHP – Installation and Configuration

- Install PHP5 and other supporting packages:

```
cd /usr/ports/lang/php5
make config
make install clean
```

NOTE: I decided compile the FPM module into PHP to replace the old FastCGI method. Make sure to do the same while running make config.

- Copy the PHP configuration file using the following command

```
cp /usr/local/etc/php.ini-production /usr/local/etc/php.ini
```

- Configure the mysql module for PHP:

```
cd /usr/ports/databases/php5-mysql
make config
```

NOTE: Make sure to enable the **mysql** extension

- Install the mysql module for PHP:

```
make install
make clean
```

- Install the php5-session package

```
cd /usr/ports/www/php5-session
make install clean
```

- Configure the php5-extensions package

```
cd /usr/ports/lang/php5-extensions
make config
```

NOTE: Enable all the modules that will apply to your PHP needs.

- Install the php5-extensions package

```
make install clean
```

- Install and configuring apache module for PHP:

```
cd /usr/ports/www/mod_php5
make install
make clean
```

- Edit /usr/local/etc/apache22/httpd.conf file and add the following lines:

```
LoadModule php5_module        libexec/apache22/libphp5.so

AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

- You should also search for the line that reads:

  DirectoryIndex index.html

1. and change it to read:

   DirectoryIndex index.php index.html

- Once completed, a simple call to the apachectl command for a graceful restart is needed to load the PHP module:

```
apachectl graceful
```

For future upgrades of PHP, the make config command will not be required; the selected OPTIONS are saved automatically by the FreeBSD Ports framework.

The PHP support in FreeBSD is extremely modular so the base install is very limited. It is very easy to add support using the *lang/php5-extensions* port. This port provides a menu driven interface to PHP extension installation. Alternatively, individual extensions can be installed using the appropriate port.

- Now restart the apache server by using the following command:

```
/usr/local/sbin/apachectl restart
```

Apache 2.2 Modules – Installation and Configuration

There are many modules that apache can use, the following are just a few that I use in my web server baseline. For more modules, consult the FreeBSD handbook.

- **Install & configure PHP-FPM and FastCGI with SuExec**
  1. Start by enabling the php-fpm service

```
echo 'php_fpm_enable="YES"' >> /etc/rc.conf
service php-fpm start
```

1. Then install FastCGI

```
pkg install ap22-mod_fastcgi
```

1. And add the module to /usr/local/etc/httpd.conf, or module includes directory:

```
LoadModule fastcgi_module     libexec/apache22/mod_fastcgi.so
LoadModule suexec_module        libexec/apache22/mod_suexec.so
```

1. Next add the FastCGI global configuration to either the apache configuration like above, or in the includes directory, ie. /usr/loca/etc/apache22/Includes/php-fpm.conf:

```
FastCgiIpcDir /tmp/
FastCgiConfig -autoUpdate -singleThreshold 100 -killInterval 300 -idle-timeout 240 -maxClassPr
ocesses 1 -pass-header HTTP_AUTHORIZATION
FastCgiWrapper /usr/local/sbin/suexec

<FilesMatch \.php$>
SetHandler php5-fcgi
</FilesMatch>

Action php5-fcgi /fcgi-bin

<Directory /usr/local/sbin>
Options ExecCGI FollowSymLinks
SetHandler fastcgi-script
Order allow,deny
Allow from all
</Directory>
```

1. **Configure FPM**
   - Now FPM needs some configuration. Create a directory to store per-vhost fpm configs:

     ```
     mkdir /usr/local/etc/fpm.d
     ```

   - Then edit the global php-fpm.conf, uncommenting:

     ```
     include=/usr/local/etc/fpm.d/*.conf
     ```

   - Switching the listen statement from a tcp port to:

     ```
     listen = /tmp/php-fpm.sock
     ```

   - and changing the pm to:

     ```
     pm = ondemand
     ```

There are a couple different types of process manager (pm). On demand will prefork zero (0) processes. They will only forked when needed. I chose this for lots of small sites. You may want a model that suits your setup better.

1. Now lets create a vhost. Given a site named “example.com” owned by user “luser”, here’s my template:

```
<VirtualHost *:80>
ServerName        www.example.com
DocumentRoot    /var/www/luser/example.com/htdocs
SuexecUserGroup    wwwuser1 wwwclient1
ServerAlias        example.com
ErrorLog        /var/www/luser/example.com/logs/example.com.error_log
CustomLog        /var/www/luser/example.com/logs/example.com.access_log combined

<Directory /var/www/example.com/htdocs">
    Order allow,deny
    Allow from all
    Options +Indexes +FollowSymLinks +ExecCGI +Includes +MultiViews
    AllowOverride All
</Directory>

FastCgiExternalServer /tmp/fpm-example.com -socket /tmp/php-fpm-example.com.sock -user luser -
group luser
Alias /fcgi-bin /tmp/fpm-example.com
<Location /fcgi-bin>
    Options +ExecCGI
    Order allow,deny
    Allow from all
</Location>

<LocationMatch "/(ping|fpm-status)">
    SetHandler php5-fcgi-virt
    Action php5-fcgi-virt /fcgi-bin virtual
</LocationMatch>
</VirtualHost>
```

1. And create a complimentary the FPM pool config in /usr/local/etc/fpm.d/example.com.conf:

```
[example.com]
user = luser
group = luser
listen = /tmp/php-fpm-example.com.sock
chroot = /home/luser
pm = ondemand
pm.max_children = 50
pm.status_path = /fpm-status
php_admin_value[doc_root] = /example.com/htdocs
php_admin_value[cgi.fix_pathinfo] = 0
php_admin_value[sendmail_path] = /bin/mini_sendmail -t
```

- ~~**Install PHP FastCGI as a PHP replacement**~~
  ~~Install www/mod_fcgid from ports. In httpd.conf use:~~

```
cd /usr/ports/www/mod_fcgid
make install clean
```

1. ~~Now replace the PHP module used by Apache with the FastCGI module:~~

```
LoadModule fcgid_module libexec/apache22/mod_fcgid.so

<IfModule mod_fcgid.c>
  AddHandler fcgid-script .fcgi
</IfModule>
```

~~If all went well you should be able to restart Apache and be in business. A phpinfo(); should execute and provide details. Any problems the quickest way to check PHP is to just execute php -v at a shell prompt. If it doesn't segfault it will print out a short descriptive output text.~~

~~I believe this is better than the usual script based approach you will locate on the web. It starts/spawns PHP as a long running process when Apache starts instead of starting a new CGI each time PHP script is executed. The mod_fcgid is configurable.~~

1. **Install the Phusion Passenger module**

```
cd /usr/ports/www/rubygem-passenger
make install clean
passenger-install-apache2-module
```

   - Then add the module in /usr/local/etc/apache22/httpd.conf:

```
LoadModule passenger_module /usr/ports/www/rubygem-passenger/work/passenger-4.0.41/buildou
t/apache2/mod_passenger.so
<IfModule mod_passenger.c>
  PassengerRoot /usr/ports/www/rubygem-passenger/work/passenger-4.0.41
  PassengerDefaultRuby /usr/local/bin/ruby19
</IfModule>
```

   - And now ruby web applications can be used by using the following template:

```
<VirtualHost *:80>
   ServerName www.yourhost.com
   # !!! Be sure to point DocumentRoot to 'public'!
   DocumentRoot /somewhere/public
   <Directory /somewhere/public>
      # This relaxes Apache security settings.
      AllowOverride all
      # MultiViews must be turned off.
      Options -MultiViews
   </Directory>
</VirtualHost>
```

1. **Install the Perl module**

```
cd /usr/ports/www/mod_perl2
make install clean
```

   - Add the following to /usr/local/apache22/httpd.conf:

```
LoadModule perl_module /usr/local/libexec/apache22/mod_perl.so
```

Resources

- http://harold.internal.org/tag/freebsd/
- http://fosskb.wordpress.com/2014/04/12/famp-installing-apache2-4-mariadb-php-on-freebsd-10/
- http://www5.us.freebsd.org/doc/handbook/network-apache.html
- http://httpd.apache.org/mod_fcgid/mod/mod_fcgid.html
- http://httpd.apache.org/mod_fcgid/mod/mod_fcgid.html

Related issues:

Copied to FreeBSD Administration - Support #432: Install A FreeBSD, Apache 2....	**Closed**	**05/02/2014**

History

#1 - 05/02/2014 10:57 PM - Daniel Curtis

- Description updated

- % Done changed from 60 to 80

#2 - 05/03/2014 01:08 PM - Daniel Curtis

- Description updated

#3 - 05/03/2014 02:02 PM - Daniel Curtis

- Description updated

#4 - 05/03/2014 03:06 PM - Daniel Curtis

- Description updated

- % Done changed from 80 to 90

#5 - 05/14/2014 10:32 AM - Daniel Curtis

- Description updated

#6 - 05/14/2014 12:09 PM - Daniel Curtis

- Description updated

#7 - 05/19/2014 03:44 PM - Daniel Curtis

- Description updated

#8 - 05/29/2014 02:57 PM - Daniel Curtis

- Description updated

#9 - 08/09/2014 12:19 PM - Daniel Curtis

- Subject changed from Setting Up A FreeBSD Apache MariaDB PHP (FAMP) Server to Setting Up A FreeBSD, Apache 2.2, MariaDB 5.5, PHP 5.4 (FAMP) Server

- Status changed from Resolved to Closed

- % Done changed from 90 to 100

FreeBSD has changed the default apache version to 2.4.

This issue will be kept for posterity.

#10 - 08/09/2014 12:23 PM - Daniel Curtis

- Project changed from 91 to FreeBSD Administration

#11 - 08/09/2014 01:08 PM - Daniel Curtis

- Copied to Support #432: Install A FreeBSD, Apache 2.4, MariaDB 5.5, PHP 5 (FAMP) Server added

#12 - 02/14/2015 10:35 AM - Daniel Curtis

- Target version set to FreeBSD 9

#13 - 02/14/2015 11:29 AM - Daniel Curtis

- Category set to Web Server