

Welcome to **nginx** on Fedora!

This page is used to test the proper operation of the **nginx** HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly.

Website Administrator

This is the default `index.html` page that is distributed with **nginx** on Fedora. It is located in `/usr/share/nginx/html`.

You should now put your content in a location of your choice and edit the `root` configuration directive in the **nginx** configuration file `/etc/nginx/nginx.conf`.



Для организации web сервера установим на CentOS 7 связку nginx + php-fpm. Подразумевается, что у вас уже есть настроенный сервер CentOS 7, если нет, то рекомендую статью на эту тему — CentOS 7 настройка сервера. Связка nginx и php fpm дает большую производительность, нежели apache с php.

Если у вас есть желание научиться **профессионально** строить и поддерживать высокодоступные виртуальные и кластерные среды, рекомендую познакомиться с онлайн-курсом **Администратор Linux. Виртуализация и кластеризация**. в OTUS. Курс не для новичков, для поступления нужно пройти .

Содержание:

- 1 Установка nginx на CentOS 7
- 2 Установка mysql на CentOS 7
- 3 Установка php-fpm на CentOS 7

- 4 Настройка nginx и php-fpm на CentOS 7
- 5 Установка phpmyadmin на CentOS 7

Данная статья использует в настройке компоненты только из официальных репозиториях CentOS 7. Это максимально стабильный вариант настройки. Так как это консервативная система, очень многие программы в ней старых версий. Если вам нужен самый свежий софт, рекомендую новую статью по такой же теме — установка и настройка nginx php-fpm7.

Установка nginx на CentOS 7

Ставим непосредственно web сервер:

```
# yum install -y nginx
```

Добавляем в автозагрузку:

```
# systemctl enable nginx.service
```

И запускаем:

```
# systemctl start nginx.service
```

Проверяем, запустился ли:

```
# netstat -tulpn | grep nginx
```

```
tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN 21585/nginx: master
```

Теперь можно зайти на тестовую страничку, чтобы убедиться, что nginx работает:

```
http://ip_сервера
```

Не забудьте открыть 80-й порт на фаерволе.

Welcome to **nginx** on Fedora!

This page is used to test the proper operation of the **nginx** HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly.

Website Administrator

This is the default `index.html` page that is distributed with **nginx** on Fedora. It is located in `/usr/share/nginx/html`.

You should now put your content in a location of your choice and edit the `root` configuration directive in the **nginx** configuration file `/etc/nginx/nginx.conf`.



Установка mysql на CentOS 7

Практически ни один web сервер сейчас не обходится без mysql. В centos 7 вместо mysql устанавливается mariadb, которая имеет полную совместимость с mysql, в том числе понимает все ее команды. Так что ставим на сервер:

```
# yum install -y mariadb mariadb-server net-tools
```

Добавляем в автозапуск:

```
# systemctl enable mariadb.service  
ln -s '/usr/lib/systemd/system/mariadb.service' '/etc/systemd/system/multi-user.target.wants/mariadb.service'
```

И запускаем:

```
# systemctl start mariadb.service
```

Проверяем, запустилась ли mysql:

```
# netstat -tulpn | grep mysql  
tcp 0 0 0.0.0.0:3306 0.0.0.0:* LISTEN 21488/mysql
```

Все в порядке.

Теперь воспользуемся скриптом безопасности, который позволит задать пароль root и произвести некоторые настройки:

```
# /usr/bin/mysql_secure_installation  
/usr/bin/mysql_secure_installation: line 379: find_mysql_client: command not found
```

```
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB  
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!
```

```
In order to log into MariaDB to secure it, we'll need the current  
password for the root user. If you've just installed MariaDB, and  
you haven't set the root password yet, the password will be blank,  
so you should just press enter here.
```

```
Enter current password for root (enter for none):  
OK, successfully used password, moving on...
```

```
Setting the root password ensures that nobody can log into the MariaDB  
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 MariaDB installation has an anonymous user, allowing anyone to log into MariaDB 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, MariaDB 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 MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

Установка php-fpm на CentOS 7

Устанавливаем php-fpm и некоторые модули, которые скорее всего пригодятся в дальнейшей работе:

```
# yum install -y php-fpm php-cli php-mysql php-gd php-ldap php-odbc php-pdo php-pecl-memcache php-pear php-xml php-xmlrpc
php-mbstring php-snmp php-soap
```

Так же я предлагаю установить **APC** — бесплатный и открытый кэшер кода PHP, который кэширует и оптимизирует промежуточный код PHP, ускоряя его исполнение. Установим APC из хранилища PHPpecl. Для того, чтобы скомпилировать пакет APC нужно установить Centos Development tools.

```
# yum install -y php-devel
# yum groupinstall 'Development Tools'
```

Компилируем APC:

```
# pecl install apc
```


На все вопросы просто жмем enter, выбирая параметры по-умолчанию.

Теперь отредактируем /etc/php.ini как нам рекомендует инсталлятор:

```
# mcedit /etc/php.ini
```

Задаем параметр **cgi.fix_pathinfo=0** и добавляем в конце строку **extension=apc.so**
Заодно установим сразу же часовой пояс: **date.timezone = «Europe/Moscow»**

Создадим файл настроек APC:

```
# mcedit /etc/php.d/apc.ini
extension = apc.so
apc.enabled = 1
apc.cache_by_default = 1
apc.ttl = 7200
apc.shm_size = 64M
apc.max_file_size = 4M
apc.stat = 1
```

Добавляем php-fpm в автозагрузку:

```
# systemctl enable php-fpm.service
ln -s '/usr/lib/systemd/system/php-fpm.service' '/etc/systemd/system/multi-user.target.wants/php-fpm.service'
```

И запускаем:

```
# systemctl start php-fpm.service
```

Проверяем:

```
# netstat -tulpn | grep php-fpm  
tcp 0 0 127.0.0.1:9000 0.0.0.0:* LISTEN 2907/php-fpm: maste
```

Настройка nginx и php-fpm на CentOS 7

Теперь отредактируем конфиги для совместной работы. Приводим конфиг nginx к следующему виду:

```
# mcedit /etc/nginx/nginx.conf  
  
user nginx;  
worker_processes 1;  
  
pid /var/run/nginx.pid;  
error_log /var/log/nginx-error.log warn;  
  
events {  
worker_connections 1024;  
}  
  
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.sitel.ru;
rewrite ^ http://sitel.ru$request_uri? permanent;
}

# wordpress
server {
listen 80;
server_name sitel.ru;
root /web/sites/sitel.ru/www;
index index.php index.html index.htm;
access_log /web/sites/sitel.ru/log/access.log main;
error_log /web/sites/sitel.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:/var/run/php-fpm/php5-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:/var/run/php-fpm/php5-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;  
}  
}  
}
```

worker_processes рекомендуется сделать равным количеству ядер процессора
fastcgi_pass указывает на unix сокет, по которому идет связь с php-fpm
site1.ru — адрес сайта с блогом wordpress, настройки приведены специально под него
pma.site1.ru — отдельный виртуальный домен под phpmyadmin, его мы поставим позже

Более подробно о том, как выполнить базовую настройку nginx читайте в отдельной обширной статье.

Перезапускаем nginx:

```
# systemctl restart nginx.service
```

Теперь правим конфиг php-fpm, приводим его к следующему виду:

```
# mcedit /etc/php-fpm.d/www.conf  
  
[www]  
listen = /var/run/php-fpm/php5-fpm.sock  
listen.allowed_clients = 127.0.0.1  
user = nginx  
group = nginx  
# 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
slowlog = /var/log/php-fpm/www-slow.log
php_admin_value[error_log] = /var/log/php-fpm/www-error.log
php_admin_flag[log_errors] = on
php_value[session.save_handler] = files
php_value[session.save_path] = /var/lib/php/session
```

Перезапускаем php-fpm:

```
# systemctl restart php-fpm.service
```

Создаем необходимые каталоги:

```
# mkdir /web && mkdir /web/sites && mkdir /web/sites/site1.ru && mkdir /web/sites/site1.ru/www && mkdir
/web/sites/site1.ru/log && mkdir /web/sites/pma.site1.ru && mkdir /web/sites/pma.site1.ru/www && mkdir
/web/sites/pma.site1.ru/log
# chown -R nginx:nginx /web
```

Теперь все готово, можно проверять. Создаем в папке /web/sites/site1.ru/www тестовый файл:

```
# mcedit /web/sites/site1.ru/www/index.php
<?php
phpinfo();
```

```
?>
```

Заходим по адресу: <http://site1.ru/> и если наблюдаем информацию о php, то все сделано правильно.

PHP Version 5.4.16



System	Linux zeroxzed.ru 3.10.0-229.1.2.el7.x86_64 #1 SMP Fri Mar 27 03:04:26 UTC 2015 x86_64
Build Date	Oct 31 2014 13:01:09
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/curl.ini, /etc/php.d/dom.ini, /etc/php.d/fileinfo.ini, /etc/php.d/gd.ini, /etc/php.d/json.ini, /etc/php.d/ldap.ini, /etc/php.d/mbstring.ini, /etc/php.d/memcache.ini, /etc/php.d/mysql.ini, /etc/php.d/mysqli.ini, /etc/php.d/odbc.ini, /etc/php.d/pdo.ini, /etc/php.d/pdo_mysql.ini, /etc/php.d/pdo_odbc.ini, /etc/php.d/pdo_sqlite.ini, /etc/php.d/phar.ini, /etc/php.d/posix.ini, /etc/php.d/snmp.ini, /etc/php.d/soap.ini, /etc/php.d/sqlite3.ini, /etc/php.d/sysvmsg.ini, /etc/php.d/sysvsem.ini, /etc/php.d/sysvshm.ini, /etc/php.d/wddx.ini, /etc/php.d/xmlreader.ini, /etc/php.d/xmlrpc.ini, /etc/php.d/xmlwriter.ini, /etc/php.d/xsl.ini, /etc/php.d/zip.ini

Если нет, то идем в логи и смотрим, в чем ошибка.

Установка phpmyadmin на CentOS 7

Для удобства управления базами веб сайтов я всегда использую phpmyadmin. Установим ее:

```
# yum install -y phpmyadmin
```

Копируем файлы панели в наш виртуальный домен, созданный ранее:

```
# cp -R /usr/share/phpMyAdmin/* /web/sites/pma.site1.ru/www  
# chown -R nginx:nginx /web/sites/pma.site1.ru/www
```

Заходим по адресу <http://pma.site1.ru/> и проверяем, все ли в порядке.

У меня при первом запуске в браузере открылся просто белый лист. Начал разбираться в чем дело. В логе ошибок nginx этого виртуального хоста увидел ошибку:

```
*43 FastCGI sent in stderr: "PHP message: PHP Fatal error: Call to undefined function __() in  
/web/sites/pma.site1.ru/www/libraries/core.lib.php on line 245" while reading response header from upstream, client:  
77.47.234.119, server: pma.site1.ru, request: "GET /index.php HTTP/1.1", upstream: "fastcgi://unix:/var/run/php-fpm/php5-  
fpm.sock:", host: "pma.site1.ru"
```

Немного погуглил на эту тему и нашел, в чем причина ошибки. Проблема с директорией для файлов сессий. Чтобы исправить ошибку, создаем эту директорию и выставляем на нее нужные права:

```
# cd /var/lib/php/  
# mkdir session
```

```
# chown nginx:nginx session/
```

После этого загрузилась панель phptuadmin:



Добро пожаловать в phpMyAdmin

Язык - *Language*

Русский - Russian ▼

Авторизация 

Пользователь:

Пароль:

Вперёд

Более подробную информацию об установке и настройке phpmyadmin смотрите в отдельной статье.

На этом все, настройка nginx + php-fpm на CentOS7 закончена.

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 .

Помогла статья? Подписывайся на telegram канал автора

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