

Ошибка NGINX – 24: Too Many Open Files – и её решение

Описание проблемы

Во процессе обслуживания высокого количества соединений (высокая нагрузка), в логе ошибок Вашего сервера могут появляться записи 24: Too Many Open Files. Происходит это потому что сервер nginx пытается открыть больше файловых дескрипторов чем ему позволено. Ограничение может быть наложено конфигурацией nginx сервера или конфигурацией операционной системы. На каждое соединение nginx открывает как минимум два дескриптора, один для соединения, второй для передаваемого файла.

Диагностика

Для выяснение причин проведем короткую диагностику. Прежде всего посмотрим конфигурационный файл nginx, обычно он расположен в

```
/etc/nginx/nginx.conf
```

Нас интересуют следующие строки

```
user [xxx];
worker_processes [x];
worker_rlimit_nofile [xxx];
events {
    worker_connections [xxxx];
}
```

`user` – от имени какого пользователя nginx работает в системе, нам необходимо будет проверить лимиты для данного пользователя.

`worker_processes` – кол-во процессов nginx работающих при

запуске демона, так как лимит операционной системы общий для всех процессов, соответственно нам надо будет устанавливать лимиты исходя из кол-ва процессов.

`worker_rlimit_nofile` – максимальное кол-во файловых дескрипторов на один процесс.

`worker_connections` (может присутствовать только в контексте `events`) – максимальное кол-во соединений на один процесс.

Далее выясним лимиты установленные операционной системой, для этого перейдем под пользователя `nginx`

```
su nginx
```

Если получаем от OS сообщение вида

```
This account is currently not available.
```

Значит в файле `/etc/passwd` для нашего пользователя указана оболочка запрещающая вход в систему и работу с консолью, например `/sbin/nologin`

Отредактируем файл `/etc/passwd`, заменим строку

```
nginx:x:500:500:nginx user:/var/cache/nginx:/sbin/nologin
```

на строку

```
nginx:x:500:500:nginx user:/var/cache/nginx:/bin/bash
```

и вновь выполним `su nginx`.

Теперь посмотрим лимиты:

```
ulimit -Hn
```

```
ulimit -Sn
```

`-Hn` – жесткий лимит максимального кол-ва открытых файловых дескрипторов

`-Sn` – мягкий лимит максимального кол-ва открытых файловых дескрипторов

Отличие между жестким и мягким лимитами в том, что жесткий может быть установлен только root пользователем, а мягкий может быть установлен пользователем на которого наложен лимит, но не больше чем жесткий лимит.

Теперь мы знаем текущие ограничения OS наложенные на пользователя nginx, можно выйти из под su

```
exit
```

Посмотрим глобальное ограничение на количество дескрипторов, для этого выполним команду

```
sysctl fs.file-max
```

Итак мы знаем все ограничения, теперь попробуем сделать так чтобы наш nginx получил возможность открывать столько дескрипторов, сколько ему нужно.

Решение проблемы

Посчитаем сколько нам нужно дескрипторов для nginx

$(worker_connections * 4 * worker_processes) + 100 =$ примерно
нужное кол-во дескрипторов

Соответственно для будущего расширения это значение можно умножить еще на два и задавать лимиты.

Например, $(1024 * 4 * 2) + 100 = 8292$

Значение не большое, можно его свободно увеличить до 25000

Устанавливаем глобальные лимиты операционной системы, в файле /etc/sysctl.conf прописываем или изменяем строку:

```
fs.file-max = 203463
```

Устанавливаем лимиты на пользователя, в файле /etc/security/limits.conf прописываем или изменяем строки:

```
nginx soft nofile 25000
```

```
nginx hard nfile 50000
```

Применяем установленные лимиты к системе

```
sysctl -p
```

Проверяем что все применилось

```
sysctl fs.file-max
```

```
su nginx
```

```
ulimit -Hn
```

```
ulimit -Sn
```

```
exit
```

Устанавливаем лимиты для nginx

```
worker_rlimit_nfile 12500
```

Перезагружаем конфигурацию nginx

```
nginx -s reload
```

Ошибка более не должна повторяться.

Возвращаем shell для nginx в файле /etc/passwd в исходное состояние

```
nginx:x:500:500:nginx user:/var/cache/nginx:/sbin/nologin
```

<https://devmems.wordpress.com/tag/nginx/>