

Ранее я уже рассказывал, как установить и настроить FreeBSD 10.1 в качестве web сервера ( 1 часть, 2 часть, 3 часть ) Там мы использовали Apache, nginx и php. Здесь же я хочу рассказать об установке и настройке связки nginx+php-fpm. По общепризнанному мнению, такая комбинация работает быстрее, чем связка с apache. Я на своем опыте в этом убедился. При прочих равных условиях, сайт wordpress выдерживает большую нагрузку с nginx и php fpm, нежели apache и php.

Если у вас есть желание научиться строить и поддерживать высокодоступные и надежные системы, рекомендую познакомиться с **онлайн-курсом «Администратор Linux»** в OTUS. Курс не для новичков, для поступления нужно пройти .

Содержание:

- 1 Установка и настройка mysql на FreeBSD
- 2 Установка и настройка PHP-FPM на FreeBSD
- 3 Установка и настройка nginx на FreeBSD
- 4 Установка phpmyadmin на web сервер FreeBSD
- 5 Дополнительные материалы по FreeBSD

Приступим. У нас имеется:

```
# uname -v  
FreeBSD 10.1-RELEASE
```

Обновим до последней версии с помощью вот этой инструкции. Проверяем, что получилось:

```
# uname -v
```

```
FreeBSD 10.1-RELEASE-p8
```

Отлично, обновления прошли. Я не буду давать общие рекомендации по настройке сервера, я их уже давал во 2-й части инструкции, ссылки на которую я приводил выше. Можно посмотреть при желании. Приступим сразу к делу.

## Установка и настройка mysql на Freebsd

Устанавливаем mysql:

```
# cd /usr/ports/databases/mysql55-server  
# make config-recursive  
# make install clean
```

Добавляем Mysql в автозагрузку:

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

Создаем для него конфигурационный файл:

```
# touch /etc/my.cnf
```

Конфигурацию каждый под свои потребности может сделать, описаний настроек в интернете много. Я предлагаю свой универсальный вариант:

```
[client]  
port = 3306
```

```
socket = /tmp/mysql.sock

[mysqld]
port = 3306
bind-address = 127.0.0.1
socket = /tmp/mysql.sock
skip-external-locking
key_buffer_size = 256M
max_allowed_packet = 1M
table_open_cache = 256
sort_buffer_size = 1M
read_buffer_size = 1M
read_rnd_buffer_size = 4M
myisam_sort_buffer_size = 64M
thread_cache_size = 8
query_cache_size= 16M
thread_concurrency = 8
log-bin=mysql-bin
log=/var/log/mysql.log
binlog_format=mixed
server-id = 1

[mysqldump]
quick
max_allowed_packet = 16M

[mysql]
no-auto-rehash

[myisamchk]
key_buffer_size = 128M
sort_buffer_size = 128M
read_buffer = 2M
```

```
write_buffer = 2M

[mysqlhotcopy]
interactive-timeout
```

В моем варианте сервер жестко привязан к локальному интерфейсу и удаленно с ним работать нельзя. Если вам нужны удаленные подключения, то в параметре bind-address добавьте внешний адрес сервера.

Создадим файл логов:

```
touch /var/log/mysql.log
chown mysql:mysql /var/log/mysql.log
```

Запускаем Mysql:

```
# /usr/local/etc/rc.d/mysql-server start
```

Проверяем, все ли в порядке:

```
# sockstat | grep mysql

mysql mysqld 46460 13 tcp4 127.0.0.1:3306 *:*
mysql mysqld 46460 15 stream /tmp/mysql.sock
```

Демон запущен и слушает локальный интерфейс.

Теперь запустим скрипт для настройки безопасности:

```
# /usr/local/bin/mysql_secure_installation
```

Задаем пароль для root и далее на все вопросы отвечаем утвердительно:

```
Setting the root password ensures that nobody can log into the MySQL
root user without the proper authorisation.

Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MySQL installation has an anonymous user, allowing anyone
to log into MySQL without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
... Success!

By default, MySQL comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MySQL
installation should now be secure.

Thanks for using MySQL!
```

На этом установка и настройка mysql закончена. Идем дальше.

## Установка и настройка PHP-FPM на FreeBSD

Начиная с версии php 5.3 php-fpm идет как модуль, поэтому просто начинаем установку php и указываем в опциях fpm:

```
# cd /usr/ports/lang/php5  
# make config-recursive
```





```
php5-5.4.39
lqooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo x
x+[x] CLI Build CLI version x
x+[x] CGI Build CGI version x
x+[x] FPM Build FPM version x
x+[ ] EMBED Build embedded library x
x+[ ] DEBUG Enable debug x
x+[ ] DTRACE Enable DTrace support x
x+[x] IPV6 Enable ipv6 support x
x+[ ] MAILHEAD Enable mail header patch x
x+[x] LINKTHR Link thread lib (for threaded extensions) x
x+[ ] ZTS Force Zend Thread Safety (ZTS) build x
m x
< OK > <Cancel> x
ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo j
```

```
# make install clean
```

Ставим теперь дополнения php-extensions:

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

Дополнений много, все ставить не обязательно, можно будет остальное доставить потом в случае необходимости. Я обычно выбираю наиболее популярные и необходимые:

- BZZ, STYPE, CURL, DOM, EXIF, FILTER, FTP, GD, GETTEXT, HASH, ICONV, JSON, MBSTRING, MCRYPT, MYSQL, MYSQLI, PDF, PDO, PDO\_SQLITE, PHAR, POSIX, SESSION, SIMPLEXML, SQLITE3, YOKENIZER, XML, XMLREADER, XMLWRITER, ZIP

```
# make install clean
```

Добавляем php-fpm в автозагрузку:

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

Редактируем файл с настройками, приводим его к следующему виду:

```
# mcedit /usr/local/etc/php-fpm.conf

[global]
pid = run/php-fpm.pid

[www]
user = www
group = www

# работа через unix сокет, а не tcp, дает небольшой прирост производительности
listen = /tmp/php-fpm.sock
# права доступа к сокету
listen.owner = www
listen.group = www
listen.mode = 0660

# как будут создаваться новые рабочие процессы
pm = dynamic
# максимальное количество рабочих процессов
pm.max_children = 10
# число запущенных процессов при старте сервера
```

```
pm.start_servers = 2
# минимальное и максимальное количество процессов в простое
pm.min_spare_servers = 1
pm.max_spare_servers = 3
```

Проверяем синтаксис настроек:

```
# php-fpm -t

NOTICE: configuration file /usr/local/etc/php-fpm.conf test is successful
```

Стартуем php-fpm:

```
# /usr/local/etc/rc.d/php-fpm start
```

Проверяем запустился ли:

```
# sockstat | grep php-fpm

www php-fpm 504 0 stream /tmp/php-fpm.sock
www php-fpm 503 0 stream /tmp/php-fpm.sock
root php-fpm 502 4 stream -> ??
root php-fpm 502 6 stream -> ??
root php-fpm 502 7 stream /tmp/php-fpm.sock
```

Все в порядке. Идем дальше.

## Установка и настройка nginx на FreeBSD

Устанавливаем nginx на наш FreeBSD сервер:

```
# cd /usr/ports/www/nginx  
# make config-recursive
```

Выбираем настройки по-умолчанию.

```
# make install clean
```

Добавляем nginx в автозагрузку:

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

Редактируем файл настроек. Я приведу пример с двумя виртуальными серверами, на одном из которых будут настройки для wordpress, другой будет для phptoadmin, я люблю его вешать на отдельный поддомен.

```
# mcedit /usr/local/etc/nginx/nginx.conf  
  
user www;  
worker_processes 1;  
  
pid /var/run/nginx.pid;  
error_log /var/log/nginx-error.log warn;
```

```
events {
worker_connections 1024;
use kqueue;
}

http {
include mime.types;
default_type application/octet-stream;

log_format main '$remote_addr - $remote_user [$time_local] '
'$request' $status $bytes_sent '
'$http_referer' '$http_user_agent' '
'$gzip_ratio'';

sendfile on;
tcp_nopush on;
tcp_nodelay on;
keepalive_timeout 65;
server_tokens off;

fastcgi_intercept_errors on;
client_max_body_size 50m;

server {
listen 80;
server_name www.site1.ru;
rewrite ^ http://site1.ru$request_uri? permanent;
}

# wordpress
server {
listen 80;
server_name site1.ru;
```

```
root /web/sites/site1.ru/www/;
index index.php index.html index.htm;
access_log /web/sites/site1.ru/log/access.log main;
error_log /web/sites/site1.ru/log/error.log;

location / {
try_files $uri $uri/ /index.php?q=$uri&$args;
}
location ~* ^.+.(js|css|png|jpg|jpeg|gif|ico)$ {
access_log off;
expires max;
}
location ~ \.php$ {
fastcgi_pass unix:/tmp/php-fpm.sock;
fastcgi_index index.php;

fastcgi_param DOCUMENT_ROOT /web/sites/site1.ru/www/;
fastcgi_param SCRIPT_FILENAME /web/sites/site1.ru/www$fastcgi_script_name;
fastcgi_param PATH_TRANSLATED /web/sites/site1.ru/www$fastcgi_script_name;

include fastcgi_params;
fastcgi_param QUERY_STRING $query_string;
fastcgi_param REQUEST_METHOD $request_method;
fastcgi_param CONTENT_TYPE $content_type;
fastcgi_param CONTENT_LENGTH $content_length;
fastcgi_intercept_errors on;
fastcgi_ignore_client_abort off;
fastcgi_connect_timeout 60;
fastcgi_send_timeout 180;
fastcgi_read_timeout 180;
fastcgi_buffer_size 128k;
fastcgi_buffers 4 256k;
fastcgi_busy_buffers_size 256k;
```

```
fastcgi_temp_file_write_size 256k;
}
location = /favicon.ico {
log_not_found off;
access_log off;
}
location = /robots.txt {
allow all;
log_not_found off;
access_log off;
}
location ~ /\.ht {
deny all;
}
}
# phpmyadmin
server {
listen 80;
server_name pma.site1.ru;
root /web/sites/pma.site1.ru/www/;
index index.php index.html index.htm;
access_log /web/sites/pma.site1.ru/log/access.log main;
error_log /web/sites/pma.site1.ru/log/error.log;

location ~ /\.php$ {
fastcgi_pass unix:/tmp/php-fpm.sock;
fastcgi_index index.php;

fastcgi_param DOCUMENT_ROOT /web/sites/pma.site1.ru/www/;
fastcgi_param SCRIPT_FILENAME /web/sites/pma.site1.ru/www$fastcgi_script_name;
fastcgi_param PATH_TRANSLATED /web/sites/pma.site1.ru/www$fastcgi_script_name;

include fastcgi_params;
```

```
fastcgi_param QUERY_STRING $query_string;
fastcgi_param REQUEST_METHOD $request_method;
fastcgi_param CONTENT_TYPE $content_type;
fastcgi_param CONTENT_LENGTH $content_length;
fastcgi_intercept_errors on;
fastcgi_ignore_client_abort off;
fastcgi_connect_timeout 60;
fastcgi_send_timeout 180;
fastcgi_read_timeout 180;
fastcgi_buffer_size 128k;
fastcgi_buffers 4 256k;
fastcgi_busy_buffers_size 256k;
fastcgi_temp_file_write_size 256k;
}
}
}
```

Создаем необходимые каталоги:

```
# mkdir /web && mkdir /web/sites && mkdir /web/sites/sitel.ru && mkdir /web/sites/sitel.ru/www && mkdir
/web/sites/sitel.ru/log && mkdir /web/sites/pma.sitel.ru && mkdir /web/sites/pma.sitel.ru/www && mkdir
/web/sites/pma.sitel.ru/log
# chown -R www:www /web
```

Проверяем конфигурацию:

```
# rehash
# nginx -t
```



```
nginx: the configuration file /usr/local/etc/nginx/nginx.conf syntax is ok
nginx: [emerg] mkdir() "/var/tmp/nginx/client_body_temp" failed (2: No such file or directory)
nginx: configuration file /usr/local/etc/nginx/nginx.conf test failed
```

Почему-то nginx не смог создать папку во временном каталоге. Поможем ему сами:

```
# mkdir /var/tmp/nginx
# nginx -t

nginx: the configuration file /usr/local/etc/nginx/nginx.conf syntax is ok
nginx: configuration file /usr/local/etc/nginx/nginx.conf test is successful
```

Все в порядке. Наш web сервер на FreeBSD готов. Запускаем nginx и проверяем, нормально ли запустился:

```
# /usr/local/etc/rc.d/nginx start
# sockstat | grep nginx

www      nginx      3488  10 tcp4   *:80          *:*
www      nginx      3488  11 stream -> ??
root     nginx      3487   3 stream -> ??
root     nginx      3487  10 tcp4   *:80          *:*
root     nginx      3487  11 stream -> ??
```

Все в порядке, сервер запущен и слушает 80-й порт. Теперь проверим работу web сервера с помощью тестовой странички. Создаем ее:

```
# touch /web/sites/site1.ru/www/index.php
```

```
# mcedit /web/sites/site1.ru/www/index.php
```

```
/web/sites/site1.ru/www/index.php  
<?phpinfo() ?>
```

Переходим на сайт по адресу <http://site1.ru> Не забудьте настроить DNS для доступа к установленному серверу, либо отредактируйте в windows файл hosts, добавив туда строку:

```
192.168.1.21 site1.ru
```

где 192.168.1.21 адрес web сервера Freebsd 10.

Вы должны увидеть картинку:





<b>System</b>	FreeBSD hyperv-freebsd 10.1-RELEASE-p8 FreeBSD 10.1-RELEASE-p8 #0 r281084M: Sun Apr 5 00:59:08 MSK 2015 root@hyperv-freebsd:/usr/obj/usr/src/sys/GENERIC amd64
<b>Build Date</b>	Apr 9 2015 15:24:00
<b>Configure Command</b>	'./configure' '--with-layout=GNU' '--localstatedir=/var' '--with-config-file-scan-dir=/usr/local/etc/php' '--disable-all' '--enable-libxml' '--enable-mysqlnd' '--with-libxml-dir=/usr/local' '--with-pcre-regex=/usr/local' '--with-zlib-dir=/usr' '--program-prefix=' '--enable-fpm' '--with-fpm-user=www' '--with-fpm-group=www' '--with-regex=php' '--with-zend-vm=CALL' '--prefix=/usr/local' '--mandir=/usr/local/man' '--infodir=/usr/local/info' '--build=amd64-portbld-freebsd10.1'
<b>Server API</b>	FPM/FastCGI
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/usr/local/etc
<b>Loaded Configuration File</b>	(none)
<b>Scan this dir for additional .ini files</b>	/usr/local/etc/php
<b>Additional .ini files parsed</b>	/usr/local/etc/php/extensions.ini
<b>PHP API</b>	20100412

Если видите ее у себя, поздравляю, вы все настроили правильно и web сервер полностью готов к работе. Остался только маленький штрих. Для удобства управления mysql установим phpmysqladmin.

## Установка phpmyadmin на web сервер FreeBSD

```
# cd /usr/ports/databases/phpmyadmin  
# make install clean
```

Перемещаем файлы из установочной директории по-умолчанию в наш поддомен pma.site1.ru:

```
# mv /usr/local/www/phpMyAdmin/* /web/sites/pma.site1.ru/www  
# mv /usr/local/www/phpMyAdmin/.*/ /web/sites/pma.site1.ru/www
```

Меняем владельца:

```
# chown -R www:www /web/sites/pma.site1.ru/www
```

Настраиваем DNS записи на pma.site1.ru или просто добавляем в винде в файл hosts строку

```
192.168.1.21 pma.site1.ru
```

И заходим по адресу <http://pma.site1.ru>



## Добро пожаловать в phpMyAdmin

Язык - *Language*

Русский - Russian ▼

Авторизация 

Пользователь:

Пароль:

Вперёд

У вас должно открыться окно приветствия phpmyadmin. Вводим сюда пользователя и пароль root, которые мы задали в самом начале настройки и попадаем в панель управления базами.

На этом настройка web сервера на базе Freebsd 10.1 и связки nginx + php-fpm закончена.

Помогла статья? Подписывайся на telegram канал автора

Анонсы всех статей, плюс много другой полезной и интересной информации, которая не попадает на сайт.

## Дополнительные материалы по Freebsd

---

### Онлайн курс по Linux

Если у вас есть желание научиться строить и поддерживать высокодоступные и надежные системы, рекомендую познакомиться с **онлайн-курсом «Администратор Linux»** в OTUS. Курс не для новичков, для поступления нужны базовые знания по сетям и установке Linux на виртуалку. Обучение длится 5 месяцев, после чего успешные выпускники курса смогут пройти собеседования у партнеров. Что даст вам этот курс:

- Знание архитектуры Linux.
- Освоение современных методов и инструментов анализа и обработки данных.
- Умение подбирать конфигурацию под необходимые задачи, управлять процессами и обеспечивать безопасность системы.
- Владение основными рабочими инструментами системного администратора.
- Понимание особенностей развертывания, настройки и обслуживания сетей, построенных на базе Linux.
- Способность быстро решать возникающие проблемы и обеспечивать стабильную и бесперебойную работу системы.

Проверьте себя на вступительном тесте и смотрите подробнее программу по .

### Рекомендую полезные материалы по FreeBSD:

- Установка
- Настройка
- Обновление
- Шлюз
- Прокси сервер
- Веб сервер NGINX
- Веб сервер Apache

Описание установки FreeBSD 11 на одиночный диск, либо на софтовый raid1, сделанный средствами zfs, которые поддерживает стандартный установщик. Базовая настройка FreeBSD, которую можно выполнить после установки сервера общего назначения. Представлены некоторые рекомендации по повышению удобства пользования и безопасности.

Описание и нюансы обновления системы FreeBSD с помощью утилиты freebsd-update. Показано пошагово на конкретном примере обновления.

Настройка FreeBSD шлюза для обеспечения выхода в интернет. Используется ipfw и ядерный nat, dnsmasq в качестве dhcp и dns сервера. Мониторинг сетевой активности с помощью iftop.

Подробная настройка на FreeBSD прокси сервера squid + sams2 - панели управления для удобного администрирования.

Настройка максимально быстрого web сервера на базе FreeBSD и nginx + php-fpm. Существенный прирост производительности по сравнению с классическим apache.

Настройка web сервера на FreeBSD в связке с apache, nginx, php и mysql. Пошаговая установка и настройка каждого компонента.