

4.2 Apache Log Management

This article can still be improved by adding a longer term larger solution for multi-site hosting for the error logs and a logging rotation solution if we are not using Ubuntu.

Strategy

By default Apache just keeps its main log files in `/var/log/apache2/` as single files, **access.log** - all incoming requests and responses
error.log - any errors

other_vhosts_access.log - access log for VirtualHosts that don't define their own logfile

This is fine for casual servers. However, as you get more serious the following will happen,

1. Traffic Increases - your single log files start becoming too big to look through.
2. It Gets Harder to Distinguish Sites - if you are hosting more than one site, it becomes hard to distinguish what is what.

To solve this we need to do log management focusing on two things,

1. Enable automatic roll over of logs.
2. Split the log files by virtual hosts.

Ubuntu and Apache

If you use apt-get to install Apache on Ubuntu log management is already happening by default. This is done by [logrotate](#). Settings can be found in `/etc/logrotate.d/apache2` and the defaults are,

- Logs roll over to a new log each week.
- Logs roll over if they reach size x.
- Old log is gzipped.

If someone looks up size x please share it here.

Distinguishing Between Virtual Hosts

The ubuntu logrotate is great but what about distinguishing between different virtual hosts.

There are two approaches I am aware of and I personally only have figured out approach 1 where each virtual host has its own set of log files which I call "Virtual Host Based Logging". Further information is at the [Apache2 Log Files documentation](#).

1. **Virtual Host Based Logging**
 - a. Pro - Simple to understand. You use a packaged Apache program to rotate the logs.
 - b. Pro - If you drop these files into the `/etc/logrotate.d/apache2` with .log extensions they automatically rotate (provided you are using Ubuntu).
 - c. Con - Not feasible when you have too many (how many is this?) as you run out of [file descriptors](#).
2. **Use a single file and split among virtual hosts.**
 - a. You use an external program to both rotate and split the logs.
 - b. Con - The technique used for access.log is not available for the error log (to reference why here) so is it really that good?

Virtual Host Based Logging

This approach is already outlined as part of my [Apache Setup article](#).

Split and Rotate Logs with External Program

Contenders

There does not seem to be a clear winner in this department. Here is what I looked at that appear to be popular via Google searches.

logrotate - use what Ubuntu uses. Some possible references,

<http://blog.andrewbeacock.com/2008/02/how-to-use-logrotate-with-apaches.html> - talks about using logrotate with Tomcat without requiring a restart to rollover log file.

Vlogger - Similar (and intended as a replacement for) **cronolog** or **httplog**. This looks really easy to use, creates symlinks and can hook into a database. On the negative side, the package seemed last updated in 2005.

Cronolog - the recommended solution from the [Apache2 Log Files documentation](#) but it was last updated in 2002. The biggest negative, I could not in 5 minutes and never bothered to go further on figuring out how to pipe for rollover **and** how to split at the same time. The problem is that during roll over, the cronolog actually stops logging for a short time before writing to a new file.

References

1. [Apache Log Files documentation](#)
2. Webmin
3. ...