настройка веб-сервера CentOs x64 (Версия документа 1.0)

Кузнецов Денис (https://github.com/Kalistos/centos-web-server)

Веб-сервер Centos x64

2014

# Заказ

Позже будут приведены примеры заказа сервера на различных хостингах.

Заказываем сервер у хостинга, после чего хостинг удобным ему способом сообщает IP адрес сервера и пароль к юзеру root. С этими данными заходим на новый сервер. Можно сменить пароль, если есть желание. Это можно сделать с помощью следующей команды:

passwd

# Общие настройки сервера

## Кодировка

Кодировка сервера должна быть "UTF-8". Проверяем текущую кодировку:

locale

Будет выведен следующий результат:

LANG=

LC\_CTYPE="en\_US.UTF-8"

LC\_NUMERIC="en\_US.UTF-8"

LC\_TIME="en\_US.UTF-8"

LC\_COLLATE="en\_US.UTF-8"

LC\_MONETARY="en\_US.UTF-8"

LC\_MESSAGES="en\_US.UTF-8"

LC\_PAPER="en\_US.UTF-8"

LC\_NAME="en\_US.UTF-8"

LC\_ADDRESS="en\_US.UTF-8"

LC\_TELEPHONE="en\_US.UTF-8"

LC\_MEASUREMENT="en\_US.UTF-8"

LC\_IDENTIFICATION="en\_US.UTF-8"

LC\_ALL=en\_US.UTF-8

Этот результат говорит о том, что установлена кодировка "UTF-8". Это то, что нужно, значит дополнительных действий не потребуется.

## Время

Очень часто бывает так, что физически сервер находится в другой временной зоне, например в Америке или в Европе, и тогда местное время не будет устраивать нас, тогда нужно изменить часовой пояс.

Для этого необходимо удалить файл текущей временной зоны:

rm /etc/localtime

Далее необходимо выбрать правильную временную зону в директории /usr/share/zoneinfo, для Москвы это будет файл /usr/share/zoneinfo/Europe/Moscow и потом скопировать этот файл как текущую временную зону /etc/localtime.

cp /usr/share/zoneinfo/Europe/Moscow /etc/localtime

Далее необходимо в файле /etc/sysconfig/clock изменить параметр ZONE на необходимый часовой пояс. Для Москвы будет

ZONE="Europe/Moscow"

Теперь временная зона настроена верно.

## Репозитории

Необходимо установить дополнительные репозитории, для установки необходимых пакетов и их обновлений.

Конечный список необходимых репозиториев:

* base
* extras
* updates
* epel
* remi
* rpmforge
* atomic
* nginx

Первые три репозитория являются базовыми и поэтому подключены изначально. Остальные необходимо подключить.

Для установки одно из репозиториев нам понадобится команда wget, поэтому необходимо сначала добавить её, добавляем:

yum install wget

Далее необходимо подключить дополнительные репозитории.

Подключаем репозиторий epel:

rpm --import https://fedoraproject.org/static/0608B895.txt

rpm -ihv <http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm>

Подключаем репозиторий remi, для установки этого репозитория необходим подключенный репозиторий epel, который мы подключили выше:

rpm --import http://rpms.famillecollet.com/RPM-GPG-KEY-remi

rpm -ihv http://rpms.famillecollet.com/enterprise/remi-release-6.rpm

После установки этот репозиторий будет выключен по умолчанию, так что его необходимо будет включить. Для этого необходимо открыть файл /etc/yum.repos.d/remi.repo и в директиве [remi] необходимо изменить "enabled=0" на "enabled=1".

Подключаем репозиторий rpmforge:

rpm --import http://apt.sw.be/RPM-GPG-KEY.dag.txt

rpm -ihv <http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.3-1.el6.rf.x86_64.rpm>

Подключаем репозиторий nginx:

rpm --import http://nginx.org/keys/nginx\_signing.key

rpm -ihv <http://nginx.org/packages/centos/6/noarch/RPMS/nginx-release-centos-6-0.el6.ngx.noarch.rpm>

Подключаем репозиторий atomic:

wget -q -O - http://www.atomicorp.com/installers/atomic | sh

После установки всех репозиториев проверяем список подключенных репозиториев. Это можно сделать с помощью команды:

yum repolist

## Обновление

После подключения всех репозиториев необходимо обновить все имеющиеся пакеты. Это можно сделать с помощью следующей команды:

yum update

# Установка PHP

Устанавливаем PHP и модули к нему:

yum install php php-devel php-gd php-mbstring php-mcrypt php-xml php-pear php-pdo php-mysql

PHP с модулями установлен.

# Установка MySQL

Устанавливаем MySQL:

yum install mysql mysql-server

MySQL установлен.

## Запуск mysql

Для того, чтобы MySQL запускался сам после перезагрузки сервера необходимо запустить следующую команду:

chkconfig mysql on

Запускаем MySQL сейчас:

service mysqld start

В первый раз запуск MySQL выведет много текста, в котором написано, что надо установить пароль для супер пользователя базы данных.

## Настройка

Сразу же после установки необходимо зайти в командную строку MySQL следующей командой:

mysql -u root

Данная команда работает ровно до тех пор, пока не будет установлен пароль для супер пользователя.

Далее устанавливаем пароль для пользователя root:

SET PASSWORD FOR 'root'@'localhost' = PASSWORD('new-password');

После чего выходим из командной строки MySQL:

exit;

# Установка веб-сервера nginx

Веб-сервер nginx является очень быстрым и легким сервером, главное достоинство которого в том, что он не создает под каждый запрос отдельный процесс, поэтому при многочисленных одновременных запросах нагрузка на систему минимальная.

## Установка

Устанавливаем веб-сервер nginx:

yum install nginx

## Общая конфигурация

Конфигурационный файл nginx находится по адресу /etc/nginx/nginx.conf.

### Количество процессов

Параметр worker\_processes в конфигурационном файле определяет сколько будет процессов nginx. Их должно быть столько сколько есть ядер процессора. На недорогих серверах это число обычно равно единице. Устанавливаем:

worker\_processes 1;

### Подключение конфигурационных файлов площадок

Параметр include в конфигурационном файле определяет какие еще конфигурационные файлы должны подключиться. Необходимо подключить конфигурационные файлы площадок. Пусть они будут расположены в директории /etc/nginx/www. Этой директории изначально не существует, поэтому создадим её:

mkdir /etc/nginx/www

Далее необходимо, чтобы все конфигурационные файлы в этой папке подключались. В конфигурационном файле устанавливаем (заменяем стандартный include):

include /etc/nginx/www/\*.conf;

## Дефолтная конфигурация площадок

Для быстрого подключения новых площадок необходимы дефолтные заготовки.

Эти заготовки будем хранить в папке /etc/nginx/default. Этой папки по умолчанию не существует, так что создадим её:

mkdir /etc/nginx/default

Создаем дефолтные конфигурации nginx, которыми мы потом будем пользоваться. Допустим IP вашего сервера 192.168.0.1.

Создаем файл prod.conf в /etc/nginx/default со следующим содержимым:

server {

listen 192.168.0.1:80;

server\_name www.example.com;

rewrite ^ http://example.com$request\_uri? permanent;

}

server {

listen 192.168.0.1:80;

server\_name example.com;

error\_log /www/example.com/log/nginx-error;

access\_log /www/example.com/log/nginx-access;

location ~\* \.(jpg|jpeg|gif|png|bmp|svg|swf|htm|js|ico|css|mp3|ogg|mpe?g|gz|tar|avi|zip|gz|bz2|rar|woff|ttf|eot|doc|docx|ppt|pptx|xls|xlsx|rtf|txt|pps|ppsx|pdf|odt|ods|odp)$ {

try\_files $uri $uri/ @drupal;

expires 7d;

root /www/example.com/www;

}

location ~ /\.ht {

deny all;

}

location / {

proxy\_pass http://127.0.0.1:6081;

proxy\_set\_header Host $host;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Real-IP $remote\_addr;

}

location @drupal {

proxy\_pass http://127.0.0.1:6081;

proxy\_set\_header Host $host;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Real-IP $remote\_addr;

}

}

Далее создаем файл dev.conf в /etc/nginx/default со следующим содержимым:

server {

listen 192.168.0.1:80;

server\_name www.example.com;

rewrite ^ http://example.com$request\_uri? permanent;

}

server {

listen 192.168.0.1:80;

server\_name example.com;

satisfy any;

allow 127.0.0.1;

deny all;

allow all;

auth\_basic 'Private Area';

auth\_basic\_user\_file /www/example.com/.htpasswd;

error\_log /www/example.com/log/nginx-error;

access\_log /www/example.com/log/nginx-error;

location ~\* \.(jpg|jpeg|gif|png|bmp|svg|swf|htm|js|ico|css|mp3|ogg|mpe?g|gz|tar|avi|zip|gz|bz2|rar|woff|ttf|eot|doc|docx|ppt|pptx|xls|xlsx|rtf|txt|pps|ppsx|pdf|odt|ods|odp)$ {

try\_files $uri $uri/ @drupal;

root /www/example.com/www;

}

location ~ /\.ht {

deny all;

}

location / {

proxy\_pass http://127.0.0.1:8081;

proxy\_set\_header Host $host;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Real-IP $remote\_addr;

}

location @drupal {

proxy\_pass http://127.0.0.1:8081;

proxy\_set\_header Host $host;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Real-IP $remote\_addr;

}

}

Разница в конфигурациях заключена в следующем:

1. Для dev конфигурации добавлена http авторизация. Не зачем посторонним людям и роботам ходить на dev сайты.
2. Для dev конфигурации исключено кеширование динамического контента на уровне varnish. Для разработки кеширование Varnish только мешает.
3. Для dev конфигурации исключено кеширование статического контента на стороне клиента. Для разработки кеширование статического контента в браузере только мешает.

## Запуск nginx

Для того, чтобы nginx запускался сам после перезагрузки сервера необходимо запустить следующую команду:

chkconfig nginx on

Запускаем nginx сейчас:

service nginx start

# Установка веб-сервера apache

Веб-сервер apache является мощным сервером, главное достоинство которого в том, что он легко конфигурируемый и способен решать любые задачи, а также под этот сервер написано множество расширений, улучшающую его функциональность.

## Установка

Устанавливаем веб-сервер apache:

yum install httpd

Устанавливаем дополнительный модуль itk к apache:

yum install httpd-itk

## Общая конфигурация

Конфигурационный файл apache находится по адресу /etc/httpd/conf/httpd.conf.

### Подключение конфигурационных файлов площадок

Параметр Include в конфигурационном файле определяет какие еще конфигурационные файлы должны подключиться. Необходимо подключить конфигурационные файлы площадок. Пусть они будут расположены в директории /etc/httpd/www. Этой директории изначально не существует, поэтому создадим её:

mkdir /etc/httpd/www

Далее необходимо, чтобы все конфигурационные файлы в этой папке подключались. В конфигурационном файле дописываем в конце (не заменяем дефолтный Include):

Include www/\*.conf

### Настройка портов

Параметр Listen в конфигурационном файле определяет порт на котором будет работать сервер apache. По умолчанию стоит порт 80. Но так как у нас установлен nginx, то именно он работает на 80 порту. На одном порту не может работать несколько веб-серверов. Поэтому переводим apache на порт 8081. Для этого в конфигурационном файле устанавливаем значение:

Listen 127.0.0.1:8081

А также необходимо раскомментировать строку NameVirtualHost \*:80 (она закомментирована по умолчанию) и изменить значение на

NameVirtualHost \*:8081

### Настройка имени сервера

Параметр ServerName в конфигурационном файле определяет название сервера, так как apache расположен не на порту 80, то это локальный веб-сервер. Поэтому так его и назовём. Для этого в конфигурационном файле устанавливаем значение:

ServerName localhost

## Настройка модуля itk

Для работоспособности модуля itk необходимо открытх файл /etc/sysconfig/httpd и в самом конце дописать строку

HTTPD=/usr/sbin/httpd.itk

## Дефолтная конфигурация площадок

Для быстрого подключения новых площадок необходимы дефолтные заготовки.

Эти заготовки будем хранить в папке /etc/httpd/default. Этой папки по умолчанию не существует, так что создадим её:

mkdir /etc/httpd/default

Создаем дефолтные конфигурации apache, которыми мы потом будем пользоваться.

Создаем файл default.conf в /etc/httpd/default со следующим содержимым:

<VirtualHost \*:8081>

ServerAdmin admin@example.com

ServerName example.com

ErrorLog /www/example.com/log/httpd-error

CustomLog /www/example.com/log/httpd-access combined

DocumentRoot /www/example.com/www

<Directory /www/example.com/www>

Options FollowSymLinks

AllowOverride All

Order allow,deny

Allow from All

</Directory>

AssignUserId user user

</VirtualHost>

## Запуск apache

Для того, чтобы apache запускался сам после перезагрузки сервера необходимо запустить следующую команду:

chkconfig httpd on

Запускаем apache сейчас:

service httpd start

# Установка веб-сервера varnish

Кеширующий веб-сервер varnish сильно повышает производительность сервера, экономя аппаратные ресурсы на однотипных анонимных запросах кы страницам с динамическим контентом.

## Внимание

На данный момент этот раздел не дописан до конца.

На текущий момент в веб-сервере Varnish используются базовые настройки. И хотя по данной инструкции веб-сервер Varnish будет работать без ошибок, он не будет работать на всю свою мощность. На текущий момент те возможности, которые сейчас используются веб-сервером Varnish могут быть легко повторены на веб-сервере nginx. На текущий момент нет управления сбросом кэша и нет интеграции с друпалом, то есть тех элементов, которые и обосновывают отдельное применение веб-сервера Varnish. Данные элементы будут добавлены в самое ближайшее время.

## Установка

Устанавливаем веб-сервер varnish:

rpm --nosignature -i http://repo.varnish-cache.org/redhat/varnish-3.0/el6/noarch/varnish-release/varnish-release-3.0-1.el6.noarch.rpm

syum install varnish

## Общая конфигурация

Первый конфигурационный файл varnish находится по адресу /etc/sysconfig/varnish.

Второй конфигурационный файл varnish находится по адресу /etc/varnish/default.vcl.

### Настройка портов

Необходимо настроить параметр DAEMON\_OPTS в первом конфигурационном файле следующим образом:

DAEMON\_OPTS="-a :6081 \

-T localhost:6082 \

-f /etc/varnish/default.vcl \

-u varnish -g varnish \

-S /etc/varnish/secret \

-s file,/var/lib/varnish/varnish\_storage.bin,1G"

mkdir /etc/httpd/www

### Настройка бекенда

Необходимо настроить backend во втором конфигурационном файле следующим образом:

backend default {

.host = "127.0.0.1";

.port = "8081";

}

## Запуск varnish

Для того, чтобы varnish запускался сам после перезагрузки сервера необходимо запустить следующую команду:

chkconfig varnish on

Запускаем varnish сейчас:

service varnish start

# Установка дополнительного ПО

## Установка GIT

Устанавливаем систему контроля версий файлов GIT:

yum install git

Установка Git не описана до конца. Есть ошибка при использовании синонимов. Исправление этой ошибки будет добавлено в следующей версии.

## Установка Drush

Устанавливаем командную строку Drush:

yum install php-drush-drush

# Дополнительные настройки

## Права доступа по умолчанию

Необходимо сделать так, чтобы файлы пользователей площадок создавались с правами 750. Пользователи площадок — это пользователи с ID больше 499. Внесем изменения в необходимый файл. Код представленный ниже определяет с каким umask по умолчанию заходят пользователи. Umask определяется разницей значения 777 и желаемым значением прав для новых файлов. Так что если нам необходимо чтобы пользователи площадок по умолчанию создавали файлы с правами 750, необходимо запускать пользователя с umask 027. Это и сделаем ниже для пользователей с ID Ю 499.

Заменяем в файле /etc/bashrc

if [ $UID -gt 199 ] && [ "`id -gn`" = "`id -un`" ]; then

umask 002

else

umask 022

fi

на

if [ $UID -gt 499 ]; then

umask 027

elif [ $UID -gt 199 ] && [ "`id -gn`" = "`id -un`" ]; then

umask 002

else

umask 022

fi

## Права новых файлов

Каталог /etc/skel/ содержит файлы, которые будут по умолчанию в каталоге нового пользователя.

Изменяем права файлов в каталоге /etc/skel/ на «rw-------». Не нужно предоставлять лишний доступ к своим файлам. Даже если там ничего секретного..

Добавляем в каталог /etc/skel/ папку www с правами 750. В этой папке у пользователя будут хранится файлы сайта.

Добавляем в каталог /etc/skel/ папку backup с правами 750. В этой папке у пользователя будут хранится бекапы сайта.

Добавляем в каталог /etc/skel/ папку log с правами 750. В этой папке у пользователя будут хранится бекапы сайта.

# Площадки

## Общая информация о структуре хранения площадок

Место на сервере где мы будем создавать сайт будем называть площадками. На них могут быть размещены как сайты, так и прочие сервисы и скрипты.

Площадки хранятся в директории /www, изначально её не существует, поэтому необходимо создать её, внутри нее будут лежать сами площадки по имени сайтов и сервисов, пример:

* /www/site1.ru
* /www/site2.com

Внутри каждой площадки следующие каталоги:

* /www/site.ru/www - исполняемые файлы этого сайта
* /www/site.ru/log - логи nginx и apache для этого сайта
* /www/site.ru/backup - папка для складирования различных бекапов этого сайта

## Добавление пользователя

Для каждой площадки необходимо создавать своего собственного пользователя. То есть: один пользователь – одна площадка. Создать пользователя можно с помощью этой команды:

useradd -m -d /www/site.com user

После чего появится уникальный пользователь с уникальной группой.

## Добавление пользователя nginx в группу пользователя

Добавляем пользователя nginx в группу нового пользователя. Это необходимо для того, чтобы nginx мог отдавать статику с этой площадки:

usermod -a -G user nginx

## Добавление конфигурационного файла nginx

В зависимости от типа площадки prodaction или development копируем файл prod.conf или dev.conf из директории /etc/nginx/default в директорию /etc/nginx/www и переименовываем скопированный файл по названию веб адреса площадки. Если веб адрес площадки example.com, то переименовываем на example.com.conf.

## Добавление конфигурационного файла apache

Копируем файл default.conf из директории /etc/httpd/default в директорию /etc/httpd/www и переименовываем скопированный файл по названию веб адреса площадки. Если веб адрес площадки example.com, то переименовываем на example.com.conf.

## Создание базы данных

Необходимо зайти в командную строку MySQL следующей командой:

mysql -u root -p

После чего необходимо создать базу данных «examplecom»:

CREATE DATABASE examplecom;

После чего необходимо создать пользователя для этой базы данных «examplecom»:

GRANT ALL PRIVILEGES ON examplecom.\* TO 'examplecom'@'localhost' IDENTIFIED BY 'password' WITH GRANT OPTION;

## Результат

В результате этих действий была создана новая площадка, для которой был созданы:

1. Пользователь площадки
2. Домашняя директория площадки
3. Пользователь базы данных для площадки
4. База данных для площадки
5. Конфигурация для веб-сервера nginx
6. Конфигурация для веб-сервера apache

Теперь можно заливать свой сайт на эту площадку.

# Использованные команды

Ниже представлен список команд, использованных в документе.

### Проверка текущей кодировки сервера

locale

### Добавление команды wget

yum install wget

### Подключение репозитория epel

rpm --import https://fedoraproject.org/static/0608B895.txt

rpm -ihv http://dl.fedoraproject.org/pub/epel/6/x86\_64/epel-release-6-8.noarch.rpm

### Подключение репозитория remi

Для установки этого репозитория необходим подключенный репозиторий epel.

rpm --import http://rpms.famillecollet.com/RPM-GPG-KEY-remi

rpm -ihv http://rpms.famillecollet.com/enterprise/remi-release-6.rpm

После установки этот репозиторий будет выключен по умолчанию, так что его необходимо будет включить. Для этого необходимо открыть файл /etc/yum.repos.d/remi.repo и в директиве [remi] изменить «enabled=0» на «enabled=1».

### Подключение репозитория rpmforge

rpm --import http://apt.sw.be/RPM-GPG-KEY.dag.txt

rpm -ihv http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.3-1.el6.rf.x86\_64.rpm

### Подключение репозитория centalt

rpm --import http://centos.alt.ru/pub/repository/centos/RPM-GPG-KEY-CentALT

rpm -ihv http://centos.alt.ru/repository/centos/6/x86\_64/centalt-release-6-1.noarch.rpm

После установки этот репозиторий будет включен по умолчанию, но лучше его выключить, так как он содержит нежелательные пакеты, и использовать его только при установке некоторых пакетов. Для этого необходимо открыть файл /etc/yum.repos.d/centalt.repo и в директиве [CentALT] изменить «enabled=1» на «enabled=0».

### Подключение репозитория nginx

rpm --import http://nginx.org/keys/nginx\_signing.key

rpm -ihv <http://nginx.org/packages/centos/6/noarch/RPMS/nginx-release-centos-6-0.el6.ngx.noarch.rpm>

После установки у этого репозитория будет неправильная конфигурация (что очень странно, видимо какая-то ошибка на стороне репозитория) и установка пакетов из этого репозитория будет невозможна, поэтому необходимо поправить ее. Для этого необходимо открыть файл /etc/yum.repos.d/nginx.repo и в директиве [nginx] изменить «baseurl=http://nginx.org/packages/centos/6/$basearch/» на   
«baseurl=http://nginx.org/packages/mainline/centos/6/$basearch/».

### Подключение репозитория atomic

wget -q -O - http://www.atomicorp.com/installers/atomic | sh

rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY.art.txt

### Список подключенных репозиториев

yum repolist

### Обновление всех пакетов

yum update

### Установка веб-сервера nginx

yum install nginx

### Установка веб-сервера apache

yum install apache