Disable Apache access.log and error.log logging on Debian Linux and FreeBSD

Author : admin

Many times disabling logging on a busy websites is quite beneficial, especially if more than few Gigabytes are written in Apache visitors log **(access.log)** every day. Too much visitors to Apache webserver could pose significantly increase disk writes and be negative for overall server performance.

Disabling the log is handy also for websites which already integrate a different type of visitors logging lets say - via MySQL, PostgreSQL (SQL) ...

From security perspective disabling logging is a very stupid idea thought, however on systems which are experiencing high load and you need to sacrifice logging to reduce a bit the load (especially if you cannot afford to get a new server hardware), disabling it is an option.

1. Disabling access.log and error on Debian Linux

a) Disabling access.log logging
As most Debian users already know on *Debian GNU Linux* Apache logs all incoming (port 80) Apache requests to **/var/log/apache2/access.log** and **/var/log/apache2/error.log**

Disabling logging is very simple, just comment out line in **/etc/apache2/sites-enabled/000-default**:

CustomLog ${APACHE_LOG_DIR}/access.log combined

to

#CustomLog ${APACHE_LOG_DIR}/access.log combined

Then restart the webserver to re-read new config value:

/etc/init.d/apache2 restart
....

Of course this is one of the ways to disable access.log logging. Other ways are to make logging gets logged in good old */dev/null*. To use /dev/null forwardingp put *Customlog /dev/null* in */etc/apache2/sites-enabled/000-default*

CustomLog /dev/null

In Debian Lenny and older Debian releases *Customlog* Apache directive is found in **/etc/apache2/apache2.conf**.

b) Disabling error.log logging

Same procedure applies for disabling *error.log*, comment out default **ErrorLog** directive, restart Apache and you're done:

ErrorLog ${APACHE_LOG_DIR}/error.log

should become:

ErrorLog /dev/null

Usually just comming *ErrorLog ${APACHE_LOG_DIR}/error.log* is supposed to work, unfortunately for reason on Debian Squeeze this worked not commenting it and restarting Apache failed to restart apache with error:

/etc/init.d/apache2 restart
Restarting web server: apache2 ... waiting (2)No such file or directory: apache2: could not open error log file /etc/apache2/logs/error_log.
Unable to open logs
Action 'start' failed.
The Apache error log may have more information.
failed!

Thus to disalbe error.log you need to add *ErrorLog /dev/null* in /etc/apache2/apache2.conf and once again restart Apache.

ErrorLog /dev/null

```
# /etc/init.d/apache2 restart
```

Bear in mind that if you use some custom virtualhosts which has the *ErrorLog* directive in (let's say /etc/apache2/sites-enabled/{website-domain.com,website-domain1.com} etc. you need to change there too.

2. Disabling access.log and error.log logging on FreeBSD

On FreeBSD to disable access.log add *CustomLog /dev/null* to **/usr/local/etc/httpd.conf** and just like on Linux restart Apache:

```
freebsd# /usr/local/etc/rc.d/apache2 restart
....
```

Disaling error.log on BSD is done by changing:

```
ErrorLog /var/log/httpd-error.log
```

to

```
ErrorLog /dev/null
```

BTW disaling **error.log** is quite a stupid idea but in some situation, where you don't update software versions and don't change often webserver script interpreter and (processed) server side executables / PHP scripts it could be ok.

Still it is much better to change the amount of Apache logged information and keep error.log logging by changing:

```
LogLevel crit
```

Using *LogLevel crit*, will prevent Apache from logging numerous not so useless warnings in *error.log*, so if you have a very busy server with high loads you better use it.

Don't expect that disabling logging will drastically improve performance usually even on Apache servers which serve more than 20 000 of requests daily disabling access.log / error.log could would probably reduce load with from 00.1 to maximum 2-3 percentage.