

## rtorrent+rutorrent+nginx+php-fpm. Подводные камни

[Предыдущая статья](#) про связку rtorrent+rutorrent+nginx+php-fpm была написана сразу после успешной установки и начальной настройки этой связки. В процессе эксплуатации выявились некоторые подводные камни, о которых я и хочу рассказать.

### RPC

Как оказалось, чтобы связка nginx с rutorrent+rtorrent работала нормально, достаточно в файле **conf/config.php** в дереве каталогов установки rutorrent прописать такие строчки:

```
$scgi_port = 33333;  
  
$scgi_host = "127.0.0.1";
```

При этом в файле настроек rtorrent **~/.rtorrent.rc** достаточно прописать сетевой сокет:

```
scgi_port = 127.0.0.1:33333
```

В nginx больше не нужно писать backend для /RPC2, а из скриптов запуска rtorrent нужно убрать работу с локальным UNIX-сокетом для управления rtorrent.

### Права доступа

Естественно, для того, чтобы rutorrent успешно работал, нужно пользователю, от имени которого работают демоны nginx и php-fpm, дать права на все файлы и каталоги установки rutorrent. В моём случае это делает команда:

```
sudo chown -R http:http /srv/http/nginx/rutorrent.eternity/htdocs
```

## Сокет php-fpm

При локальном использовании php-fpm лучше посадить на UNIX-сокеты. Для этого в файле **/etc/php/php-fpm.conf** необходимо закомментировать строку с сетевым сокетом:

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

И вписать ниже строку с UNIX-сокетом:

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

После этого необходимо перенастроить nginx. В файле **/etc/nginx/conf/nginx.conf** приводим блок backend'a к такому виду:

```
upstream backend {  
server unix:/var/run/php-fpm/php-fpm.sock;  
}
```

Использование UNIX-сокетов уменьшает нагрузку на систему, так как не вычисляются контрольные суммы, а поток данных направляется непосредственно в принимающий буфер. Критично только для встраиваемых решений.

## Безопасность

При локальном использовании nginx лучше всего посадить на 127.0.0.1. Для этого в файле **/etc/nginx/conf/sites-enabled/rutorrent.eternity** строку **listen** следует привести

к такому виду:

```
listen 127.0.0.1:80;
```

Если этого не сделать, веб-интерфейсом сможет воспользоваться тот, кто знает IP-адрес компьютера. Чтобы иметь возможность удалённо управлять rtorrent'ом, следует позаботиться об аутентификации (хотя бы через базовую аутентификацию nginx).

## geoip

Для того, чтобы заработал модуль rtorrent под названием geoip, необходимо установить соответствующее расширение для PHP. В моём случае (*Arch Linux*) это действие выглядит так:

```
sudo pacman -S php-geoip
```

Потом в файле **/etc/php/conf.d/geoip.ini** необходимо снять комментарий с единственной строки, чтобы она выглядела так:

```
extension=geoip.so
```

Также необходимо активировать плагин JSON. Делается это созданием в каталоге **/etc/php/conf.d** файла **json.ini** с таким содержимым:

```
extension=json.so
```

**Без плагина json ничего работать не будет.**

## Вспомогательные программы

Для того, чтобы rutorrent нашёл дополнительные программы (*curl*, *stat*, *mediainfo*), во-первых, их нужно установить:

```
sudo pacman -S curl mediainfo
```

Во-вторых, необходимо разрешить их выполнение. Для этого в файле **/etc/php/php.ini** нужно закомментировать строчку `open_basedir`:

```
;open_basedir = /srv/http:/home:/tmp:/usr/share/pear/
```

И, в-третьих, требуется дополнительные программы прописать в конфигурации rutorrent. CURL и stat прописываются в файле **conf/config.php** дерева каталогов установки rutorrent таким образом:

```
$pathToExternals = array(  
    "php" => "",  
    "curl" => '/usr/bin/curl',  
    "gzip" => "",  
    "id" => "",  
    "stat" => '/usr/bin/stat',  
);
```

Путь к программе *mediainfo* указывается в файле **plugins/mediainfo/conf.php** дерева каталогов установки rutorrent таким образом:

```
$pathToExternals['mediainfo'] = '/usr/bin/mediainfo';
```

Эти пути можно и не указывать, если для пользователя, от имени которого работают демоны веб-части связки, установлена переменная окружения PATH.

## Вкусняшки

Лично для себя я сделал скрипт с таким содержимым:

```
#!/usr/bin/env bash

delay="1000"
pid1=`pidof rtorrent`

if [[ $pid1 != "" ]]
then
notify-send -t $delay "Останавливаю rtorrentd..."
sudo rc.d stop rtorrentd
notify-send -t $delay "rtorrentd остановлен"
else
notify-send -t $delay "Запускаю rtorrentd..."
sudo rc.d start rtorrentd
notify-send -t $delay "rtorrentd запущен"
fi
```

Этому скрипту даются права на выполнение:

```
sudo chown root:root x-rtorrentd-wrapper.sh
sudo chmod 755 x-rtorrentd-wrapper.sh
```

А сам скрипт перемещается в **/usr/bin**:

```
sudo mv x-rtorrentd-wrapper.sh /usr/bin
```

Потом где-то на панельке своего DE (я всё делал в xfce) вешается кнопка на запуск этого скрипта. Если rtorrent не запущен, то скрипт его запустит, а если запущен, то выключит, при этом выводя всплывающее оповещение на экран через libnotify. Конечно же, для этого нужно установить libnotify:

```
sudo pacman -S libnotify
```

### Перезапуск демонов

После выполнения всех перечисленных действий необходимо перезапустить php-fpm и nginx:

```
sudo rc.d restart php-fpm nginx
```

В итоге шансы, что rutorrent полноценно заработает, существенно увеличиваются.

**UPDATE 1:** добавил некоторые объяснения.

**UPDATE 2:** обновил раздел с RPC. Спасибо [svin0](#).

pfactum, вчера в 16:00