

# FAMP

Welcome to FreeBSD!

1. Boot FreeBSD [default]
2. Boot FreeBSD with ACPI disabled
3. Boot FreeBSD in Safe Mode
4. Boot FreeBSD in single user mode
5. Boot FreeBSD with verbose logging
6. Escape to loader prompt
7. Reboot

Select option, [Enter] for default  
or [Space] to pause timer 8



Apache2.4, MySQL 5.7 or 8.0, PHP 7.4, 8.0 or 8.1 on FreeBSD 12.3, 12.4, 13.0 and 13.1 (FAMP)

December 2, 2022 femtopc.com

**Synopsys:** Most Web content is driven by an Apache Web Server with PHP language and MySQL database server. This document is a supplement to FemtoPC's "FAMP" installation script file for FreeBSD 12.3, 12.4, 13.0 and 13.1. Make sure of your CMS requirements and choose the correct MySQL and PHP versions.

[https://www.femtopc.com/shell\\_scripts/FAMP/Apache24-MySQL-PHP/](https://www.femtopc.com/shell_scripts/FAMP/Apache24-MySQL-PHP/)

#### Requirements:

1. FreeBSD server, on discrete hardware or running in a virtual machine.
2. Script designed for server with clean FreeBSD install, no other software installed.
3. Disk requirement: After installation, only 3.1GB are used, including the vanilla FreeBSD installation.
4. Must be run with 'root' access, or as a user that su to root.
5. Server must have Internet access.
6. Script file must run in /root directory.
7. Depending on Internet speed and server processing power, installation is complete in under 10 minutes.

#### Step-by-step:

1. Login to FreeBSD server as root, or su to root.

2. Fetch the FAMP package

```
fetch https://www.femtopc.com/shell_scripts/FAMP/Apache24-MySQL-PHP/install.gz
```

3. Unzip the download

```
Tar -xvf install.gz
```

4. Run the shell program and answer when prompted.

```
./famp.sh
```

5. Script will check to make sure you are running as root, in the /root directory and have Internet access. It will then check for any Apache, MySQL and PHP packages and exit if found. The install versions of MySQL and PHP will be requested. After packages are installed, the configuration will be automatically done. The php.ini can choose either the 'production' source or the 'developer' source before the file is modified. A password will be requested for MySQL root user. Please make it secure. View the logfile.log to troubleshoot any errors.

Here is a breakdown of the comments and commands that are run in the script.

```
# install Apache2.4
```

```
pkg install -y apache24
```

```
# install MySQL80-Server or MySQL57
```

```
pkg install -y mysql80-server ; pkg install -y mysql57-server
```

```
# install PHP81, PHP80 or PHP74
```

```
pkg install -y php81 ; pkg install -y php80 ; pkg install -y php74
```

```
# install PHP81-extensions, PHP80-extensions or PHP74-extensions
```

```
# Be sure to match the extensions version with the main PHP version, they cannot mix
```

```
pkg install -y php81-extensions ; pkg install -y php80-extensions ; pkg install -y php74-extensions
```

```
# install extra extensions for most content management where xx = 74, 80 or 81
```

```
# Choose the extensions that are required by the target CMS (Content Management System)
```

```
pkg install -y phpxx-bz2
```

```
pkg install -y phpxx-curl
```

```
pkg install -y phpxx-exif
```

```
pkg install -y phpxx-fileinfo
```

```
pkg install -y phpxx-gd
```

```
pkg install -y phpxx-gettext
```

```
pkg install -y phpxx-intl
```

```
pkg install -y phpxx-json
```

```
pkg install -y phpxx-mbstring
```

```
pkg install -y phpxx-mysqli
```

```
pkg install -y php74-openssl (already install in php80-extensions)
```

```
pkg install -y phpxx-pdo_mysql
```

```
pkg install -y phpxx-sockets
```

```
pkg install -y phpxx-sodium
```

```
pkg install -y phpxx-tidy
```

```
pkg install -y phpxx-zip
```

```
pkg install -y phpxx-zlib
```

```
# install Apache mod_php81, mod_php80 or mod_php74
```

```
pkg install -y mod_php81 ; pkg install -y mod_php80 ; pkg install -y mod_php74
```

```
# These are what the configuration routines do
# modify the /usr/local/etc/apache24/httpd.conf file
# make a backup and then edit
cd /usr/local/etc/apache24
cp httpd.conf httpd.conf.original
ee httpd.conf

# set the ServerAdmin, in the script it is set to root. You may also set a webmaster email here.
# set the ServerName to the IP address of server. A domain name could be set here if available.
# set the second time AllowOverride None, where None is capital "N" to All. This will allow .htaccess files
to work.
# set DirectoryIndex from just index.html to index.php index.htm index.html to capture PHP files
# setup Apache to run PHP by creating an Includes/php.conf file
cd /usr/local/etc/apache24/Includes
ee php.conf

<FilesMatch "\.php$" >
    SetHandler application/x-httpd-php
</FilesMatch>
<FilesMatch "\.phps$" >
    SetHandler application/x-httpd-php-source
</FilesMatch># return to our root directory

cd /root

# Make Apache start automatically when FreeBSD boots
sysrc apache24_enable="yes"

# Startup the Apache Server
service apache24 start

# Setup to run MySQL server
# Make Server start automatically when FreeBSD boots.
sysrc mysql_enable="yes"

# start the mysql server for the first time
service mysql-server start

# MySQL81 and MySQL 80 have no password set for root (MySQL57 has a password stored in
.mysql_secret)
```

```
# Set your new password; it MUST BE in single quotes
# If MySQL57 is installed, use the password from the .mysql_secret file when prompted
/usr/local/bin/mysqladmin -u root password 'your new password'
# For MySQL 80 or 81, just press Enter when prompted for a password, as none was assigned

# Now configure PHP
cd /usr/local/etc

# choose a development or a production environment
# development (displays errors in web pages)
cp php.ini-development php.ini
# or production (will not display web page errors)
cp php.ini-production php.ini
# edit the php.ini file
ee php.ini

# find where memory_limit = 128M and increase to 1024M
# find where post_max_size = 8M and set to 32M (or set your limit here)
# find where upload_max_filesize = 2M and set to 1G (or your limit here)
# find where max_file_uploads = 20 and increase to 80
# lastly, find ;date.timezone and uncomment (delete ;) and enter timezone (IMPORTANT!!)
# hint: find your server's setting by doing "cat /var/db/zoneinfo"
# after changes to php.ini, Apache must be restarted
service apache24 restart

# at this point you have a working FAMP server
cd /root

# This completes the FAMP server install and setup. View the default web page at:
http://your_server_ip_address/ or check PHP at http://your_ip_address/phpinfo.php
<end>
```