By Falko Timme

Published: 2008-08-07 19:37

How To Set Up WebDAV With Lighttpd On Debian Etch

Version 1.0
 Author: Falko Timme <ft [at] falkotimme [dot] com>
Last edited 07/24/2008

This guide explains how to set up WebDAV with lighttpd on a Debian Etch server. WebDAV stands for ***Web-based Distributed Authoring and Versioning*** and is a set of extensions to the HTTP protocol that allow users to directly edit files on the lighttpd server so that they do not need to be downloaded/uploaded via FTP. Of course, WebDAV can also be used to upload and download files.

I do not issue any guarantee that this will work for you!

1 Preliminary Note

I'm using a Debian Etch server with the IP address *192.168.0.100* here.

2 Installing WebDAV

You can install lighttpd (if it's not already installed), the lighttpd WebDAV module and the *apache2-utils* package (which contains the tool *htpasswd* which we will need later on to generate a password file for the WebDAV share) as follows:

```
apt-get install lighttpd lighttpd-mod-webdav apache2-utils
```

Afterwards, create the directory */var/run/lighttpd* and make it owned by the *www-data* user and group. This directory will contain an SQLite database needed by WebDAV:

```
mkdir /var/run/lighttpd/
```

```
chown www-data:www-data /var/run/lighttpd/
```

Next, we enable the module *mod_auth*:

```
lighty-enable-mod auth
```

... and open */etc/lighttpd/lighttpd.conf* to make sure that the modules *mod_alias* and *mod_webdav* are enabled in the *server.modules* stanza:

```
vi /etc/lighttpd/lighttpd.conf
```

```
[...]
server.modules              = (
            "mod_access",
            "mod_alias",
            "mod_accesslog",
#           "mod_rewrite",
#           "mod_redirect",
#           "mod_status",
#           "mod_evhost",
#           "mod_compress",
#           "mod_usertrack",
#           "mod_rrdtool",
            "mod_webdav",
#           "mod_expire",
#           "mod_flv_streaming",
#           "mod_evasive"
 )
[...]
```

Restart lighttpd afterwards:

```
/etc/init.d/lighttpd restart
```

3 Creating A Virtual Host

I will now create a lighttpd vhost (*www.example.com*) in the directory */var/www/web1/web*. If you already have a vhost for which you'd like to enable WebDAV, you must adjust this tutorial to your situation.

First, we create the directory */var/www/web1/web* and make the lighttpd user (*www-data*) the owner of that directory:

```
mkdir -p /var/www/web1/web

chown www-data:www-data /var/www/web1/web
```

Then we open */etc/lighttpd/lighttpd.conf* and add the following vhost to the end of the file:

```
vi /etc/lighttpd/lighttpd.conf
```

```
[...]
$HTTP["host"] == "www.example.com" {
  server.document-root = "/var/www/web1/web"
}
```

Afterwards we restart lighttpd:

```
/etc/init.d/lighttpd restart
```

4 Configure The Virtual Host For WebDAV

Now we create the WebDAV password file `/var/www/web1/passwd.dav` with the user `test` (the `-c` switch creates the file if it does not exist):

```
htpasswd -c /var/www/web1/passwd.dav test
```

You will be asked to type in a password for the user `test`.

(Please don't use the `-c` switch if `/var/www/web1/passwd.dav` is already existing because this will recreate the file from scratch, meaning you lose all users in that file!)

Now we change the permissions of the `/var/www/web1/passwd.dav` file so that only `root` and the members of the `www-data` group can access it:

```
chown root:www-data /var/www/web1/passwd.dav

chmod 640 /var/www/web1/passwd.dav
```

Now we modify our vhost in `/etc/lighttpd/lighttpd.conf` so that it looks as follows:

```
vi /etc/lighttpd/lighttpd.conf
```

```
$HTTP["host"] == "www.example.com" {
  server.document-root = "/var/www/web1/web"
  alias.url = ( "/webdav" => "/var/www/web1/web" )
  $HTTP["url"] =~ "^/webdav($|/)" {
    webdav.activate = "enable"
    webdav.is-readonly = "disable"
    webdav.sqlite-db-name = "/var/run/lighttpd/lighttpd.webdav_lock.db"
    auth.backend = "htpasswd"
    auth.backend.htpasswd.userfile = "/var/www/web1/passwd.dav"
    auth.require = ( "" => ( "method" => "basic",
                             "realm" => "webdav",
                             "require" => "valid-user" ) )
  }
}
```

The `alias.url` directive makes (together with `$HTTP["url"] =~ "^/webdav($|/)"`) that when you call `/webdav`, WebDAV is invoked, but you can still access the whole document root of the vhost. All other URLs of that vhost are still "normal" HTTP.

Restart lighttpd afterwards:

```
/etc/init.d/lighttpd restart
```

5 Testing WebDAV

We will now install `cadaver`, a command-line WebDAV client:

```
apt-get install cadaver
```

To test if WebDAV works, type:

```
cadaver http://www.example.com/webdav/
```

You should be prompted for a user name. Type in `test` and then the password for the user `test`. If all goes well, you should be granted access which means WebDAV is working ok. Type `quit` to leave the WebDAV shell:

```
server1:~# cadaver http://www.example.com/webdav/
  Authentication required for webdav on server `www.example.com':
  Username: test
  Password:
  dav:/webdav/> quit
  Connection to `www.example.com' closed.
server1:~#
```

6 Configure A Windows XP Client To Connect To The WebDAV Share

This is described on http://www.howtoforge.com/setting-up-webdav-with-apache2-on-debian-etch-p2.

Please specify the port in the WebDAV URL, e.g. `http://www.example.com:80/webdav`. For some strange reason this makes Windows XP accept the normal username (e.g. `test`) - otherwise Windows XP expects NTLM usernames (that would have the form `www.example.comtest`).

7 Configure A Linux Client (GNOME) To Connect To The WebDAV Share

This is described on http://www.howtoforge.com/setting-up-webdav-with-apache2-on-debian-etch-p3.

8 Links

- WebDAV: http://www.webdav.org
- Lighttpd: http://www.lighttpd.net
- Debian: http://www.debian.org