Я уже писал статью по данной теме, и она формально даже не устарела, если брать все пакеты из официальных репозиториев. Сегодня я настрою производительный веб сервер на свежих версиях nginx, php-fpm, где сам php версии 7.1. Сейчас использовать версию php54, которую предлагает CentOS по-умолчанию, очень странно, поэтому я решил актуализировать статью и все настроить в соответствии с современными реалиями.

Содержание:

Введение

Ранее я рассказывал о настройке nginx и php-fpm. В принципе, статья полностью актуальна, по ней получится настроить веб сервер, если вас устраивают версии предложенных в стандартном репозитории пакетов. Если же хочется версий посвежее, то читайте далее.

Работать будем на сервере под управлением CentOS 7. Если у вас его еще нет, то читайте мои статьи на тему установки и базовой настройки centos. Не забудьте уделить внимание теме настройки iptables. В данной статье я ее не буду касаться, хотя тема важная для web сервера.

В своей тестовой среде я буду использовать следующие сущности.

hl.zeroxzed.ru	имя тестового виртуального хоста и сайта
/web/sites	директория для размещения виртуальных хостов
95.169.190.64	внешний ip адрес сервера
p1m2a.zeroxzed.ru	имя виртуального хоста для phpmyadmin

Подопытным сервером будет выступать виртуальная машина от keyweb, расположенная на сервере в Германии. Характеристики следующие:

Процессор	2 ядра
Память	8 Gb
Диск	150 Gb SSD

Указанный сервер включает в себя администрирование. То есть вы можете его заказать и прислать в техподдержку ссылку на эту статью. Они за вас все настроят. Если все же хотите настраивать сами, то берите сервер из линейки PRO. Они дешевле, но настраивать все придется самому. Чем я и займусь

Установка nginx на CentOS 7

Для установки самой свежей стабильной версии nginx на centos подключим родной репозиторий.

```
# rpm -Uvh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7.ngx.noarch.rpm
```

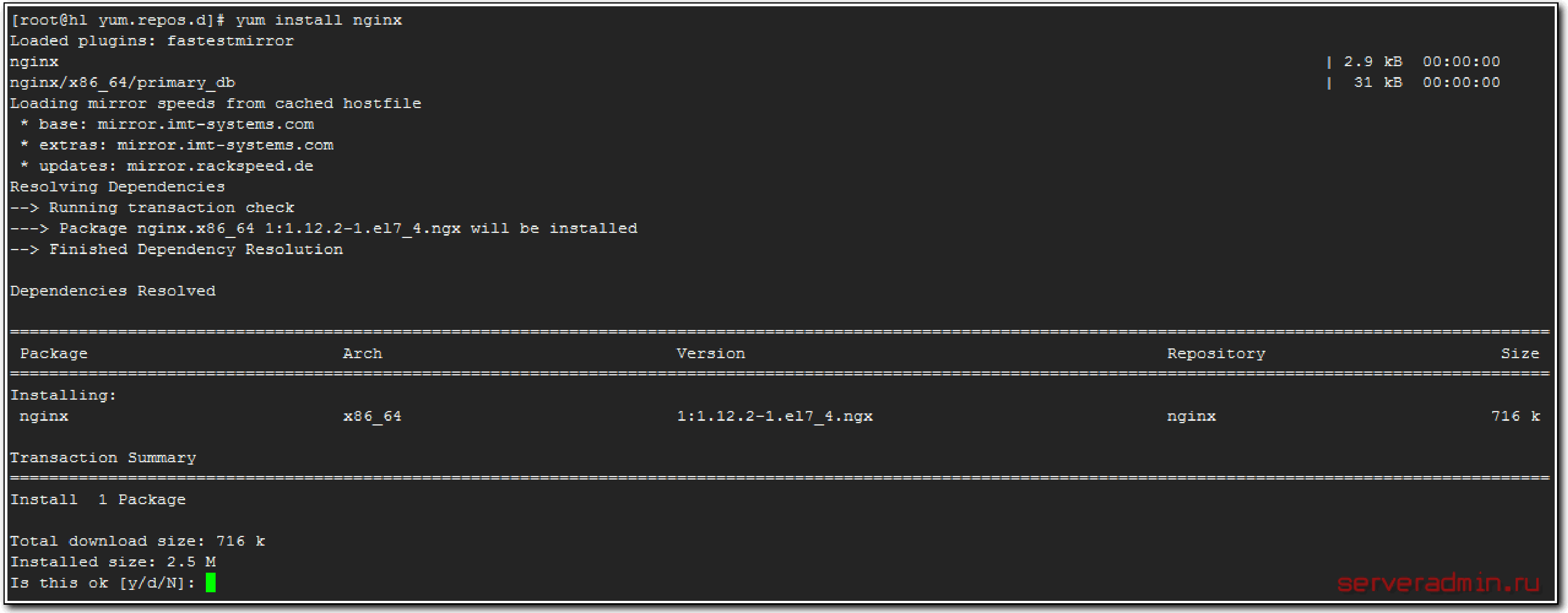
Если по какой-то причине ссылка изменится или устареет, то можно создать файл с конфигурацией репозитория nginx вручную. Для этого рисуем такой конфиг */etc/yum.repos.d/nginx.repo.*

```
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/centos/7/$basearch/
```

```
gpgcheck=0
enabled=1
```

Устанавливаем nginx на сервер.

```
# yum install nginx
```

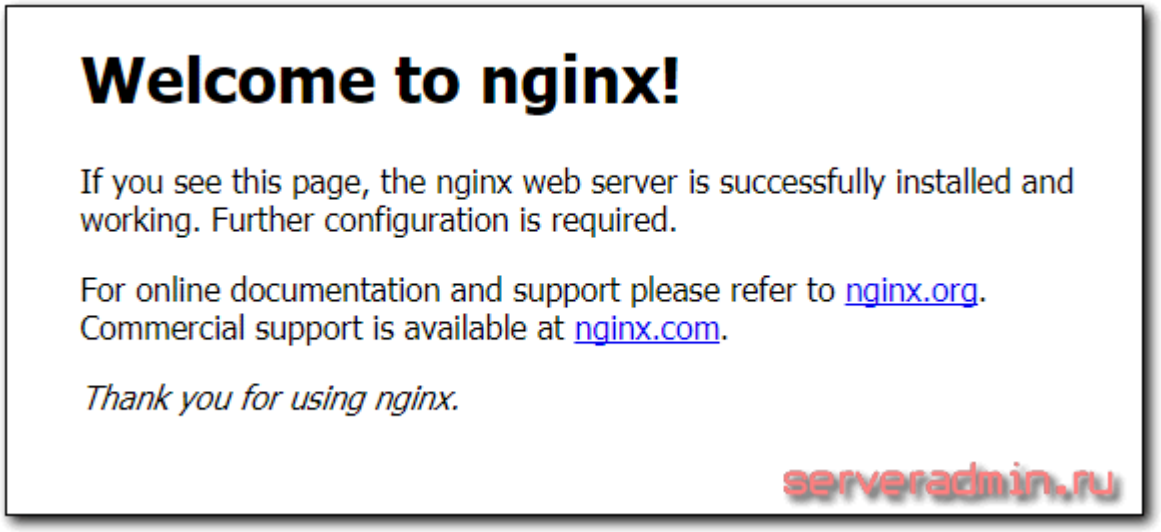


```
[root@h1 yum.repos.d]# yum install nginx
Loaded plugins: fastestmirror
nginx                                                                     | 2.9 kB  00:00:00
nginx/x86_64/primary_db                                                   |  31 kB  00:00:00
Loading mirror speeds from cached hostfile
 * base: mirror.imt-systems.com
 * extras: mirror.imt-systems.com
 * updates: mirror.rackspeed.de
Resolving Dependencies
--> Running transaction check
---> Package nginx.x86_64 1:1.12.2-1.el7_4.ngx will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================
 Package              Arch                  Version                       Repository        Size
================================================================================================
Installing:
 nginx                x86_64                1:1.12.2-1.el7_4.ngx          nginx            716 k

Transaction Summary
================================================================================================
Install  1 Package

Total download size: 716 k
Installed size: 2.5 M
Is this ok [y/d/N]:
```


Запускаем nginx и добавляем в автозагрузку.

```
# systemctl start nginx
# systemctl enable nginx
```

Проверяем, запустился ли web сервер. Для этого идем по ссылке *http://95.169.190.64/*. Вы должны увидеть стандартную страницу заглушку.



Если страница не открывается, то скорее всего вы не настроили firewall. Свою статью по его настройке я приводил в самом начале.

Настройка nginx

Расскажу, как настроить nginx для работы разных виртуальных хостов. Создадим виртуальный хост и подготовим директории для размещения исходников сайта и панели управления phpmyadmin.

```
# mkdir -p /web/sites/hl.zeroxzed.ru/www && mkdir /web/sites/hl.zeroxzed.ru/log
# mkdir -p /web/sites/p1m2a.zeroxzed.ru/www && mkdir /web/sites/p1m2a.zeroxzed.ru/log
```

Создадим конфиги nginx для этих виртуальных хостов. Я сразу буду делать их с учетом https, который мы настроим позже. Так что после создания не надо перезапускать веб сервер и проверять работу — будут ошибки. Виртуальный хост сайта показан на примере **wordpress**. Конфигурация собрана на основе рекомендаций из официальной документации конкретно для веб сервера nginx.

```
# mcedit /etc/nginx/conf.d/hl.zeroxzed.ru.conf
```

```
server {
    listen 80;
    server_name hl.zeroxzed.ru;
    root /web/sites/hl.zeroxzed.ru/www/;
    index index.php index.html index.htm;
    access_log /web/sites/hl.zeroxzed.ru/log/access.log main;
    error_log /web/sites/hl.zeroxzed.ru/log/error.log;

    location / {
    return 301 https://hl.zeroxzed.ru$request_uri;
    }

    location ~* ^.+.(js|css|png|jpg|jpeg|gif|ico|woff)$ {
    return 301 https://hl.zeroxzed.ru$request_uri;
    }

    location ~ \.php$ {
    return 301 https://hl.zeroxzed.ru$request_uri;
    }

    location = /favicon.ico {
    log_not_found off;
    access_log off;
    }

    location = /robots.txt {
    rewrite ^ /robots.txt break;
    allow all;
    log_not_found off;
    access_log off;
```

```
    }

    location ~ /\.ht {
    deny all;
    }
}

server {
     listen  80;
     server_name  www.hl.zeroxzed.ru;
     rewrite ^ https://hl.zeroxzed.ru$request_uri? permanent;
}

server {
    listen 443 ssl http2;
    server_name hl.zeroxzed.ru;
    root /web/sites/hl.zeroxzed.ru/www/;
    index index.php index.html index.htm;
    access_log /web/sites/hl.zeroxzed.ru/log/ssl-access.log main;
    error_log /web/sites/hl.zeroxzed.ru/log/ssl-error.log;

    keepalive_timeout           60;
    ssl_certificate             /etc/letsencrypt/live/hl.zeroxzed.ru/fullchain.pem;
    ssl_certificate_key         /etc/letsencrypt/live/hl.zeroxzed.ru/privkey.pem;
    ssl_protocols               TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers 'ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-
GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-
SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-
SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-
AES256-SHA:DHE-RSA-AES256-SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-
SHA:AES:CAMELLIA:DES-CBC3-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-
SHA:!KRB5-DES-CBC3-SHA';
    ssl_dhparam                 /etc/ssl/certs/dhparam.pem;
```

```
    add_header                  Strict-Transport-Security 'max-age=604800';

    location / {
    try_files $uri $uri/ /index.php?$args;
    }

    location ~* ^.+.(js|css|png|jpg|jpeg|gif|ico|woff)$ {
    access_log off;
    expires max;
    }

    location ~ \.php$ {
    try_files  $uri =404;
    fastcgi_pass   unix:/var/run/php-fpm/php-fpm.sock;
    #fastcgi_pass    127.0.0.1:9000;
    fastcgi_index index.php;
    fastcgi_param DOCUMENT_ROOT /web/sites/hl.zeroxzed.ru/www/;
    fastcgi_param SCRIPT_FILENAME /web/sites/hl.zeroxzed.ru/www$fastcgi_script_name;
    fastcgi_param PATH_TRANSLATED /web/sites/hl.zeroxzed.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_param HTTPS on;
    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;
    }
}

server {
     listen  443 ssl http2;
     server_name  www.hl.zeroxzed.ru;
     rewrite ^ https://hl.zeroxzed.ru$request_uri? permanent;
}
```

В данной конфигурации настроены все необходимые редиректы, при этом отключен редирект файла robots.txt. Он отдельно отдается по http и https. Это требуется для яндекса во время перехода с http на https и склейки зеркал.

Для phpmyadmin рисуем конфиг попроще.

```
# mcedit /etc/nginx/conf.d/p1m2a.zeroxzed.ru.conf
```

```
server {
    listen 443 ssl http2;
    server_name p1m2a.zeroxzed.ru;
    root /web/sites/p1m2a.zeroxzed.ru/www/;
    index index.php index.html index.htm;
    access_log /web/sites/p1m2a.zeroxzed.ru/log/ssl-access.log main;
    error_log /web/sites/p1m2a.zeroxzed.ru/log/ssl-error.log;

    keepalive_timeout           60;
    ssl_certificate             /etc/letsencrypt/live/p1m2a.zeroxzed.ru/fullchain.pem;
    ssl_certificate_key         /etc/letsencrypt/live/p1m2a.zeroxzed.ru/privkey.pem;
    ssl_protocols               TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers 'ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-
GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-
SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-
SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-
AES256-SHA:DHE-RSA-AES256-SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-
SHA:AES:CAMELLIA:DES-CBC3-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-
SHA:!KRB5-DES-CBC3-SHA';
    ssl_dhparam                 /etc/ssl/certs/dhparam.pem;
    add_header                  Strict-Transport-Security 'max-age=604800';

    location ~ \.php$ {
    fastcgi_pass   unix:/var/run/php-fpm/php-fpm.sock;
    #fastcgi_pass    127.0.0.1:9000;
    fastcgi_index index.php;
    fastcgi_param DOCUMENT_ROOT /web/sites/p1m2a.zeroxzed.ru/www/;
    fastcgi_param SCRIPT_FILENAME /web/sites/p1m2a.zeroxzed.ru/www$fastcgi_script_name;
    fastcgi_param PATH_TRANSLATED /web/sites/p1m2a.zeroxzed.ru/www$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;
    }
}

server {
     listen  443 ssl http2;
     server_name  www.p1m2a.zeroxzed.ru;
     rewrite ^ https://p1m2a.zeroxzed.ru$request_uri? permanent;
}

server {
    listen 80;
    server_name p1m2a.zeroxzed.ru;
    root /web/sites/p1m2a.zeroxzed.ru/www/;
    index index.php index.html index.htm;
    access_log /web/sites/p1m2a.zeroxzed.ru/log/access.log main;
    error_log /web/sites/p1m2a.zeroxzed.ru/log/error.log;

    location / {
    return 301 https://p1m2a.zeroxzed.ru$request_uri;
    try_files $uri $uri/ /index.php?$args;
    }
}
```

Сохраняем конфиги виртуальных хостов nginx и продолжаем настройку производительного веб сервера.

Установка php-fpm 7.1

Установка и настройка 7-й версии php на centos не очень простая задача. Ранее я уже рассказывал как обновить php до 7-й версии, но в итоге откатился назад. Прошло прилично времени и откатываться уже не будем, так как большинство проблем исправлены.

Основные трудности возникают с тем, что в официальных репозиториях очень старые версии php, но при этом они часто есть в зависимостях к другим пакетам. В итоге, обновившись неаккуратно до 7.1 можно получить проблемы с установкой и обновлением, к примеру, phpmyadmin или zabbix. В комментариях к моим статьям я иногда вижу эти ошибки и по тексту ошибок сразу понимаю, что проблема с зависимостями.

Вторая проблема в том, что надо определить, какой репозиторий использовать для установки php7. Их существует очень много. К примеру, мой хороший знакомый в своей статье по настройке web сервера использует репозиторий Webtatic. В принципе, чтобы просто поставить php 7-й версии это нормальный вариант. Но если вы после этого захотите установить phpmyadmin через yum уже ничего не получится. Будет ошибка зависимостей, которые нужно будет как-то руками разбирать.

То же самое будет и с другими пакетами. К примеру, zabbix без плясок с бубнами скорее всего не встанет. В сторонних репозиториях есть еще одна проблема. Иногда они закрываются. И это станет для вас большой проблемой на боевом сервере. Так что к выбору репозитория нужно подходить очень аккуратно и внимательно. Я до сих пор иногда встречаю настроенные сервера centos 5 с очень популярным в прошлом репозиторием centos.alt.ru, который закрылся. Сейчас это уже не так актуально, так как таких серверов осталось мало, но некоторое время назад мне это доставляло серьезные неудобства.

Для установки свежей версии php я буду использовать репозиторий **Remi**. Это известный и популярный репозиторий, который ведет сотрудник RedHat. И хотя надежность репозитория, который ведет один человек не так высока, но ничего лучше и надежнее remi лично я не нашел для своих целей. Если вы можете что-то посоветовать на этот счет — комментарии в вашем распоряжении. Буду благодарен за дельный совет.

Подключаем remi репозиторий для centos 7.

```
# rpm -Uhv http://rpms.remirepo.net/enterprise/remi-release-7.rpm
```

Я получил ошибку:

```
Retrieving http://rpms.remirepo.net/enterprise/remi-release-7.rpm
warning: /var/tmp/rpm-tmp.nwcDV1: Header V4 DSA/SHA1 Signature, key ID 00f97f56: NOKEY
```

```
error: Failed dependencies:
       epel-release = 7 is needed by remi-release-7.3-2.el7.remi.noarch
```

Тут все понятно, нужен репозиторий epel. Те, кто готовили сервер по моей статье по базовой настройке сервера его уже подключили, а те кто не делали этого, подключают сейчас:

```
# yum install epel-release
```

После этого повторяем установку remi, все должно пройти нормально. Проверим список подключенных репозиториев.

```
# yum repolist
```

```
repo id                                                repo name                                                                  status
base/7/x86_64                                          CentOS-7 - Base                                                             9,591
epel/x86_64                                            Extra Packages for Enterprise Linux 7 - x86_64                             12,024
extras/7/x86_64                                        CentOS-7 - Extras                                                            278
nginx/x86_64                                           nginx repo                                                                    90
remi-safe                                              Safe Remi's RPM repository for Enterprise Linux 7 - x86_64                 2,526
updates/7/x86_64                                       CentOS-7 - Updates                                                         1,041
repolist: 25,550
[root@hl conf.d]#
```


У меня такая картинка получилась.

Активируем репу **remi-php71**, для этого выполняем команду:

```
# yum-config-manager --enable remi-php71
```

Если получаете ошибку:

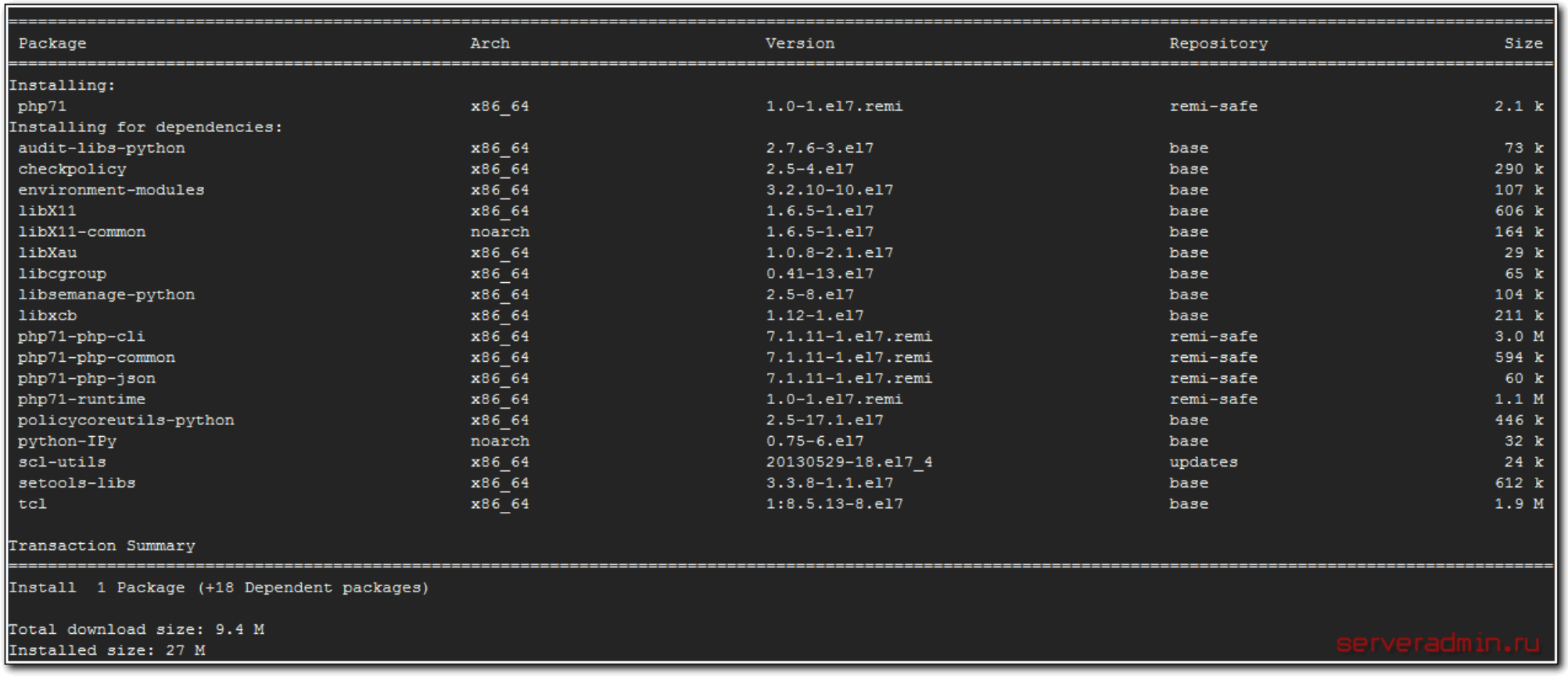
```
bash: yum-config-manager: command not found
```

то установите пакет **yum-utils**.

```
# yum install yum-utils
```

Теперь устанавливаем **php7.1**.

```
# yum install php71
```



```
================================================================================
 Package                 Arch          Version                 Repository   Size
================================================================================
Installing:
 php71                   x86_64        1.0-1.el7.remi          remi-safe   2.1 k
Installing for dependencies:
 audit-libs-python       x86_64        2.7.6-3.el7             base         73 k
 checkpolicy             x86_64        2.5-4.el7               base        290 k
 environment-modules     x86_64        3.2.10-10.el7           base        107 k
 libX11                  x86_64        1.6.5-1.el7             base        606 k
 libX11-common           noarch        1.6.5-1.el7             base        164 k
 libXau                  x86_64        1.0.8-2.1.el7           base         29 k
 libcgroup               x86_64        0.41-13.el7             base         65 k
 libsemanage-python      x86_64        2.5-8.el7               base        104 k
 libxcb                  x86_64        1.12-1.el7              base        211 k
 php71-php-cli           x86_64        7.1.11-1.el7.remi       remi-safe   3.0 M
 php71-php-common        x86_64        7.1.11-1.el7.remi       remi-safe   594 k
 php71-php-json          x86_64        7.1.11-1.el7.remi       remi-safe    60 k
 php71-runtime           x86_64        1.0-1.el7.remi          remi-safe   1.1 M
 policycoreutils-python  x86_64        2.5-17.1.el7            base        446 k
 python-IPy              noarch        0.75-6.el7              base         32 k
 scl-utils               x86_64        20130529-18.el7_4       updates      24 k
 setools-libs            x86_64        3.3.8-1.1.el7           base        612 k
 tcl                     x86_64        1:8.5.13-8.el7          base        1.9 M

Transaction Summary
================================================================================
Install  1 Package (+18 Dependent packages)

Total download size: 9.4 M
Installed size: 27 M
```


Установим **php-fpm** и наиболее популярные модули, которые могут пригодится в процессе эксплуатации веб сервера.

```
# yum install 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 php-zip
```

```
================================================================================================================================================
Installing:
 php-cli                        x86_64         7.1.11-1.el7.remi                          remi-php71          4.6 M
 php-fpm                        x86_64         7.1.11-1.el7.remi                          remi-php71          1.6 M
 php-gd                         x86_64         7.1.11-1.el7.remi                          remi-php71           72 k
 php-ldap                       x86_64         7.1.11-1.el7.remi                          remi-php71           63 k
 php-mbstring                   x86_64         7.1.11-1.el7.remi                          remi-php71          575 k
 php-mysqlnd                    x86_64         7.1.11-1.el7.remi                          remi-php71          227 k
 php-odbc                       x86_64         7.1.11-1.el7.remi                          remi-php71           81 k
 php-pdo                        x86_64         7.1.11-1.el7.remi                          remi-php71          121 k
 php-pear                       noarch         1:1.10.5-2.el7.remi                        remi-php71          354 k
 php-pecl-memcache              x86_64         3.0.9-0.9.20170802.e702b5f.el7.remi.7.1    remi-php71           83 k
 php-snmp                       x86_64         7.1.11-1.el7.remi                          remi-php71           60 k
 php-soap                       x86_64         7.1.11-1.el7.remi                          remi-php71          203 k
 php-xml                        x86_64         7.1.11-1.el7.remi                          remi-php71          206 k
 php-xmlrpc                     x86_64         7.1.11-1.el7.remi                          remi-php71           77 k
Installing for dependencies:
 fontconfig                     x86_64         2.10.95-11.el7                             base                229 k
 fontpackages-filesystem        noarch         1.44-8.el7                                 base                9.9 k
 gd-last                        x86_64         2.2.5-1.el7.remi                           remi-safe           133 k
 jbigkit-libs                   x86_64         2.0-11.el7                                 base                 46 k
 libXpm                         x86_64         3.5.12-1.el7                               base                 55 k
 libjpeg-turbo                  x86_64         1.2.90-5.el7                               base                134 k
 libpng                         x86_64         2:1.5.13-7.el7_2                           base                213 k
 libtiff                        x86_64         4.0.3-27.el7_3                             base                170 k
 libtool-ltdl                   x86_64         2.4.2-22.el7_3                             base                 49 k
 libwebp                        x86_64         0.3.0-7.el7                                base                170 k
 libxslt                        x86_64         1.1.28-5.el7                               base                242 k
 lm_sensors-libs                x86_64         3.4.0-4.20160601gitf9185e5.el7             base                 41 k
 lyx-fonts                      noarch         2.2.3-1.el7                                epel                159 k
 net-snmp                       x86_64         1:5.7.2-28.el7                             base                321 k
 net-snmp-agent-libs            x86_64         1:5.7.2-28.el7                             base                704 k
 net-snmp-libs                  x86_64         1:5.7.2-28.el7                             base                748 k
 perl-Data-Dumper               x86_64         2.145-3.el7                                base                 47 k
 php-common                     x86_64         7.1.11-1.el7.remi                          remi-php71          1.0 M
 php-json                       x86_64         7.1.11-1.el7.remi                          remi-php71           57 k
 php-process                    x86_64         7.1.11-1.el7.remi                          remi-php71           75 k
 unixODBC                       x86_64         2.3.1-11.el7                               base                413 k

Transaction Summary
================================================================================================================================================
```

Запускаем php-fpm и добавляем в автозагрузку.

```
# systemctl start php-fpm
# systemctl enable php-fpm
```

Проверяем, запустился ли он.

```
# netstat -tulpn | grep php-fpm
tcp 0 0 127.0.0.1:9000 0.0.0.0:* LISTEN 9084/php-fpm: maste
```

Все в порядке, повис на порту 9000. Запустим его через unix сокет. Для этого открываем конфиг */etc/php-fpm.d/www.conf* и комментируем строку:

```
;listen = 127.0.0.1:9000
```

Вместо нее добавляем несколько других:

```
listen = /var/run/php-fpm/php-fpm.sock
listen.mode = 0660
listen.owner = nginx
listen.group = nginx
```

Заодно измените пользователя, от которого будет работать php-fpm. Вместо apache укажите nginx.

```
user = nginx
group = nginx
```

Перезапускаем php-fpm.

```
# systemctl restart php-fpm
```

Проверяем, стартовал ли указанный сокет.

```
# ll /var/run/php-fpm/php-fpm.sock
srw-rw----. 1 nginx nginx 0 Oct 26 18:08 /var/run/php-fpm/php-fpm.sock
```

На текущий момент с настройкой php-fpm закончили, двигаемся дальше.

Для того, чтобы проверить работу нашего веб сервера, нужно установить ssl сертификаты. Без них nginx с текущим конфигом не запустится. Исправляем это.

Настройка бесплатного ssl сертификата Lets Encrypt

Устанавливаем пакет **certbot** для получения бесплатного ssl сертификата от let's encrypt.

```
# yum install certbot
```

Запускаем программу для генерации сертификата.

```
# certbot certonly
```

Вам в консоли будут заданы несколько вопросов. Вот мои ответы, необходимые для успешного получения сертификата. Первый раз мы получим сертификаты, используя временный веб сервер самого certbot, так как наш еще не работает. Далее обновлять сертификаты будем в автоматическом режиме с помощью временной директории в корне виртуального хоста.

```
# certbot certonly
Saving debug log to /var/log/letsencrypt/letsencrypt.log

How would you like to authenticate with the ACME CA?
-------------------------------------------------------------------------------
1: Spin up a temporary webserver (standalone)
```

```
2: Place files in webroot directory (webroot)
-------------------------------------------------------------------------------
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 1
Plugins selected: Authenticator standalone, Installer None
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel): zeroxzed@gmail.com
Starting new HTTPS connection (1): acme-v01.api.letsencrypt.org

-------------------------------------------------------------------------------
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.1.1-August-1-2016.pdf. You must agree
in order to register with the ACME server at
https://acme-v01.api.letsencrypt.org/directory
-------------------------------------------------------------------------------
(A)gree/(C)ancel: A

-------------------------------------------------------------------------------
Would you be willing to share your email address with the Electronic Frontier
Foundation, a founding partner of the Let's Encrypt project and the non-profit
organization that develops Certbot? We'd like to send you email about EFF and
our work to encrypt the web, protect its users and defend digital rights.
-------------------------------------------------------------------------------
(Y)es/(N)o: N
Please enter in your domain name(s) (comma and/or space separated)  (Enter 'c'
to cancel): hl.zeroxzed.ru
Obtaining a new certificate
Performing the following challenges:
tls-sni-01 challenge for hl.zeroxzed.ru
Waiting for verification...
Cleaning up challenges

IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:
```

```
  /etc/letsencrypt/live/hl.zeroxzed.ru/fullchain.pem
  Your key file has been saved at:
  /etc/letsencrypt/live/hl.zeroxzed.ru/privkey.pem
  Your cert will expire on 2018-01-24. To obtain a new or tweaked
  version of this certificate in the future, simply run certbot
  again. To non-interactively renew *all* of your certificates, run
  "certbot renew"
- Your account credentials have been saved in your Certbot
  configuration directory at /etc/letsencrypt. You should make a
  secure backup of this folder now. This configuration directory will
  also contain certificates and private keys obtained by Certbot so
  making regular backups of this folder is ideal.
- If you like Certbot, please consider supporting our work by:

  Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
  Donating to EFF:                    https://eff.org/donate-le
```

Для успешного создания бесплатных ssl сертификатов от lets encrypt у вас должны быть корректно настроены DNS записи для доменов, на которые выпускаются сертификаты.

Итак, сертификаты получили. Теперь можно проверить конфигурацию nginx и запустить его. Проверяем конфиг:

```
# nginx -t
```

Если получаете ошибку:

```
nginx: [emerg] BIO_new_file("/etc/ssl/certs/dhparam.pem") failed (SSL: error:02001002:system library:fopen:No such file or
directory:fopen('/etc/ssl/certs/dhparam.pem','r') error:2006D080:BIO routines:BIO_new_file:no such file)
nginx: configuration file /etc/nginx/nginx.conf test failed
```

То генерируете необходимый ключ:

```
# openssl dhparam -out /etc/ssl/certs/dhparam.pem 4096
```

Генерация будет длиться долго (у меня 20 минут длилось на двух ядрах). Снова проверяйте конфигурацию. Если ошибок нет, то перезапустим nginx.

```
# systemctl restart nginx
```

Настройка nginx на этом завершена. Он должен корректно запуститься и работать в рабочем режиме.

Теперь сделаем так, чтобы сертификаты автоматически обновлялись перед истечением срока действия. Для этого необходимо изменить конфигурации доменов. Они располагаются в директории */etc/letsencrypt/renewal*. Так как мы генерировали сертификаты с помощью временного веб сервера, наш текущий конфиг *hl.zeroxzed.ru.conf* выглядит вот так:

```
# renew_before_expiry = 30 days
version = 0.18.1
archive_dir = /etc/letsencrypt/archive/hl.zeroxzed.ru
cert = /etc/letsencrypt/live/hl.zeroxzed.ru/cert.pem
privkey = /etc/letsencrypt/live/hl.zeroxzed.ru/privkey.pem
chain = /etc/letsencrypt/live/hl.zeroxzed.ru/chain.pem
fullchain = /etc/letsencrypt/live/hl.zeroxzed.ru/fullchain.pem

# Options used in the renewal process
[renewalparams]
authenticator = standalone
installer = None
account = e9c86e6aa57b45f9614bc7c0015927a5
```

Приводим его к следующему виду:

```
# renew_before_expiry = 30 days
version = 0.18.1
archive_dir = /etc/letsencrypt/archive/hl.zeroxzed.ru
```

```
cert = /etc/letsencrypt/live/hl.zeroxzed.ru/cert.pem
privkey = /etc/letsencrypt/live/hl.zeroxzed.ru/privkey.pem
chain = /etc/letsencrypt/live/hl.zeroxzed.ru/chain.pem
fullchain = /etc/letsencrypt/live/hl.zeroxzed.ru/fullchain.pem

# Options used in the renewal process
[renewalparams]
authenticator = webroot
installer = None
account = e9c86e6aa57b45f9614bc7c0015927a5
post_hook = nginx -s reload
[[webroot_map]]
www.hl.zeroxzed.ru = /web/sites/hl.zeroxzed.ru/www
hl.zeroxzed.ru = /web/sites/hl.zeroxzed.ru/www
```

По аналогии делаете с остальными виртуальными хостами, для которых используете бесплатные сертификаты let’s encrypt. Осталось дело за малым — настроить автоматический выпуск новых ssl сертификатов, взамен просроченным. Для этого добавляем в */etc/crontab* следующую строку:

```
# Cert Renewal
30 2 * * * root /usr/bin/certbot renew --post-hook "nginx -s reload" >> /var/log/le-renew.log
```

Все, с сертификатами закончили. Двигаемся дальше в настройке web сервера.

Установка mariadb 10 на CentOS 7

Дошла очередь до установки сервера баз данных для web сервера на CentOS 7 — **MariaDB**. По аналогии с другим софтом, в официальном репозитории очень старая версия mariadb — 5.5. Я же буду устанавливать последнюю стабильную версию на момент написания статьи — 10.2.

Для того, чтобы подключить репозиторий MariaDB, можно воспользоваться специальной страницей на официальном сайте, где можно задать параметры системы и получить конфиг репозитория.

В моем случае конфиг получился следующий.

```
# cat /etc/yum.repos.d/mariadb.repo
```

```
[mariadb]
name = MariaDB
baseurl = http://yum.mariadb.org/10.2/centos7-amd64
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=1
```

Устанавливаем последнюю версию mariadb на centos.

```
# yum install MariaDB-server MariaDB-client
```

```
================================================================================================================================================
 Package                                  Arch                   Version                                  Repository                   Size
================================================================================================================================================
Installing:
 MariaDB-client                           x86_64                 10.2.9-1.el7.centos                      mariadb                      48 M
 MariaDB-compat                           x86_64                 10.2.9-1.el7.centos                      mariadb                     2.8 M
     replacing  mariadb-libs.x86_64 1:5.5.56-2.el7
 MariaDB-server                           x86_64                 10.2.9-1.el7.centos                      mariadb                     108 M
Installing for dependencies:
 MariaDB-common                           x86_64                 10.2.9-1.el7.centos                      mariadb                     155 k
 boost-program-options                    x86_64                 1.53.0-27.el7                            base                        156 k
 galera                                   x86_64                 25.3.20-1.rhel7.el7.centos               mariadb                     8.0 M
 libaio                                   x86_64                 0.3.109-13.el7                           base                         24 k
 perl-Compress-Raw-Bzip2                  x86_64                 2.061-3.el7                              base                         32 k
 perl-Compress-Raw-Zlib                   x86_64                 1:2.061-4.el7                            base                         57 k
 perl-DBI                                 x86_64                 1.627-4.el7                              base                        802 k
 perl-IO-Compress                         noarch                 2.061-2.el7                              base                        260 k
 perl-Net-Daemon                          noarch                 0.48-5.el7                               base                         51 k
 perl-PlRPC                               noarch                 0.2020-14.el7                            base                         36 k
 rsync                                    x86_64                 3.0.9-18.el7                             base                        360 k

Transaction Summary
================================================================================================================================================
Install  3 Packages (+11 Dependent packages)                                                                          serveradmin.ru
```

Убедитесь, что база данных ставится из нужного репозитория.

Запускаем mariadb и добавляем в автозагрузку.

```
# systemctl start mariadb
# systemctl enable mariadb
```

Запускаем скрипт начальной конфигурации mysql и задаем пароль для root. Все остальное можно оставить по-умолчанию.

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

Сервер баз данных mysql для нашего web сервера готов. Продолжаем настройку. Установим панель управления mysql — phpmyadmin.

Установка phpmyadmin

Кратко расскажу про установку phpmyadmin в контексте данной статьи. Подробно не буду останавливаться на этом, так как статья и так получается очень объемная, а я еще не все рассказал. Вопрос настройки phpmyadmin я очень подробно рассмотрел отдельно. За подробностями можно сходить туда.

Устанавливаем phpmyadmin через yum. Если ранее все сделали правильно, то конфликтов с зависимостями быть не должно.

```
# yum install phpmyadmin
```

```
================================================================================================================================================
 Package                                  Arch                   Version                                 Repository                      Size
================================================================================================================================================
Installing:
 phpMyAdmin                               noarch                 4.4.15.10-2.el7                         epel                           4.7 M
Installing for dependencies:
 dejavu-fonts-common                      noarch                 2.33-6.el7                              base                            64 k
 dejavu-sans-fonts                        noarch                 2.33-6.el7                              base                           1.4 M
 libtidy                                  x86_64                 5.4.0-1.el7                             epel                           174 k
 libzip5                                  x86_64                 1.3.0-1.el7.remi                        remi-safe                       59 k
 php-bcmath                               x86_64                 7.1.11-1.el7.remi                       remi-php71                      66 k
 php-fedora-autoloader                    noarch                 1.0.0-1.el7                             epel                           9.6 k
 php-pecl-zip                             x86_64                 1.15.1-1.el7.remi.7.1                   remi-php71                      50 k
 php-php-gettext                          noarch                 1.0.12-1.el7                            epel                            23 k
 php-tcpdf                                noarch                 6.2.13-1.el7                            epel                           2.1 M
 php-tcpdf-dejavu-sans-fonts              noarch                 6.2.13-1.el7                            epel                           257 k
 php-tidy                                 x86_64                 7.1.11-1.el7.remi                       remi-php71                      59 k

Transaction Summary
================================================================================================================================================
Install  1 Package (+11 Dependent packages)                                                                          serveradmin.ru
```

Phpmyadmin по-умолчанию сконфигурирована для работы с httpd. Для того, чтобы в будущем автоматически обновлять ее, просто сделаем символьную ссылку из директории с исходниками панели в наш виртуальный хост.

```
# rm -df /web/sites/p1m2a.zeroxzed.ru/www
# ln -s /usr/share/phpMyAdmin /web/sites/p1m2a.zeroxzed.ru/www
```

Выставляем правильные права на директорию с php сессиями. Без этого работать phpmyadmin не будет.

```
# chown nginx:nginx /var/lib/php/session/
```

Можно заходить и проверять работу phpmyadmin. Ее установка закончена.

Доступ к сайту по sftp

Настройка сервера почти завершена. Если вы администратор и единственный пользователь, то больше можно ничего не делать. Вы и так сможете загрузить на сервер все что нужно тем или иным способом. Если же вы будете передавать управление сайтами другим людям, им нужен доступ к директории с исходниками сайта. Раньше для этих целей повсеместно использовали ftp. Если вы хотите так сделать, у меня есть статья по настройке ftp сервера vsftpd.

Я же предлагаю использовать **sftp** по нескольким причинам:

1. Он безопаснее.
2. Его быстрее настроить.
3. Не надо отдельно настраивать firewall.

Статью по настройке sftp доступа я уже тоже писал, все подробности там. Здесь без комментариев выполним необходимые действия.

Создаем пользователя для подключения к сайту. Я обычно использую имя пользователя пересекающееся с названием сайта. Так удобнее управлять.

```
# useradd -s /sbin/nologin hl.zeroxzed.ru
# passwd hl.zeroxzed.ru
```

Открываем конфиг ssh по пути */etc/ssh/sshd_config* и комментируем там одну строку, добавляя далее несколько новых.

```
#Subsystem sftp /usr/libexec/openssh/sftp-server
Subsystem sftp internal-sftp
Match User hl.zeroxzed.ru
ChrootDirectory /web/sites/hl.zeroxzed.ru
ForceCommand internal-sftp
```

Перезапускаем службу sshd.

```
# systemctl restart sshd
```

Этого уже достаточно, чтобы вы могли подключиться к сайту, к примеру, с помощью программы winscp. Если что-то пойдет не так и будут какие-то ошибки, то смотреть подробности нужно в логе */var/log/secure*. Но тут возникает много нюансов с правами к файлам и директориям. Дальше я расскажу, как их аккуратно и грамотно разрулить, чтобы у нас не было проблем с дальнейшей работой сайтов от разных пользователей.

Работа с сайтами разных пользователей на одном веб сервере

Самый простой способ решить проблему с правами доступа, это сделать владельцем папки с сайтом пользователя, который подключается по sftp. Тогда он сможет нормально работать с файлами, загружать и удалять их. Если доступ в качестве группы установить для nginx, то в целом все будет работать. Для каких-то сайтов такой вариант может оказаться подходящим. То есть сделать надо вот так:

```
# chown -R hl.zeroxzed.ru:nginx /web/sites/hl.zeroxzed.ru/
# chmod -R 0775 /web/sites/hl.zeroxzed.ru/
```

Но при такой схеме будут проблемы с движками сайтов, которые автоматом что-то к себе загружают. Какие-то галереи не будут работать. К примеру, wordpress не сможет автоматически загружать плагины, будет просить доступ к ftp. В общем, могут возникнуть некоторые неудобства. Сейчас мы их исправим.

Еще обращаю внимание на один нюанс. Chroot доступ для sftp не будет работать, если владельцем директории, куда чрутимся, будет не root. Только что мы сделали владельцем каталога с сайтом и всего, что внутри него пользователя hl.zeroxzed.ru. Теперь надо вернуть обратно владельцем исходного каталога рута, а все, что внутри него остается как мы и хотим — будет принадлежать hl.zeroxzed.ru.

```
# chown root:root /web/sites/hl.zeroxzed.ru/
# chmod 0755 /web/sites/hl.zeroxzed.ru/
```

А теперь сделаем все красиво. Назначаем владельцем содержимого нашего сайта только отдельного пользователя.

```
# chown -R hl.zeroxzed.ru:hl.zeroxzed.ru /web/sites/hl.zeroxzed.ru/
```

Возвращаем обратно рута владельцем корня chroot.

```
# chown root:root /web/sites/hl.zeroxzed.ru/
# chmod 0755 /web/sites/hl.zeroxzed.ru/
```

Обращаю внимание, что сначала мы рекурсивно назначаем права на все содержимое директорий, а потом возвращаем владельца root только на корень.

Добавляем пользователя nginx в группу hl.zeroxzed.ru.

```
# usermod -aG hl.zeroxzed.ru nginx
```

Создаем отдельный pool для php-fpm, который будет обслуживать сайт hl.zeroxzed.ru и будет запускаться от этого пользователя. Для этого копируем существующий конфиг */etc/php-fpm.d/www.conf* и изменяем в нем несколько строк.

```
# cd /etc/php-fpm.d && cp www.conf hl.zeroxzed.ru.conf
```

```
[hl.zeroxzed.ru]
user = hl.zeroxzed.ru
group = hl.zeroxzed.ru
listen = /var/run/php-fpm/hl.zeroxzed.ru.sock
listen.owner = hl.zeroxzed.ru
listen.group = hl.zeroxzed.ru
```

Мы поменяли название пула, запустили его от отдельного пользователя и назначили ему отдельный сокет. Теперь идем в настройки этого виртуального хоста в nginx — */etc/nginx/conf.d/hl.zeroxzed.ru.conf* и везде меняем старое значение сокета

```
fastcgi_pass unix:/var/run/php-fpm/php-fpm.sock;
```

на новое

```
fastcgi_pass unix:/var/run/php-fpm/hl.zeroxzed.ru.sock;
```

Перезапускаем nginx и php-fpm и проверяем работу сайта от отдельного пользователя.

```
# systemctl restart nginx
# systemctl restart php-fpm
```

Я рекомендую подключиться по sftp, закинуть исходники wordpress, установить его и добавить новую тему, чтобы проверить, что все корректно работает. По аналогии проделанные выше действия повторяются для всех остальных сайтов.

Ротация логов виртуальных хостов

Последний штрих в настройке web сервера — ротация логов виртуальных хостов. Если этого не сделать, то через какое-то, обычно продолжительное, время возникает проблема в связи с огромным размером лог файла.

У нас уже будет файл конфигурации **logrotate** для nginx, который был создан во время установки — */etc/logrotate.d/nginx*. Приведем его к следующему виду:

```
/var/log/nginx/*log
/web/sites/p1m2a.zeroxzed.ru/log/*log {

    create 0644 nginx nginx
    size=1M
    rotate 10
    missingok
    notifempty
    compress
    sharedscripts
    postrotate
        /bin/kill -USR1 `cat /run/nginx.pid 2>/dev/null` 2>/dev/null || true
    endscript
}

/web/sites/hl.zeroxzed.ru/log/*log {
```

```
    create 0644 hl.zeroxzed.ru hl.zeroxzed.ru
    size=1M
    rotate 10
    missingok
    notifempty
    compress
    sharedscripts
    postrotate
        /bin/kill -USR1 `cat /run/nginx.pid 2>/dev/null` 2>/dev/null || true
    endscript
}
```

Я предлагаю ротировать файлы логов по достижению ими размера в 1Мб, сжимать после ротации и хранить 10 архивов с логом. Для виртуальных хостов, работающих от отдельного пользователя, новые логи создаются сразу с соответствующими правами, чтобы у пользователя был доступ к ним. Для всех остальных хостов можно использовать самое первое правило, просто добавляя туда новые пути для логов.

Это просто пример конфигурации. Все параметры вы можете поменять по своему усмотрению. Примеров конфигурации logrotate в интернете много.

На этом все. Я рассмотрел все основные моменты, которые необходимы для установки и настройки производительного web сервера на основе nginx и php-fpm последних версий. При этом рассказал о некоторых вещах, которые повышают удобство и гибкость эксплуатации сервера.

Заключение

> Не понравилась статья и хочешь научить меня администрировать? Пожалуйста, я люблю учиться. Комментарии в твоем распоряжении. Расскажи, как сделать правильно!

Тема настройки веб сервера обширна. Рассмотреть все варианты в одной статье невозможно, так как функционал будет разниться, в зависимости от назначения сервера. Тем не менее приведу еще несколько ссылок на материалы, которые имеют отношение к настройке web сервера:

- Полный бэкап сервера или отдельных сайтов.

- Мониторинг веб сервера и веб сайта с помощью zabbix.
- Защита админки wordpress с помощью fail2ban.
- Если у вас будут проблемы с ботами, то пригодится статья по блокировке доступа к сайту по странам.

Если еще что-то полезное вспомню, добавлю ссылки. Пока вроде все. Жду комментариев и отзывов. Написал все по своему опыту, как я обычно настраиваю веб сервера. Возможно что-то можно сделать более удобно и правильно.

Эта статья будет первой из цикла статей по настройке современного веб сервера. Далее мы будем защищать web сервер и готовить его к максимальным нагрузкам.