# Web

- FTP
  - FTP-сервер ProFTPd на CentOS 7
- PHP
  - Установка PHP 7 и 8 на Linux CentOS 7
- NGINX
  - NGINX на CentOS 7 — установка и настройка
- Apache
  - Web сервер на CentOS 7
- Mysql
  - Установка MariaDB-server на Rocky Linux или Ubuntu

# FTP

# FTP-сервер ProFTPd на CentOS 7

## Подготовка системы

Прежде чем начать настройку нашего сервера, выполним следующие действия:

- Настроим правильное время на сервере. Это позволит нам видеть корректное значение в соответствующем атрибуте файлов и папок.
- Настроим брандмауэр. Откроем порты, которые нужны для работы ftp.
- Отключим или настроим SELinux. На выбор.

Рассмотрим эти действия подробнее.

### 1. Настройка времени

Зададим правильный часовой пояс:

```
timedatectl set-timezone Asia/Yekaterinburg
```

> * в данном примере будет задана зона по времени Екатеринбурга. Список всех доступных зон можно посмотреть командой `timedatectl list-timezones.`

Теперь установим утилиту для синхронизации времени и запустим ее в качестве сервиса:

```
yum install chrony -y
systemctl enable chronyd --now
```

### 2. Настройка брандмауэра

Нам нужно открыть порты 20, 21 для работы ftp и динамический диапазон — мы будем использовать 60000-65535.

В CentOS, как правило, используется утилита управления брандмауэром на базе firewalld, но мы также рассмотрим и iptables.

а) При использовании Firewalld:

```
firewall-cmd --permanent --add-port=20-21/tcp
firewall-cmd --permanent --add-port=60000-65535/tcp
firewall-cmd --reload
```

б) При использовании iptables:

```
iptables -I INPUT -p tcp --match multiport --dports 20,21,60000:65535 -j ACCEPT
service iptables save
```

## 3. Настройка SELinux

Для отключения SELinux вводим две команды:

```
sed -i 's/^SELINUX=.*/SELINUX=disabled/g' /etc/selinux/config
setenforce 0
```

# Установка и базовая настройка ProFTPd

Устанавливаем EPEL репозиторий:

```
yum install epel-release -y
```

Устанавливаем ProFTPd:

```
yum install proftpd -y
```

Внесем небольшие правки в начальный конфигурационный файл:

```
nano /etc/proftpd.conf
```

Добавим строки:

```
UseIPv6 off
IdentLookups off
```

```
PassivePorts 60000 65535
```

Разрешаем сервис и запускаем его:

```
systemctl enable proftpd --now
```

Можно пробовать подключаться под любой системной учетной записью.

Если необходимо добавить отдельного пользователя, вводим команду:

```
useradd ftpuser -m
passwd ftpuser
```

> * в данном примере мы создали пользователя ftpuser. Второй командой мы задали пароль.

# ProFTPd через TLS

Откроем файл:

```
nano /etc/sysconfig/proftpd
```

Зададим значение для опции **PROFTPD_OPTIONS**:

```
PROFTPD_OPTIONS="-DTLS"
```

> * опция DTLS включает TLS.

Генерируем сертификат:

```
openssl req -x509 -days 1461 -nodes -newkey rsa:2048 -sha256 -keyout /etc/pki/tls/certs/proftpd.pem -out /etc/pki/tls/certs/proftpd.pem -subj "/C=RU/ST=SPb/L=SPb/O=Global Security/OU=IT Department/CN=cdn.mylbt.ru/CN=mylbt"
```

Перезапускаем сервис:

```
systemctl restart proftpd
```

# Виртуальные пользователи

Хранить пользователей можно в файле и базе данных. Рассмотрим настройку и того, и другого.

## В файле

Устанавливаем proftpd-utils:

```
yum install proftpd-utils -y
```

Создаем каталог для хранения конфигурационных файлов froftpd:

```
mkdir /etc/proftpd.d
```

Создаем файл с паролями:

```
ftpasswd --passwd --file=/etc/proftpd.d/ftpd.passwd --name=vuser1 --uid=48 --gid=48 --home=/var/www --shell=/sbin/nologin
```

> * где **/etc/proftpd/ftpd.passwd** — полный путь до файла, в котором хранятся пользователи; **vuser1** — имя пользователя (логин); **uid** и **gid** — идентификаторы пользователя и группы системной учетной записи, от которой будет выступать сервер; **/var/www** — домашний каталог пользователя; **/sbin/nologin** — оболочка, запрещающая локальный вход пользователя в систему.

Редактируем proftpd.conf:

```
nano /etc/proftpd.conf
```

Комментируем следующую строку:

```
#AuthOrder       ...
```

Добавляем следующее:

```
RequireValidShell off
AuthUserFile /etc/proftpd.d/ftpd.passwd
AuthPAM off
LoadModule mod_auth_file.c
AuthOrder mod_auth_file.c
```

Перезапускаем сервис:

```
systemctl restart proftpd
```

Можно пробовать подключиться под созданным пользователем (в нашем примере, **vuser1**).

# В базе данных MySQL (MariaDB)

Устанавливаем компонент proftpd-mysql:

```
yum install proftpd-mysql -y
```

Если не установлена, ставим MariaDB:

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

Разрешаем и запускаем сервис:

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

Задаем пароль для суперпользователя базы данных:

```
mysqladmin -u root password
```

Подключаемся к базе данных:

```
mysql -uroot -p
```

Создаем базу данных, таблицу и пользователя:

```
> CREATE DATABASE proftpd DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
```

```
> CREATE TABLE `proftpd`.`users` (
`username` VARCHAR( 32 ) NOT NULL ,
`password` CHAR( 41 ) NOT NULL ,
`uid` INT NOT NULL ,
`gid` INT NOT NULL ,
`homedir` VARCHAR( 255 ) NOT NULL ,
`shell` VARCHAR( 255 ) NOT NULL DEFAULT '/sbin/nologin',
UNIQUE (`username`)
```

```
) ENGINE = MYISAM CHARACTER SET utf8 COLLATE utf8_general_ci;
```

```
    > GRANT SELECT ON proftpd.* TO proftpu@localhost IDENTIFIED BY 'proftpdpass';
```

> * данными командами мы создали базу данных **proftpd**. В ней таблицу **users** и пользователя **proftpu** с паролем **proftpdpass**, которому дали право подключаться только с локального сервера.

Добавляем пользователя в таблицу и отключаемся от базы:

```
    > INSERT INTO `proftpd`.`users` VALUES ('sqluser1', ENCRYPT('sqlpassword'), '48', '48', '/var/www',
'/sbin/nologin');
> \q
```

> * в данном примере мы создаем пользователя **sqluser1** с паролем **sqlpassword**.

Создаем файл с конфигурацией для SQL:

```
    mkdir /etc/proftpd.d
```

```
    nano /etc/proftpd.d/sql.conf
```

```
    SQLBackend mysql
SQLEngine on
SQLAuthTypes Crypt
SQLConnectInfo proftpd@localhost proftpu proftpdpass
SQLUserInfo users username password uid gid homedir shell
SQLAuthenticate users*
SQLMinUserUID 33
SQLMinUserGID 33
SQLLogFile /var/log/proftpd/sql.log
```

Настраиваем proftpd (добавляем строки):

```
    nano /etc/proftpd.conf
```

```
    LoadModule mod_sql.c
LoadModule mod_sql_mysql.c
Include /etc/proftpd.d/sql.conf
```

```
AuthOrder mod_sql.c
```

Перезапускаем сервис:

```
systemctl restart proftpd
```

# Возможные ошибки

## 1. A certificate in the chain was signed using an insecure algorithm

Ошибку можно увидеть в некоторых FTP-клиентах при попытке подключиться к серверу, который использует SSL.

**Причина**: как и следует из сообщения, клиенту не нравится алгоритм шифрования, так как он устарел и не соответствуем требованиям безопасности.

**Решение**: необходимо использовать сертификат с более стойким алгоритмом шифрования, например sha256. Чтобы получить такой сертификат с помощью утилиты openssl нужно добавить ключ **-sha256** (в инструкции выше используется именно такой подход).

## 2. lib permission denied

При попытке работы с каталогом lib, сервис выдает ошибку **permission denied**.

**Причина**: при использовании chroot в proftpd используется модуль RLimitChroot, который запрещает работу с «чувствительными» каталогами — lib, etc и так далее.

**Решение**: если у нас есть необходимость постоянно работать с каталогами, защищенными модулем RLimitChroot, то его действие можно отключить.

Открываем конфигурационный файл:

```
nano /etc/proftpd.conf
```

Добавляем строку:

```
RLimitChroot off
```

Чтобы настройка применилась, перезапускаем сервис:

```
systemctl restart proftpd
```

# PHP

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.

PHP

# Установка PHP 7 и 8 на Linux CentOS 7

## Добавление репозиториев и установка

Первый репозиторий, который мы добавим — EPEL, второй на выбор — либо REMI, либо webtatic.

### Epel

Установка выполняется командой:

```
yum install epel-release -y
```

### REMI

Добавляем репозиторий:

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

По умолчанию, репозитории для разных версий php отключены. Мы должны сами определить версию устанавливаемой php.

Посмотреть список доступных для включения версий можно командой:

```
ls /etc/yum.repos.d/remi-*
```

Включить репозиторий для нужной версии можно командой:

```
yum-config-manager --enable remi-php74
```

или, например:

```
yum-config-manager --enable remi-php81
```

* в данном примере мы будем устанавливать php версии **7.4** или **8.1**.

> Если мы получим ошибку **yum-config-manager: command not found**, устанавливаем yum-utils:
>
> ```
> yum install yum-utils -y
> ```

Устанавливаем php:

```
yum 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 -y
```

Или (без включения репозитория командой yum-config-manager):

```
yum --enablerepo=remi-php74 install php -y
```

## Webtatic

Устанавливаем репозиторий на webtatic.com:

```
rpm -Uvh https://mirror.webtatic.com/yum/el7/webtatic-release.rpm
```

Теперь устанавливаем php7:

```
yum --enablerepo=webtatic install php72w
```

* в данном примере мы поставим php версии 7.2.

# Проверка

Проверить рабочую версию php можно следующей командой:

```
php -v
```

или:

```
php -r "phpinfo();" | grep "PHP Version"
```

# Установка расширений

Если мы включили репозиторий с помощью yum-config-manager, то установка расширений выполняется обычной командой, например:

```
yum install php-mysqli
```

Иначе, расширения для нужных версий php устанавливаем с указанием репозитория, например:

```
yum --enablerepo=remi-php74 install php-mysqli
```

... или:

```
yum --enablerepo=webtatic install php-mysqli
```

# Downgrade

При обновлении PHP на более новую версию проблем не возникает, и мы можем его выполнять установкой (yum install). Для понижения версии php необходимо использовать yum downgrade:

```
yum --enablerepo=remi-php73 downgrade php php-*
```

> * обратите внимание, что мы для **downgrade** указываем **php** и все пакеты, которые начинаются на **php-** (все расширения). В противном случае, мы получим ошибку зависимостей.

# NGINX

программный продукт для развертывания веб-сервера или веб-прокси (http, mail). Получил большое распространение благодаря простоте настройки и скорости работы. Название — производное от engine x и по-русски произносится «энджинкс» или «энжин-иикс».

Может использоваться как:

Балансировщик сетевых запросов.
Независимый полноценный веб-сервер.
Прокси для почтовых протоколов — smtp, imap.
Фронтенд http с переадресацией веб-запросов на другие серверы (с помощью настройки proxy_pass). Сам, при этом, может обрабатывать часть запросов.
По сравнению с Apache, работает быстрее при отдаче статики, а также потребляем меньше ресурсов сервера. Apache, в свою очередь, совместим с большинством программ и гибок благодаря модульной системе. Очень часто, администраторы устанавливают оба веб-сервера одновременно — NGINX принимает запросы, отдает статику и перенаправляет обработку скриптов Apache. Такой подход позволяет извлечь плюсы обоих систем.

NGINX хорошо документирован. На официальном сайте можно найти документацию по установке, настройке и поддержке приложения. Инструкция полная и может использоваться как чайниками, так и опытными администраторами nginx.

# NGINX на CentOS 7 — установка и настройка

Для получения последней версии NGINX создаем файл с настройками нового репозитория:

```
nano /etc/yum.repos.d/nginx.repo
```

И приводим его к следующему виду:

```
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/centos/$releasever/$basearch/
gpgcheck=0
enabled=1
```

Обновляем систему и список пакетов:

```
yum update
```

> если система запросит подтверждение, отвечаем Y.

Устанавливаем NGINX следующей командой:

```
yum install nginx
```

По умолчанию, в CentOS работает брандмауэр, поэтому необходимо добавить рабочие порты 80 (http) и 443 (https) в правила на исключение:

```
firewall-cmd --permanent --add-port=80/tcp
firewall-cmd --permanent --add-port=443/tcp
```

> в данном примере добавлены порты **80** и **443**. Первый используется в NGINX по умолчанию. Если планируется использовать другой, необходимо добавить его. Или

> наоборот — если не запланирована работа по безопасному протоколу https, можно его не включать.

Перезапускаем firewalld:

```
firewall-cmd --reload
```

Теперь добавляем NGINX в автозапуск при загрузке CentOS

```
systemctl enable nginx
```

И запускаем веб-сервер:

```
systemctl start nginx
```

Для проверки запустите браузер на другом компьютере и введите в адресную строку IP-адрес сервера, который был настроен. Должна загрузиться тестовая страница, например:

image.png
Image not found or type unknown

Тестовая страница правильно настроенного сервера NGINX

# NGINX + PHP + PHP-FPM

В чистом виде, веб-сервер NGINX используется редко. Настроим связку с PHP и его обработчиком — PHP-FPM.

Для начала, устанавливаем тот и другой следующими командами:

```
yum install php
yum install php-fpm
```

Разрешаем автозапуск php-fpm и запускаем его:

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

**Готово.**

# Настройка NGINX для работы с PHP и PHP-FPM

Открываем настройки сайта по умолчанию:

```
nano /etc/nginx/conf.d/default.conf
```

Редактируем секцию location:

```
location / {
root   /usr/share/nginx/html;
index  index.php;
}
```

> здесь мы поменяли **index.html** на **index.php**. Эта настройка позволит автоматически искать и запускать файл **index.php**, если путь к скрипту не указан явно.

Приводим к следующему виду секцию server:

```
location ~ \.php$ {
set $root_path /usr/share/nginx/html;
fastcgi_pass 127.0.0.1:9000;
fastcgi_index index.php;
fastcgi_param SCRIPT_FILENAME $root_path$fastcgi_script_name;
include fastcgi_params;
fastcgi_param DOCUMENT_ROOT $root_path;
}
```

> где **/usr/share/nginx/html** — корневой путь по умолчанию для хранения сайта; **9000** — порт, на котором работает **php-fpm**.

Переименовываем индексный файл для текущего сайта по умолчанию:

```
mv /usr/share/nginx/html/index.html /usr/share/nginx/html/index.php
```

Отредактируем его (содержимое заменим на это):

```
nano /usr/share/nginx/html/index.php
```

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

Перезапускаем nginx:

```
systemctl restart nginx
```

Открываем сайт — мы должны увидеть сводную информацию по серверу и работе PHP.

# Совместная работа NGINX и PHP-FPM через сокет

В конфигурационном файле NGINX меняем эту строчку:

```
fastcgi_pass 127.0.0.1:9000;
```

На эту:

```
fastcgi_pass unix:/var/run/php-fpm/php5-fpm.sock;
```

Открываем настройки php-fpm:

```
nano /etc/php-fpm.d/www.conf
```

Меняем:

```
listen = 127.0.0.1:9000
```

На:

```
listen = /var/run/php-fpm/php5-fpm.sock
```

Перезапускаем nginx и php-fpm:

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

# 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.

# 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
Image not found 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:

image.png
Image not found or type unknown

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 обращу 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
 endscript
}
```

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
Image not found 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.

# Mysql

Mysql - система управления базами данных. Завоевала свою популярность в среде малых и средних приложений, которых очень много в вебе. 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.

# Установка MariaDB-server на Rocky Linux или Ubuntu

> В данной инструкции установка MariaDB будет выполнена на Rocky Linux/CentOS и Ubuntu. Также мы немного расскажем о настройке СУБД после выполнения развертывания.

## Установка и запуск

Рассмотрим по отдельности для разных систем процесс установки сервера MariaDB и настройки автозапуска. Мы выполним развертывания из репозиториев. Это не требует дополнительных действий, но не позволит установить самые последние версии СУБД. Об использовании репозитория разработчика будет рассказано ниже.

### CentOS / Rocky Linux

Для CentOS 7 и 8 или Rocky Linux 8 порядок действий не отличается. Устанавливаем MariaDB следующей командой:

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

Разрешаем автозапуск демона и запускаем его:

```
systemctl enable mariadb --now
```

### Ubuntu

Устанавливаем MariaDB следующей командой:

```
apt install mariadb-server
```

Разрешаем автозапуск демона (запускать не нужно, так как в Ubuntu это происходит на автомате):

```
systemctl enable mariadb
```

# После установки

Независимо от того, на какой Linux мы установили MariaDB, выполняем следующие действия.

1. Установим пароль для основной учетной записи СУБД:

```
mysqladmin -u root password
```

> система запросит новый пароль. Его нужно ввести дважды.

2. Проверим, что сервер работает, подключившись к нему:

```
mysql -uroot -p
```

> будет запрошен пароль. Введите тот, который был установлен на предыдущем шаге инструкции.

Если появилось приглашение командной строки

```
MariaDB [(none)]>
```

**... значит сервер установлен и работает.**

3. При необходимости, настройка сервера выполняется в файле **/etc/my.cnf** и подключаемых файлах в каталоге **/etc/my.cnf.d/**. После внесения изменений не забываем перезапустить службу:

```
systemctl restart mariadb
```

4. Если предполагается удаленное подключение к СУБД, добавляем правило в брандмауэр.

## Firewalld:

```
    firewall-cmd --permanent --add-port=3306/tcp
firewall-cmd --reload
```

### Iptables:

```
    iptables -I INPUT -p tcp --dport 3306 -j ACCEPT
apt install iptables-persistent
netfilter-persistent save
```

# Подключение актуального репозитория

В примерах выше мы установили СУБД из репозиториев операционных систем. А значит — нет гарантии, что будет установлена последняя версия MariaDB. Для решения задачи мы можем подключить официальный репозиторий самого разработчика. Для этого переходим по ссылке [downloads.mariadb.org/mariadb/repositories](downloads.mariadb.org/mariadb/repositories) и выбираем нашу версию операционной системы, последний стабильный релиз mariadb и геолокацию репозитория, например:

image.png
Image not found or type unknown

Ниже появится инструкция по добавлению репозитория и установке СУБД:

image.png
Image not found or type unknown

Согласно инструкции, добавим репозиторий.

a) Для RPM (Rocky Linux / CentOS):

```
    nano /etc/yum.repos.d/mariadb.repo
```

Вставляем строки, которые мы увидели в инструкции на сайте:

```
    [mariadb]
name = MariaDB
# rpm.mariadb.org is a dynamic mirror if your preferred mirror goes offline. See https://mariadb.org/mirrorbits/
for details.
# baseurl = https://rpm.mariadb.org/10.11/centos/$releasever/$basearch
baseurl = https://mirror.docker.ru/mariadb/yum/10.11/centos/$releasever/$basearch
module_hotfixes = 1
# gpgkey = https://rpm.mariadb.org/RPM-GPG-KEY-MariaDB
```

```
gpgkey = https://mirror.docker.ru/mariadb/yum/RPM-GPG-KEY-MariaDB
gpgcheck = 1
```

б) Для DEB (Debian / Ubuntu / Astra Linux):

```
    apt install apt-transport-https curl
curl -o /etc/apt/trusted.gpg.d/mariadb_release_signing_key.asc
'https://mariadb.org/mariadb_release_signing_key.asc'
echo 'deb https://mirrors.xtom.ee/mariadb/repo/10.5/ubuntu focal main' >> /etc/apt/sources.list.d/mariadb.list
apt update
```

После настройки репозитория обновляем список пакетов и устанавливаем СУБД.

В зависимости от типа Linux команды будут разные.

а) Rocky Linux, CentOS:

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

б) Ubuntu:

```
    apt update
apt install mariadb-server
```