

# Apache

http сервер или просто веб сервер апач. Является кроссплатформенным 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.

- [Web сервер на CentOS 7](#)

# Web сервер на CentOS 7

Итак, наш веб сервер centos будет состоять из трех основных компонентов - http сервера apache, интерпретатора языка программирования php и сервера баз данных mysql.

## Настройка apache в CentOS 7

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 7 это делается очень просто:

```
yum install -y httpd
```

Добавляем apache в автозагрузку:

```
systemctl enable httpd
```

Запускаем apache в CentOS 7:

```
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://ip-address> и увидеть картинку:

[image.png](#) and or type unknown

Теперь займемся настройкой apache. Я предпочитаю следующую структуру веб хостинга:

/web	раздел для размещения сайтов
/web/site1.ru/www	директория для содержимого сайта

/web/site1.ru/logs	директория для логов сайта
--------------------	----------------------------

Создаем подобную структуру:

```
mkdir /web && mkdir /web/site1.ru && mkdir /web/site1.ru/www && mkdir /web/site1.ru/logs  
chown -R apache. /web
```

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 **site1.ru.conf**:

```
mcedit /etc/httpd/conf.d/site1.ru.conf
```

```
<VirtualHost *:80>  
    ServerName site1.ru  
    ServerAlias www.site1.ru  
    DocumentRoot /web/site1.ru/www  
    <Directory /web/site1.ru/www>  
        Options FollowSymLinks  
        AllowOverride All  
        Require all granted  
    </Directory>  
    ErrorLog /web/site1.ru/logs/error.log  
    CustomLog /web/site1.ru/logs/access.log common  
</VirtualHost>
```

## Перезапуск apache в centos

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/site1.ru/www файл index.html следующего содержания:

```
mcedit /web/site1.ru/www/index.html
```

```
<h1>Апач настроен!</h1>
```


```
chown apache. /web/site1.ru/www/index.html
```

Дальше в винде правим файл hosts, чтобы обратиться к нашему виртуальному хосту. Добавляем туда строчку:

```
192.168.1.25 site1.ru
```

где 192.168.1.25 ip адрес нашего веб сервера.

Теперь в браузере набираем адрес <http://site1.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/site1.ru/logs/error.log.

Сразу же обращаю ваше внимание на настройку ротации логов виртуальных хостов. Частенько бывает, что если сразу не настроишь, потом забываешь. Но если сайт с хорошей посещаемостью, то логи будут расти стремительно и могут занять очень много места. Лучше настроить ротацию логов веб сервера сразу же после создания. Сделать это не сложно.

Чтобы настроить ротацию логов виртуальных хостов, необходимо отредактировать файл /etc/logrotate.d/httpd. Он создается во время установки apache и включает в себя настройку ротации стандартного расположения логов. А так как мы перенесли логи каждого виртуального хоста в индивидуальную папку, необходимо добавить эти папки в этот файл:

```
mcedit /etc/logrotate.d/httpd
```

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

Мы добавили одну строку в самое начала файла. Теперь логи всех виртуальных хостов в папке /web будут ротироваться по общему правилу.

В принципе, простейший веб сервер уже готов и им можно пользоваться. Но вряд ли сейчас найдутся сайты со статическим содержимым, которым достаточно поддержки только html. Так что продолжим нашу настройку.

## Установка php в CentOS 7

Для поддержки динамического содержимого сайтов выполним следующий шаг. Установим php в CentOS 7:

```
yum install -y php
```

И следом еще несколько полезных компонентов. Установим популярные модули для php:

```
yum install -y php-mysql php-mbstring php-mcrypt php-devel php-xml php-gd
```

Выполним перезапуск apache:

```
systemctl restart httpd
```

Создадим файл в директории виртуального хоста и проверим работу php:

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

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

```
chown apache. /web/site1.ru/www/index.php
```

Заходим по адресу <http://site1.ru/index.php>

[image.png](#) image not found or type unknown

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 все виртуальных хостов. Персональные настройки каждого сайта можно сделать отдельно в файле конфигурации виртуального хоста, который мы сделали раньше. Давайте добавим туда несколько полезных настроек:

```
nano /etc/httpd/conf.d/site1.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. Теперь в выводе phpinfo можно увидеть изменение настроек.

# Обновление до php 5.6 в CentOS 7

В нашем примере мы установили на CentOS 7 php 5.4 из стандартного репозитория. А что делать, если нам нужна более новая версия, например php 5.6? В таком случае нужно выполнить обновление php.

Для этого подключим remi репозиторий:

```
wget http://rpms.remirepo.net/enterprise/remi-release-7.rpm  
rpm -Uvh remi-release-7*.rpm
```


Теперь обновляем php 5.4 до php 5.6:

```
yum --enablerepo=remi,remi-php56 install php php-common php-mysql php-mbstring php-mcrypt php-devel  
php-xml php-gd
```

Перезапускаем apache:

```
systemctl restart httpd
```

И идем смотреть вывод phpinfo - <http://site1.ru/index.php>

 image.png and or type unknown

Отлично, мы обновили php до версии 5.6.

# Установка MySQL в CentOS 7

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 7:

```
yum install -y mariadb mariadb-server
```

Добавляем mariadb в автозапуск:

```
systemctl enable mariadb.service
```

Запускаем mariadb:

```
systemctl start mariadb
```

Проверяем, запустилась или нет:

```
netstat -tulnp | grep mysqld  
tcp      0      0 0.0.0.0:3306      0.0.0.0:*        LISTEN    22276/mysqld
```

Обращаю внимание, что она даже в системе отображается как сервис `mysqld`. Теперь запускаем стандартный скрипт настройки безопасности:

```
/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.

Файл настроек `mysql/mariadb` лежит в `/etc/my.cnf`. Для обычной работы достаточно настроек по-умолчанию. Но если вы решите изменить их, не забудьте перезапустить службу баз данных.

Перезапуск `mariadb/mysql` в CentOS 7:

```
systemctl restart mariadb
```

На этом все. Базовый функционал web сервера на CentOS 7 настроен.