Сегодня займемся поднятием одной из наиболее востребованных ролей любого linux сервера, которые занимают в этом функциональном сегменте лидирующую роль. Настройка web сервера CentOS 8 на базе связки популярного http сервера **apache**, интерпретатора **php** и сервера баз данных **mysql**, или коротко — установка lamp. Данная связка является самой популярной конфигурацией на сегодняшний день среди веб хостинга. Хотя последнее время ей на пятки наступает та же компания, но на базе nginx, возможно уже и опередила, точных данных у меня нет на этот счет.

Содержание:

- 12 Заключение

> Если вам нужен высокопроизводительный web сервер на базе последних версий nginx и php-fpm, то читайте отдельную статью по этой теме — установка и настройка nginx и php-fpm7 на centos.

Цели статьи

1. Описать общий принцип настройки web сервера на базе apache.
2. Показать, как установить актуальные версии nginx, php и mariadb.
3. Привести примеры своих конфигов для перечисленных сервисов.
4. Настроить бесплатные ssl сертификаты let’s encrypt для сайтов.
5. Показать, как настроить selinux для web сервера apache (httpd).
6. Записать и показать видео настройки веб сервера apache.

Данная статья является частью единого цикла статьей про сервер Centos.

Введение

В этой статье я расскажу, как настроить web сервер на базе популярного стека технологий — apache и php с модулем prefork. В связи с выходом нового релиза Centos 8, многие статьи на эту тему стали не актуальны, так как версии софта в базовых репозиториях обновились и тот же php нет смысла ставить из стороннего репозитория.

Работать будем на сервере под управлением CentOS 8. 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.

z.serveradmin.ru	имя тестового виртуального хоста и сайта
/web/sites	директория для размещения виртуальных хостов

10.20.1.23	локальный ip адрес сервера
pma.serveradmin.ru	имя виртуального хоста для phpmyadmin

Если вам не хочется настраивать SELinux, то просто отключите его. Если же вы планируете включить и использовать после настройки, то переведите его в режим **Permissive**. В этом режиме он будет отключен, но все ограничения будут отражаться в лог файле audit.log. Это позволит нам в конце статьи сформировать модуль selinux для httpd и php и включить их.

```
# setenforce 0
```

Web сервер на CentOS 8

Итак, наш веб сервер centos будет состоять из трех основных компонентов — http сервера **apache**, интерпретатора языка программирования **php** и сервера баз данных **mysql**. Познакомимся немного с каждым из них:

1. **Apache** — http сервер или просто веб сервер апач. Является кросплатформенным ПО, поддерживающим практически все популярные операционные системы, в том числе и Windows. Ценится прежде всего за свою надежность и гибкость конфигурации, которую можно существенно расширить благодаря подключаемым модулям, которых существует великое множество. Из недостатков отмечают большую требовательность к ресурсам, по сравнению с другими серверами. Держать такую же нагрузку, как, к примеру, nginx, apache не сможет при схожих параметрах железа.
2. **PHP** — язык программирования общего назначения, который чаще всего применяется в веб разработке. На сегодняшний день это самый популярный язык в этой области применения. Поддерживается практически всеми хостинг-провайдерами.
3. **Mysql** — система управления базами данных. Завоевала свою популярность в среде малых и средних приложений, которых очень много в вебе. Так что, как и php, на сегодняшний день является самой популярной бд, использующейся на веб сайтах. Поддерживается большинством хостингов. В CentOS вместо mysql устанавливается **mariadb** — ответвление mysql. Они полностью совместимы, возможен в любой момент переход с одной субд на другую и обратно. Я встречал информацию, что mariadb пошустрее работает mysql и люди потихоньку перебираются на нее. На практике мне не довелось это наблюдать, так как никогда не работал с нагруженными базами данных. А в обычных условиях разница не заметна.

Подопытным сервером будет выступать виртуальная машина от ihor, характеристики следующие:

Процессор	2 ядра
Память	3 Gb
Диск	30 Gb SSD

Хочу сразу уточнить, что разбираю базовую дефолтную настройку. Для улучшения быстродействия, повышения надежности и удобства пользования нужно установить еще несколько инструментов, о чем я расскажу отдельно. В общем случае для организации веб сервера будет достаточно того, что есть в этой статье.

Настройка apache в CentOS 8

В CentOS служба apache называется **httpd**. Когда я только знакомился с этим дистрибутивом, мне было непривычно. В Freebsd и Debian, с которыми я до этого работал служба веб сервера называлась apache, хотя где-то я замечал, кажется во фрюхе, что файл конфигурации имеет имя httpd.conf. До сих пор я не знаю, почему распространились оба эти названия. Был бы рад, если бы со мной кто-то поделился информацией на этот счет в комментариях.

Теперь приступим к **установке apache**. В CentOS 8 это делается очень просто:

```
# dnf install httpd
```

Добавляем apache в автозагрузку:

```
# systemctl enable httpd
```

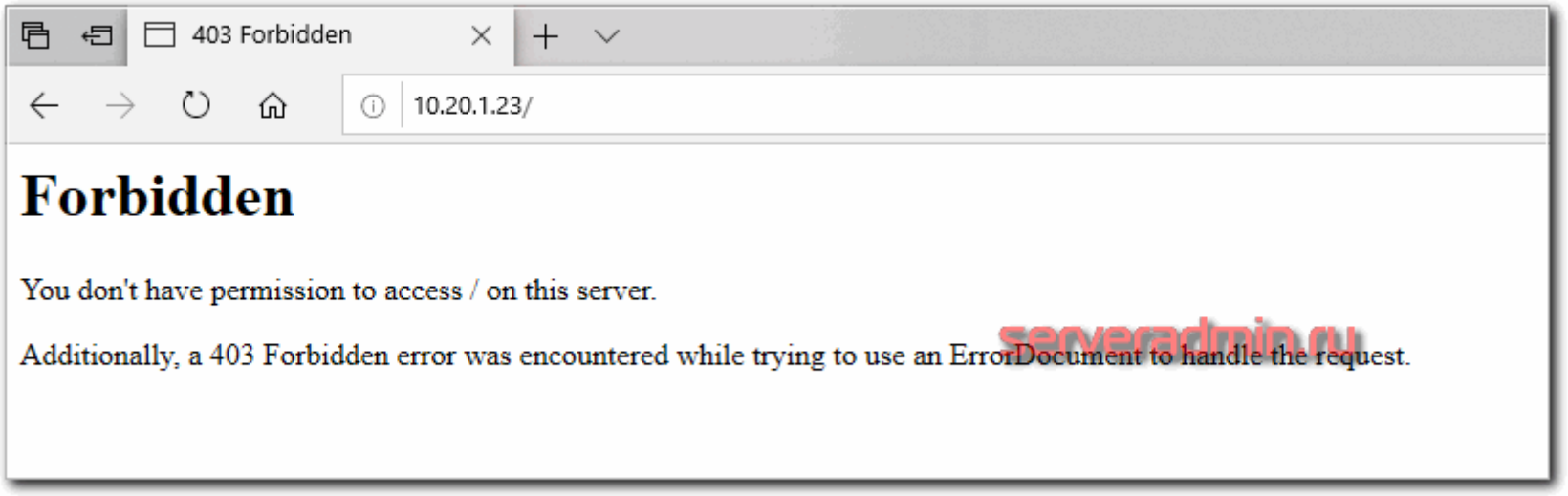
Запускаем apache в CentOS 8:

```
# systemctl start httpd
```

Проверяем, запустился ли сервер:

```
# netstat -tulnp | grep httpd
tcp6       0      0 :::80                   :::*                    LISTEN      21586/httpd
```

Все в порядке, повис на 80-м порту, как и положено. Уже сейчас можно зайти по адресу http://10.20.1.23 и увидеть картинку:



Если ничего не видите, скорее всего у вас не настроен firewall. Если не занимались его настройкой, то по-умолчанию в centos установлен firewalld. На нем открыть порты для web сервера можно следующими командами.

```
# firewall-cmd --permanent --zone=public --add-service=http
# firewall-cmd --permanent --zone=public --add-service=https
# firewall-cmd --reload
```

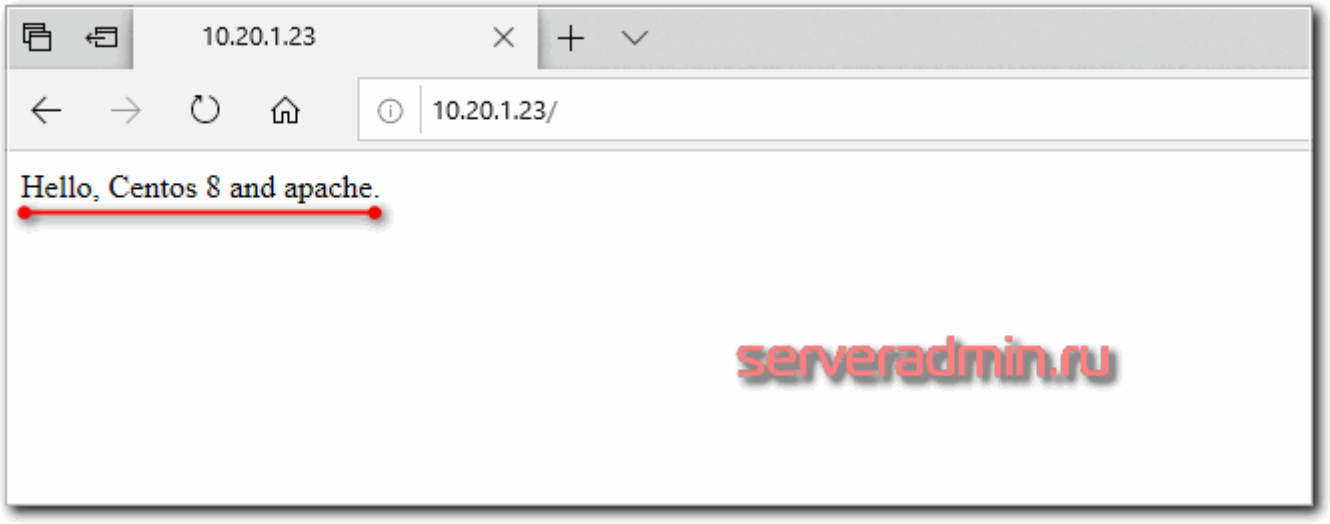
Проверить, открылись ли порты можно командой.

```
# firewall-cmd --list-all
```

Вместо ожидаемой картинки с каким-то приветствием, мы видим просто ошибку. Я не знаю, почему в Centos 8 в стандартном пакете httpd из базового репозитория не положили в директорию web сервера по-умолчанию */var/www/html* хоть какую-нибудь страничку. Там пусто, поэтому мы и видим ошибку. Давайте тогда сами что-то туда положим, чтобы просто проверить работу apache. Создаем простую страничку index.html.

```
# echo "Hello, Centos 8 and apache." >> /var/www/html/index.html
```

Проверяем страницу.



Все в порядке, веб сервер работает. Теперь займемся настройкой apache. Я предпочитаю следующую структуру веб хостинга:

/web/sites	раздел для размещения сайтов
/web/z.serveradmin.ru/www	директория для содержимого сайта
/web/z.serveradmin.ru/log	директория для логов сайта

Создаем для нее директории.

```
# mkdir -p /web/sites/z.serveradmin.ru/{www,log}
# chown -R apache. /web/sites/
```

Дальше редактируем файл конфигурации apache — **httpd.conf** по адресу */etc/httpd/conf*. Первым делом проверим, раскомментирована ли строчка в самом конце:

```
IncludeOptional conf.d/*.conf
```

Если нет, раскомментируем и идем в каталог */etc/httpd/conf.d*. Создаем там файл z.serveradmin.ru.conf:

```
# mcedit /etc/httpd/conf.d/z.serveradmin.ru.conf
```

```
<VirtualHost *:80>

    ServerName z.serveradmin.ru
    ServerAlias www.z.serveradmin.ru
    DocumentRoot /web/sites/z.serveradmin.ru/www

    ErrorLog /web/sites/z.serveradmin.ru/log/error.log
    CustomLog /web/sites/z.serveradmin.ru/log/access.log common

    <Directory /web/sites/z.serveradmin.ru/www>
        Options FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

</VirtualHost>
```

Перезапуск apache в centos

Проверим созданную конфигурацию и выполним перезапуск apache.

```
# apachectl -t
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using fe80::95:c6f3:e49f:7204. Set
the 'ServerName' directive globally to suppress this message
Syntax OK
```

В данном случае ошибок нет, показано предупреждение. Теперь делаем restart apache:

```
# systemctl restart httpd
```

Если возникли какие-то ошибки — смотрим лог apache */var/log/httpd/error_log*. Если все в порядке, то проверим, нормально ли настроен наш виртуальный хост. Для этого создадим в папке /web/sites/z.serveradmin.ru/www файл index.html следующего содержания:

```
# mcedit /web/sites/z.serveradmin.ru/www/index.html
```

```
<h1>Апач работает как надо, а надо хорошо!</h1>
```

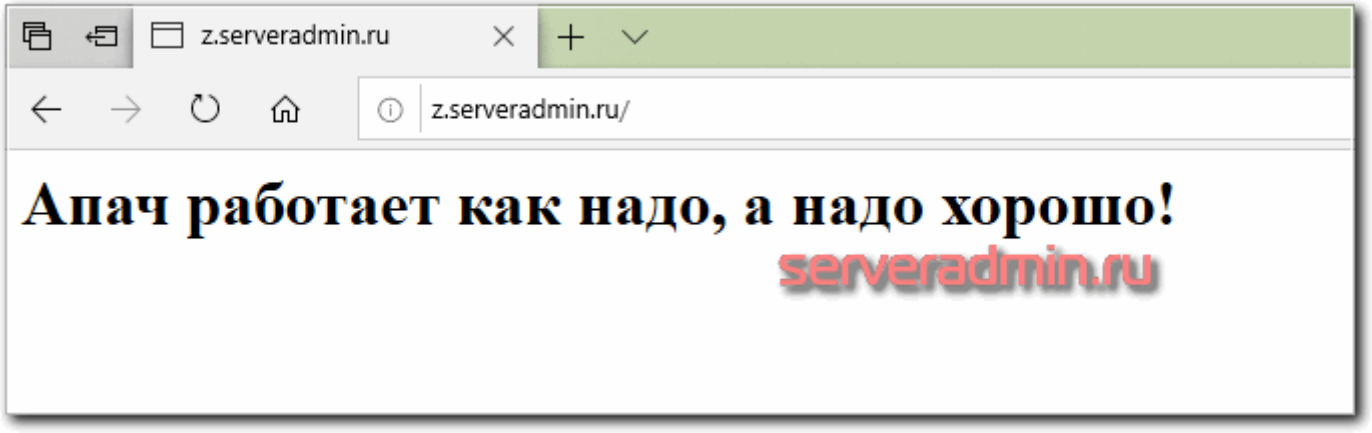
```
# chown apache. /web/sites/z.serveradmin.ru/www/index.html
```

Дальше на своем компьютере правим файл hosts, чтобы обратиться к нашему виртуальному хосту. Добавляем туда строчку:

```
10.20.1.23 z.serveradmin.ru
```

где 10.20.1.23 ip адрес нашего веб сервера. Это нужно сделать только в том случае, если настраиваете web сервер где-то в локальной сети без реально существующего доменного имени.

Теперь в браузере набираем адрес http://z.serveradmin.ru. Если видим картинку:



значит все правильно настроили. Если какие-то ошибки, то идем смотреть логи. Причем в данном случае не общий лог httpd, а лог ошибок конкретного виртуального хоста по адресу */web/z.serveradmin.ru/log/error.log*.

Установка php в CentOS 8

Установка php в Centos 8 сильно упростилась по сравнению с предыдущей версией, потому что в базовом репозитории хранится актуальная версия php 7.2, которой можно пользоваться. Пока нет необходимости подключать сторонние репозитории, так как версия 7.2 вполне свежа и актуальна. Если у вас нет необходимости использовать что-то новее, то можно остановиться на этой версии.

Устанавливаем php в CentOS 8, а так же некоторые популярные модули, которые могут пригодиться для того же phpmyadmin.

```
# dnf install php php-cli php-mysqlnd php-json php-gd php-ldap php-odbc php-pdo php-opcache php-pear php-xml php-xmlrpc php-mbstring php-snmp php-soap php-zip
```

Выполним перезапуск apache:

```
# systemctl restart httpd
```

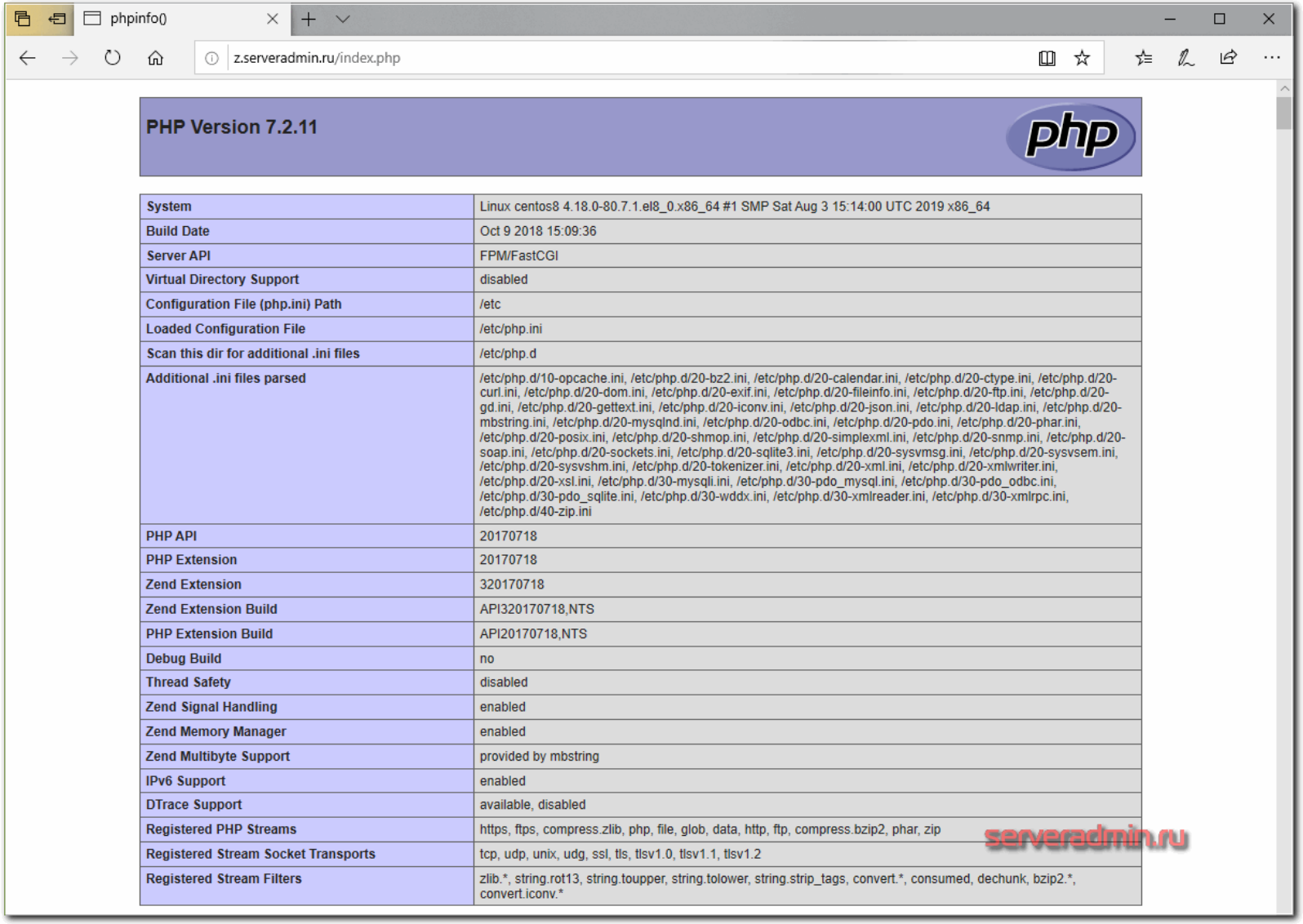
Создадим файл в директории виртуального хоста и проверим работу php:

```
# mcedit /web/sites/z.serveradmin.ru/www/index.php
```

```
<?php phpinfo(); ?>
```

```
# chown apache. /web/sites/z.serveradmin.ru/www/index.php
```

Заходим по адресу http://z.serveradmin.ru/index.php



System	Linux centos8 4.18.0-80.7.1.el8_0.x86_64 #1 SMP Sat Aug 3 15:14:00 UTC 2019 x86_64
Build Date	Oct 9 2018 15:09:36
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/10-opcache.ini, /etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-dom.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gd.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-json.ini, /etc/php.d/20-ldap.ini, /etc/php.d/20-mbstring.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-odbc.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-posix.ini, /etc/php.d/20-shmop.ini, /etc/php.d/20-simplexml.ini, /etc/php.d/20-snmp.ini, /etc/php.d/20-soap.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-sysvmsg.ini, /etc/php.d/20-sysvsem.ini, /etc/php.d/20-sysvshm.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/20-xml.ini, /etc/php.d/20-xmlwriter.ini, /etc/php.d/20-xsl.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_odbc.ini, /etc/php.d/30-pdo_sqlite.ini, /etc/php.d/30-wddx.ini, /etc/php.d/30-xmlreader.ini, /etc/php.d/30-xmlrpc.ini, /etc/php.d/40-zip.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, compress.bzip2, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, bzip2.*, convert.iconv.*

Вы должны увидеть вывод информации о php. Если что-то не так, возникли какие-то ошибки, смотрите лог ошибок виртуального хоста, php ошибки будут тоже там.

Где лежит php.ini

После установки часто возникает вопрос, а где хранятся настройки php? Традиционно они находятся в едином файле настроек. В CentOS **php.ini лежит в /etc**, прямо в корне. Там можно редактировать глобальные настройки для всех виртуальных хостов. Персональные настройки каждого сайта можно сделать отдельно в файле конфигурации виртуального хоста, который мы сделали раньше. Давайте добавим туда несколько полезных настроек:

```
# mcedit /etc/httpd/conf.d/z.serveradmin.ru.conf
```

Добавляем в самый конец, перед </VirtualHost>

```
php_admin_value date.timezone 'Europe/Moscow'
php_admin_value max_execution_time 60
php_admin_value upload_max_filesize 30M
```

Для применения настроек нужно сделать restart apache. Если у вас полностью дефолтная установка, как у меня, то скорее всего вы увидите ошибку.

Invalid command ‘php_admin_value’, perhaps misspelled or defined by a module not included in the server configuration

Суть ошибки в том, что у нас не загружен модуль mod_php. Проверим, где он подключается. Это файл */etc/httpd/conf.modules.d/15-php.conf*.

```
<IfModule !mod_php5.c>
  <IfModule prefork.c>
    LoadModule php7_module modules/libphp7.so
  </IfModule>
</IfModule>
```

Тут стоит проверка на запуск модуля. Он загружается только, если у нас загружен модуль prefork. Давайте попробуем его загрузить принудительно. Для этого комментируем все строки, кроме основной.

```
LoadModule php7_module modules/libphp7.so
```

Проверяем конфигурацию apache.

```
# apachectl -t
```

Apache is running a threaded MPM, but your PHP Module is not compiled to be threadsafe.

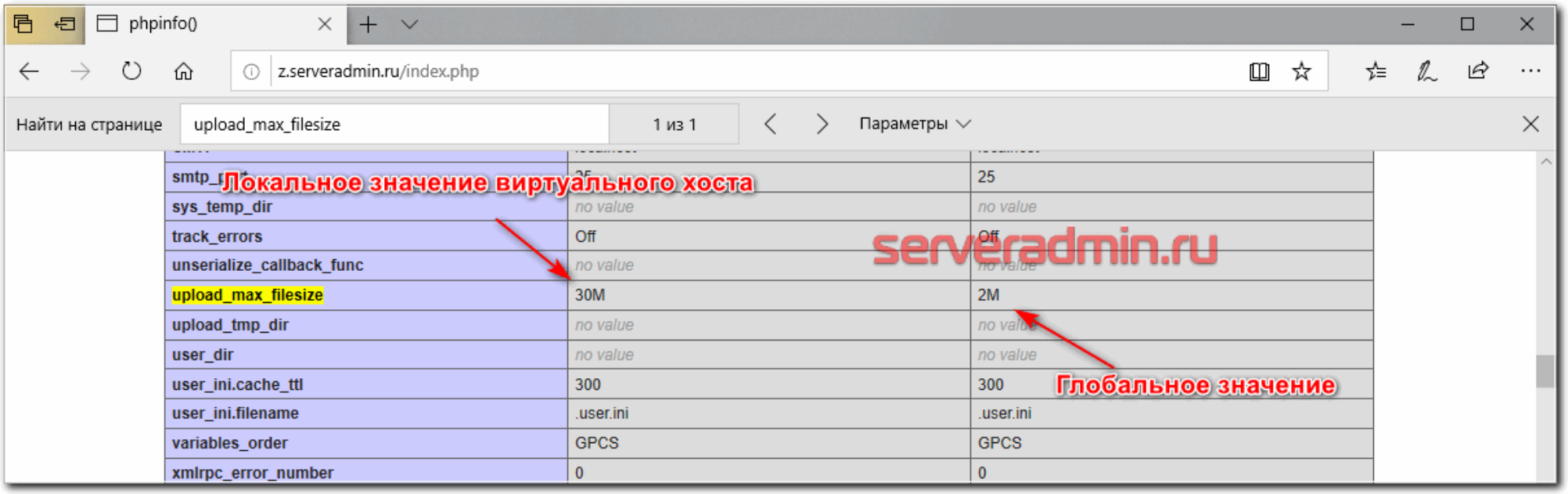
Получили новую ошибку. Смысл в том, что изначально apache сконфигурирован на работу модуля mpm_event, он подключается в конфиге */etc/httpd/conf.modules.d/00-mpm.conf.*

```
LoadModule mpm_event_module modules/mod_mpm_event.so
```

Стандартный модуль mod_php скомпилирован с поддержкой модуля mpm_prefork. С другими он работать не будет. Таким образом, чтобы у нас нормально заработал php, нам надо вместо модуля mpm_event подключить модуль mpm_prefork. Для этого в конфиге 00-mpm.conf закомментируем подключение mpm_event_module и раскомментируем prefork.

```
LoadModule mpm_prefork_module modules/mod_mpm_prefork.so
```

После этого проверяйте конфигурацию и перезапускайте apache. Все должно заработать. Теперь в выводе phpinfo можно увидеть изменение настроек.



Я подробно разобрал эти ошибки, чтобы у вас было понимание, как все устроено и куда смотреть в случае проблем. Более подробно о работе и выборе mpm модулей читайте в официальной документации apache — http://httpd.apache.org/docs/2.4/mpm.html.

Установка MySQL в CentOS 8

Как я уже писал ранее, сейчас все большее распространение получает форк mysql — mariadb. Она имеет полную совместимость с mysql, так что можно смело пользоваться. Я предпочитаю использовать именно ее.

Устанавливаем mariadb на CentOS 8:

```
# dnf install mariadb mariadb-server
```

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
```

Не буду приводить весь вывод работы этого скрипта, там все достаточно просто и понятно. Сначала задаем пароль для root (текущий пароль после установки пустой), потом удаляем анонимных пользователей, отключаем возможность подключаться root удаленно, удаляем тестового пользователя и базу.

Файлы настроек mysql/mariadb в Centos 8 лежат в директории */etc/my.cnf.d*. Для обычной работы достаточно настроек по-умолчанию. Но если вы решите изменить их, не забудьте перезапустить службу баз данных.

Перезапуск mariadb/mysql в CentOS 8:

```
# systemctl restart mariadb
```

На этом все. Базовый функционал web сервера на CentOS 8 настроен. Дальше настроим популярную панель управления mysql сервером — phpmyadmin.

Установка phpmyadmin

Для того, чтобы установить phpmyadmin на наш web сервер, достаточно просто распаковать в директорию с виртуальным хостом исходники панели. Давайте подготовим виртуальный хост. Создаем структуру папок.

```
# mkdir -p /web/sites/pma.serveradmin.ru/{www,log}
```

И готовим конфиг apache по аналогии с уже созданным доменом. Создаем файл pma.serveradmin.ru.conf.

```
# mcedit pma.serveradmin.ru.conf
```

```
<VirtualHost *:80>

    ServerName pma.serveradmin.ru
    DocumentRoot /web/sites/pma.serveradmin.ru/www

    ErrorLog /web/sites/pma.serveradmin.ru/log/error.log
    CustomLog /web/sites/pma.serveradmin.ru/log/access.log common

    <Directory /web/sites/pma.serveradmin.ru/www>
        Options FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    php_admin_value date.timezone 'Europe/Moscow'
    php_admin_value max_execution_time 360
    php_admin_value upload_max_filesize 100M

</VirtualHost>
```

После добавления конфига, не забудьте перезапустить apache.

```
# apachectl restart
```

Идем на сайт https://www.phpmyadmin.net и копируем ссылку на последнюю версию панели. Затем загружаем ее через консоль.

```
# cd ~
# wget https://files.phpmyadmin.net/phpMyAdmin/4.9.1/phpMyAdmin-4.9.1-all-languages.zip
```

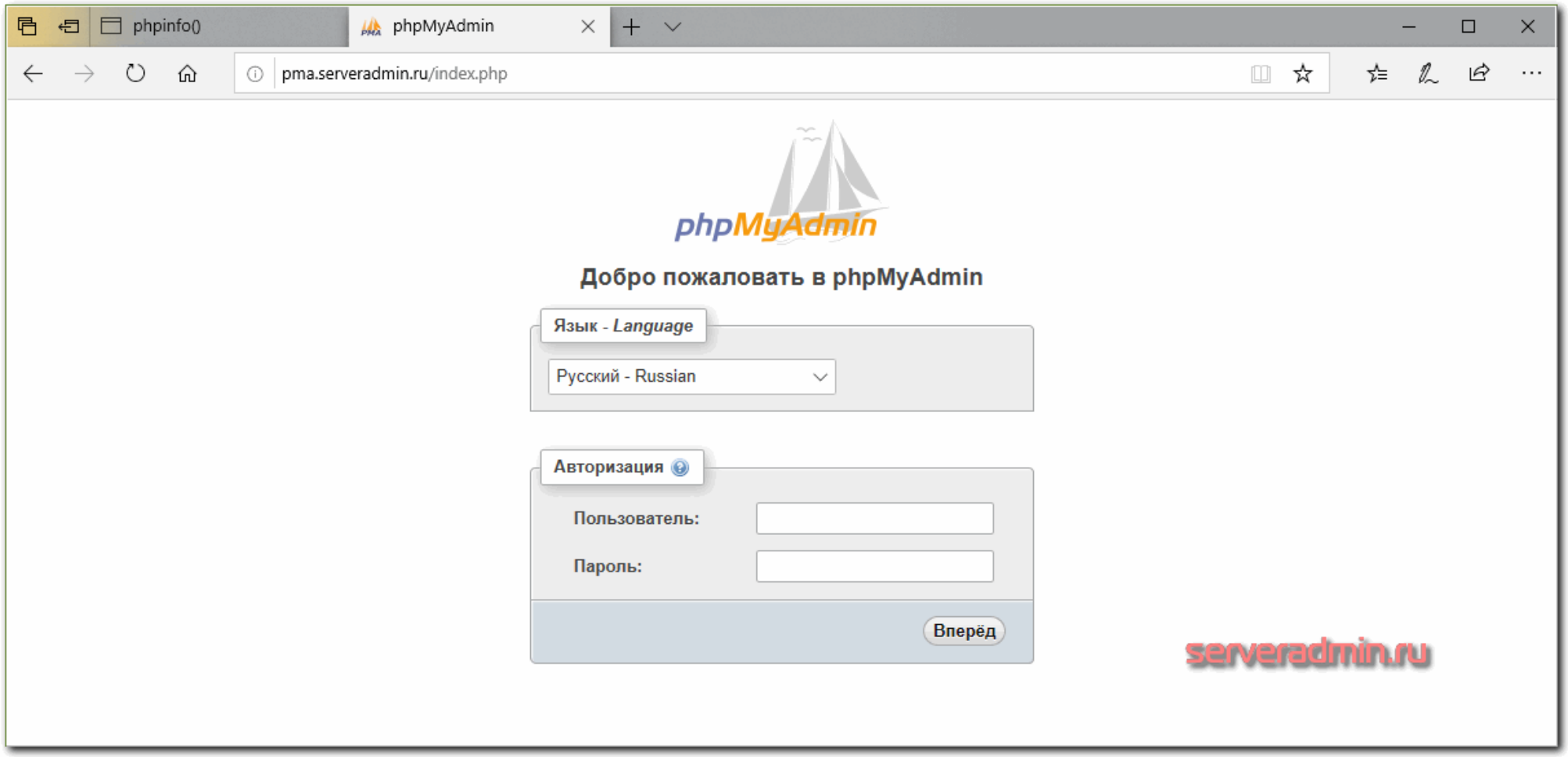
Архив упакован в zip. Если у вас нет на сервере пакета unzip, установите его.

```
# dnf install unzip
```

Распаковываем исходники в директорию виртуального хоста.

```
# unzip phpMyAdmin-4.9.1-all-languages.zip
# cp -R phpMyAdmin-4.9.1-all-languages/* /web/sites/pma.serveradmin.ru/www
# chown -R apache. /web/sites/pma.serveradmin.ru/www
```

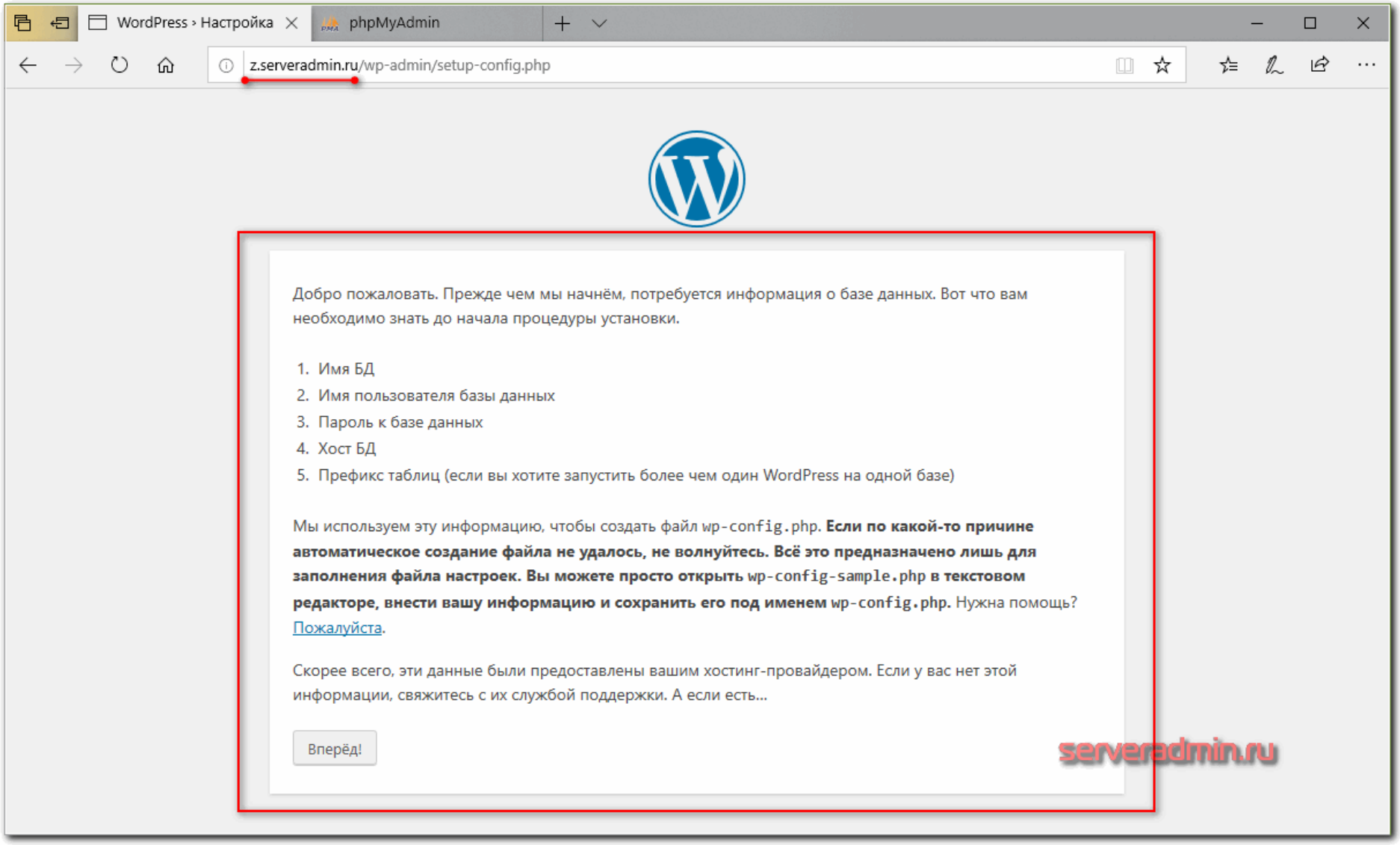
Можно заходить и проверять работу phpmyadmin, пройдя по адресу pma.serveradmin.ru. Ее установка закончена.



Более подробно вопрос установки и настройки phpmyadmin я рассматривал отдельно. Можете зайти в панель и создать базу mysql для тестового сайта, например, wordpress. Затем через консоль загрузить исходники cms и распаковать их.

```
# cd ~
# wget https://ru.wordpress.org/latest-ru_RU.tar.gz
# tar xzvf latest-ru_RU.tar.gz
# cp -R wordpress/* /web/sites/z.serveradmin.ru/www
# chown -R apache. /web/sites/z.serveradmin.ru/www
```

После этого открывайте в браузере страницу *z.serveradmin.ru* и увидите приветствие установщика wordpress.


WordPress › Настройка
phpMyAdmin
z.serveradmin.ru/wp-admin/setup-config.php
Добро пожаловать. Прежде чем мы начнём, потребуется информация о базе данных. Вот что вам необходимо знать до начала процедуры установки.
1. Имя БД
2. Имя пользователя базы данных
3. Пароль к базе данных
4. Хост БД
5. Префикс таблиц (если вы хотите запустить более чем один WordPress на одной базе)
Мы используем эту информацию, чтобы создать файл wp-config.php. Если по какой-то причине автоматическое создание файла не удалось, не волнуйтесь. Всё это предназначено лишь для заполнения файла настроек. Вы можете просто открыть wp-config-sample.php в текстовом редакторе, внести вашу информацию и сохранить его под именем wp-config.php. Нужна помощь? Пожалуйста.
Скорее всего, эти данные были предоставлены вашим хостинг-провайдером. Если у вас нет этой информации, свяжитесь с их службой поддержки. А если есть...
Вперёд!
serveradmin.ru

Настройка ssl сертификата Lets Encrypt в apache

Теперь настроим работу web сервера apache с ssl сертификатом. Хотя если быть точным, то tls сертификатом. Устанавливаем пакет **certbot** для получения бесплатного ssl сертификата от let's encrypt. В репозиториях centos 8 его пока нет, поэтому поставим вручную с сайта разрабочиков.

```
# cd ~
# wget https://dl.eff.org/certbot-auto
# mv certbot-auto /usr/local/bin/certbot-auto
# chown root /usr/local/bin/certbot-auto
# chmod 0755 /usr/local/bin/certbot-auto
```

При первом запуске certbot-auto предложит установить зависимости, которые ему нужны для работы. Он написан на python, поэтому зависимостей в виде модулей питона будет много.

```
# certbot-auto
```

```
[root@centos8 ~]# cd ~
[root@centos8 ~]# wget https://dl.eff.org/certbot-auto
--2019-10-18 12:08:21--  https://dl.eff.org/certbot-auto
Resolving dl.eff.org (dl.eff.org)... 151.101.84.201, 2a04:4e42:14::201
Connecting to dl.eff.org (dl.eff.org)|151.101.84.201|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 68699 (67K) [application/octet-stream]
Saving to: 'certbot-auto'

certbot-auto                        100%[=================================================================================>]  67.09K  --.-KB/s    in 0.05s

2019-10-18 12:08:21 (1.37 MB/s) - 'certbot-auto' saved [68699/68699]

[root@centos8 ~]# mv certbot-auto /usr/local/bin/certbot-auto
[root@centos8 ~]# chown root /usr/local/bin/certbot-auto
[root@centos8 ~]# chmod 0755 /usr/local/bin/certbot-auto
[root@centos8 ~]# certbot-auto
Bootstrapping dependencies for RedHat-based OSes that will use Python3... (you can skip this with --no-bootstrap)
dnf is /usr/bin/dnf
dnf is hashed (/usr/bin/dnf)
Last metadata expiration check: 0:00:30 ago on Fri 18 Oct 2019 12:08:13 PM MSK.
Package openssl-1:1.1.1-8.el8.x86_64 is already installed.
Package ca-certificates-2018.2.24-6.el8.noarch is already installed.
Dependencies resolved.
================================================================================================================================================
 Package                                Arch                   Version                                                 Repository                Size
================================================================================================================================================
Installing:
 gcc                                    x86_64                 8.2.1-3.5.el8                                           AppStream                 23 M
 mod_ssl                                x86_64                 1:2.4.37-12.module_el8.0.0+185+5908b0db                 AppStream                130 k
 python3-virtualenv                     noarch                 15.1.0-18.module_el8.0.0+33+0a10c0e1                    AppStream                1.7 M
 python36                               x86_64                 3.6.8-2.module_el8.0.0+33+0a10c0e1                      AppStream                 19 k
 python36-devel                         x86_64                 3.6.8-2.module_el8.0.0+33+0a10c0e1                      AppStream                 16 k
 redhat-rpm-config                      noarch                 116-1.el8.0.1                                           AppStream                 82 k
 augeas-libs                            x86_64                 1.10.1-8.el8                                            BaseOS                   392 k
 libffi-devel                           x86_64                 3.1-18.el8                                              BaseOS                    28 k
 openssl-devel                          x86_64                 1:1.1.1-8.el8                                           BaseOS                   2.3 M
Installing dependencies:
```


После установки пакетов certbot напишет ошибку, что не может сам настроить apache.

```
Complete!
Creating virtual environment...
Installing Python packages...
Installation succeeded.
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Error while running apachectl configtest.

AH00526: Syntax error on line 85 of /etc/httpd/conf.d/ssl.conf:
SSLCertificateFile: file '/etc/pki/tls/certs/localhost.crt' does not exist or is empty

Certbot doesn't know how to automatically configure the web server on this system. However, it can still get a certificate for you. Please run "certbot-auto
 certonly" to do so. You'll need to manually configure your web server to use the resulting certificate.
[root@centos8 ~]#
```


Настроим все сами. Для начала создадим самоподписанный дефолтный сертификат, чтобы apache не ругался на отсутствие файла и смог запуститься.

```
# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/pki/tls/private/localhost.key -out
/etc/ssl/certs/localhost.crt
```

Все параметры оставляйте дефолтные, не принципиально. Мы этот сертификат использовать не будет. Перезапустите apache.

```
# apachectl restart
```

Теперь выпустим сертификат для нашего домена. Имейте ввиду, чтобы получить сертификат у вас должно быть действующее доменное имя, ссылающееся на web сервер, который настраиваете. Let’s Encrypt будет по доменному имени обращаться к серверу, на котором настраиваете сертификат, чтобы проверить домен. В тестовой лаборатории с вымышленным доменным именем получить настоящий ssl сертификат не получится.

```
# certbot-auto certonly
```

```
[root@centos8 httpd]# certbot-auto certonly
Saving debug log to /var/log/letsencrypt/letsencrypt.log

How would you like to authenticate with the ACME CA?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1: Apache Web Server plugin (apache)
2: Spin up a temporary webserver (standalone)
3: Place files in webroot directory (webroot)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Select the appropriate number [1-3] then [enter] (press 'c' to cancel): 1
Plugins selected: Authenticator apache, Installer None
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel): zeroxzed@gmail.com

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must
agree in order to register with the ACME server at
https://acme-v02.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 our work
encrypting the web, EFF news, campaigns, and ways to support digital freedom.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: n
Please enter in your domain name(s) (comma and/or space separated)  (Enter 'c'
to cancel): z.serveradmin.ru
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for z.serveradmin.ru
Waiting for verification...
Cleaning up challenges

IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:
   /etc/letsencrypt/live/z.serveradmin.ru/fullchain.pem
   Your key file has been saved at:
   /etc/letsencrypt/live/z.serveradmin.ru/privkey.pem
   Your cert will expire on 2020-01-16. To obtain a new or tweaked
   version of this certificate in the future, simply run certbot-auto
   again. To non-interactively renew *all* of your certificates, run
   "certbot-auto 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
```

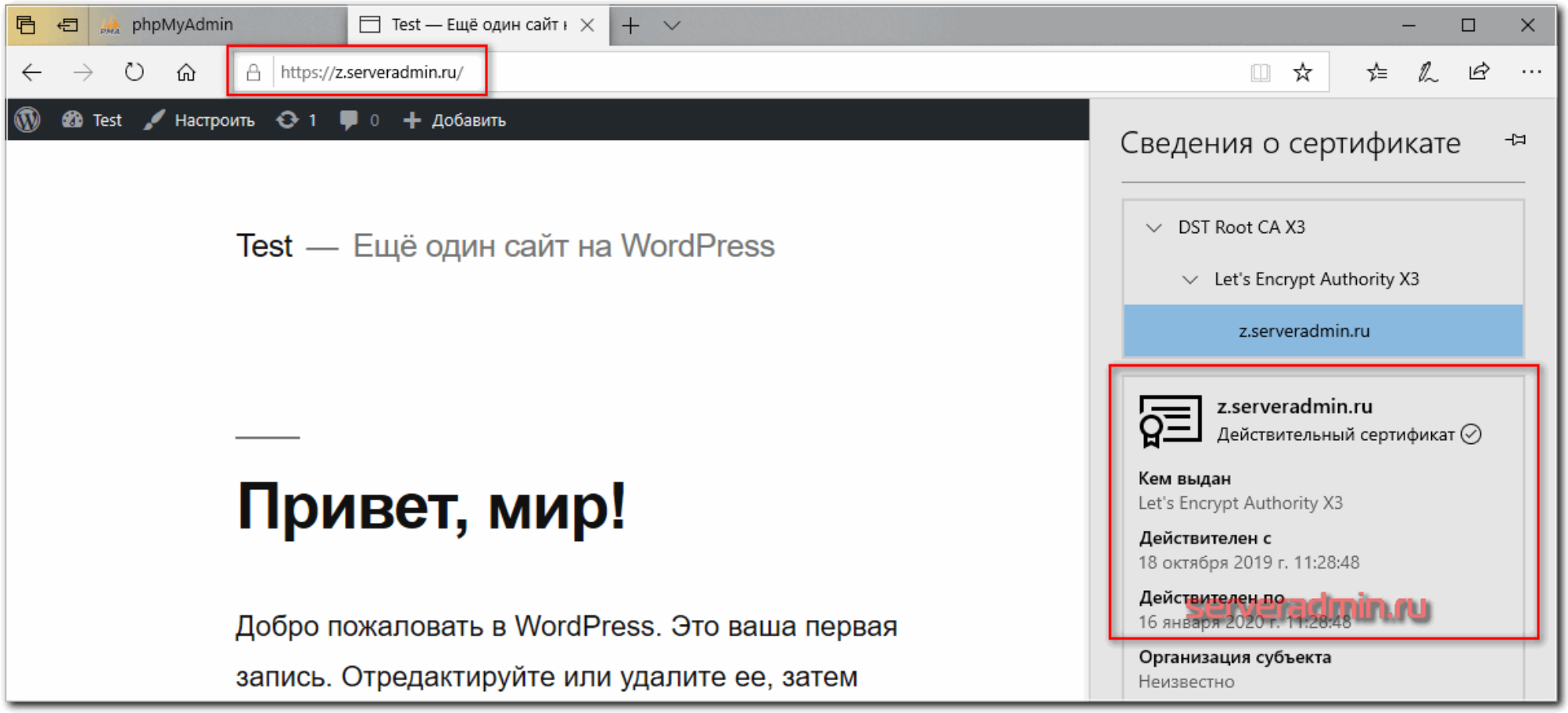
В качестве способа аутентификации выбирайте

```
1: Apache Web Server plugin (apache)
```

Дальше заполняйте в соответствии с вашими названиями. После получения сертификата, укажем его в конфигурации виртуального хоста. В моем случае в файле z.serveradmin.ru.conf. Добавляем туда параметры ssl.

```
<VirtualHost *:80 *:443>

    ServerName z.serveradmin.ru
    ServerAlias www.z.serveradmin.ru
    DocumentRoot /web/sites/z.serveradmin.ru/www

    ErrorLog /web/sites/z.serveradmin.ru/log/error.log
    CustomLog /web/sites/z.serveradmin.ru/log/access.log common

    SSLEngine on
    SSLCertificateFile /etc/letsencrypt/live/z.serveradmin.ru/fullchain.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/z.serveradmin.ru/privkey.pem

    <Directory /web/sites/z.serveradmin.ru/www>
        Options FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    php_admin_value date.timezone 'Europe/Moscow'
    php_admin_value max_execution_time 60
    php_admin_value upload_max_filesize 30M

</VirtualHost>
```

Перезапускайте apache и проверяйте работу сайта по https, зайдя по соответствующему протоколу.



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 4 * * * root /usr/local/bin/certbot-auto renew --post-hook "/usr/sbin/apachectl restart" >> /var/log/le-renew.log
```

Переадресация с http на https в apache

В настроенном ранее примере https отлично работает, но неудобно, что нет автоматической переадресации с http на https. Чтобы использовать безопасную версию сайта, необходимо вручную в браузере набирать https. Хотя все современные браузеры уже сами умеют проверять версии сайта и если есть защищенная, то они автоматически сами ее выбирают.

Тем не менее, лучше все же добавить редирект с http на https. Его можно сделать двумя различными способами:

1. Через файл .htaccess
2. С помощью настройки виртуального хоста.

Мне нравится больше второй вариант, поэтому приводим конфиг виртуального хоста к следующему виду.

```
<VirtualHost *:80>

    ServerName z.serveradmin.ru
    ServerAlias www.z.serveradmin.ru
    Redirect permanent / https://z.serveradmin.ru

</VirtualHost>

<VirtualHost *:443>

    ServerName z.serveradmin.ru
    ServerAlias www.z.serveradmin.ru
    DocumentRoot /web/sites/z.serveradmin.ru/www

    ErrorLog /web/sites/z.serveradmin.ru/log/error.log
    CustomLog /web/sites/z.serveradmin.ru/log/access.log common

    SSLEngine on
    SSLCertificateFile /etc/letsencrypt/live/z.serveradmin.ru/fullchain.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/z.serveradmin.ru/privkey.pem
```

```
    <Directory /web/sites/z.serveradmin.ru/www>
        Options FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    php_admin_value date.timezone 'Europe/Moscow'
    php_admin_value max_execution_time 60
    php_admin_value upload_max_filesize 30M

</VirtualHost>
```

Перечитывайте конфиг httpd и проверяйте. Должно работать автоматическое перенаправление на https версию.

Ротация логов веб сервера apache

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 httpd, который был создан во время установки — */etc/logrotate.d/httpd*. Приведем его к следующему виду:

```
/var/log/httpd/*log {
    missingok
    notifempty
    sharedscripts
    delaycompress
    postrotate
        /bin/systemctl reload httpd.service > /dev/null 2>/dev/null || true
    endscript
}
```

```
/web/sites/pma.serveradmin.ru/log/*log {
    size=10M
    rotate 10
    missingok
    notifempty
    compress
    sharedscripts
    postrotate
        /bin/systemctl reload httpd.service > /dev/null 2>/dev/null || true
    endscript
}

/web/sites/z.serveradmin.ru/log/*log {
    size=10M
    rotate 10
    missingok
    notifempty
    compress
    sharedscripts
    postrotate
        /bin/systemctl reload httpd.service > /dev/null 2>/dev/null || true
    endscript
}
```

Я предлагаю ротировать файлы логов по достижению ими размера в 10Мб, сжимать после ротации и хранить 10 архивов с логом. Обращаю внимание на важный нюанс при ротации логов по размеру. Скорее всего в общем случае она будет работать не так, как вы ожидаете. Подробности читайте по ссылке. Я привел пример простой конфигурации. Все параметры вы можете поменять по своему усмотрению. Примеров конфигурации logrotate в интернете много.

Настройка SELinux для web сервера apache

Раздел для тех, кто хочет настроить SELinux на своем web сервере. Сначала ставим пакет **policycoreutils-python-utils** если он еще не установлен. Он нам нужен для утилиты **semanage**.

```
# dnf install policycoreutils-python-utils
```

Теперь автоматически сформируем модуль для selinux на основе событий аудита, которые накопились, пока мы настраивали сайт. Посмотреть их можно командой.

```
# grep httpd /var/log/audit/audit.log | audit2why
```

Создаем модуль selinux.

```
# grep httpd /var/log/audit/audit.log | audit2allow -M my-httpd
```

Загружаем его.

```
# semodule -i my-httpd.pp
```

То же самое делаем для php.

```
# grep php /var/log/audit/audit.log | audit2allow -M my-php
# semodule -i my-php.pp
```

Добавим нашу директорию */web/sites* в соответствующие таблицы selinux для контента и логов.

```
# semanage fcontext -a -t httpd_sys_content_t "/web/sites(/.*)?"
# semanage fcontext -a -t httpd_log_t "/web/sites(/.*)?/log(/.*)?"
```

И отдельно добавим каталог, куда web сервер сможет писать данные. Я покажу на примере правила для сайтов wordpress, где web сервер должен уметь писать в директорию wp-content для загрузки медиафайлов, установки тем и плагинов, а так же изменять файл wp-config.php.

```
# semanage fcontext -a -t httpd_sys_rw_content_t "/web/sites/\*/www/wp-content(/.\*)?"
```

```
# semanage fcontext -a -t httpd_sys_rw_content_t "/web/sites/\*/www/wp-config.php"
```

Обновляем атрибуты файлов новым контекстом SELinux.

```
# restorecon -Rv /web/sites
```

В завершении настройки selinux для apache, добавим еще один параметр, без которого httpd не сможет писать файлы в указанные каталоги.

```
# setsebool -P httpd_unified 1
```

Теперь активируем защиту selinux и проверяем, что она работает.

```
# setenforce 1
# getenforce
Enforcing
```

Режим работы Enforcing означает, что selinux работает. Убедиться, что модули загружены, можно командой.

```
# semodule -l | grep my-
```

В целом, по selinux все. Мы просто разрешили все, что веб сервер просил. По идее, надо вдумчиво во всех правилах разбираться и разрешать только то, что считаешь нужным. Я честно скажу, что selinux знаю не очень хорошо. Дальше загрузки готовых модулей и автоматического создания модулей с помощью audit2allow я не двигался. Руками модули никогда не писал. Если есть какой-то более осмысленный и правильный способ настройки selinux на кастомной конфигурации веб сервера, буду рад полезной информации.

Хорошая практическая статья по ручной настройке selinux для web сервера — https://habr.com/ru/post/322904/. Там же есть ссылки на другие статьи автора на тему selinux. Написано содержательно и наглядно, рекомендую для тех, кто будет знакомиться с технологией.

Видео

В завершении полное видео настройки web сервера apache на основе приведенной статьи. Если у кого-то что-то не получается, посмотрите, как это сделал я.

Watch this video on YouTube

Заключение

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

На этом по настройке apache все. Я рассмотрел все основные моменты, которые необходимы для установки и настройки web сервера на основе apache и php, который обычно называют LAMP. При этом рассказал о некоторых вещах, которые повышают удобство и гибкость эксплуатации сервера.

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 или защита сайта от ddos.

Если еще что-то полезное вспомню, добавлю ссылки. Пока вроде все. Статья получилось большой и насыщенной. Было не просто все собрать воедино, проверить, связать между собой и оформить в последовательное повествование. Мог где-то ошибиться. Жду комментариев и отзывов. Написал все по своему опыту, как я обычно настраиваю веб сервера. Предполагаю что-то можно сделать более удобно и правильно. Буду рад научиться.

Напоминаю, что данная статья является частью единого цикла статьей про сервер Centos.