

Configuration

How much time do you have to set up lighttpd? 5 minutes? 10 minutes? More?

5 minutes

Want to run a fast, low-resource server for static content? It's easy. Create a text file named `lighttpd.conf` with the following content:

```
server.document-root = "/var/www/servers/www.example.org/pages/"

server.port = 3000

mimetype.assign = (
    ".html" => "text/html",
    ".txt"  => "text/plain",
    ".jpg"  => "image/jpeg",
    ".png"  => "image/png"
)
```

lighttpd will listen on TCP port 3000 and bind to all interfaces by default. The few important MIME types are assigned and the document root (the base directory that is used for all requests) is set. The files in the document root have to be readable by the user starting the web server.

First, check that your configuration is ok:

```
$ lighttpd -t -f lighttpd.conf
```

Now start the server for testing:

```
$ lighttpd -D -f lighttpd.conf
```

and point your browser to <http://127.0.0.1:3000/>

To stop the server, return to the command prompt and press ctrl-c.

A real daemon

Next you should familiarize yourself with some settings necessary for your server's security:

```
server.document-root = "/var/www/servers/www.example.org/pages/"

server.port = 80

server.username = "www"
server.groupname = "www"

mimetype.assign = (
    ".html" => "text/html",
    ".txt"  => "text/plain",
    ".jpg"  => "image/jpeg",
    ".png"  => "image/png"
)

static-file.exclude-extensions = ( ".fcgi", ".php", ".rb", "~", ".inc" )
index-file.names = ( "index.html" )
```

Now the web server is listening on port 80, the default for HTTP traffic, and will switch to the user `www` and the group `www`. The server has to be started as root to take control of port 80, but it's not necessary or a good idea to continue running as root after port acquisition, so the server switches to user `www`.

Lastly, access to view the contents of some types of files is forbidden as they are used for generating dynamic content. Requests directly to a directory are rewritten to the `index.html` file in that directory.

Assuming you have already created the `/etc/init.d/lighttpd` service as described in [TutorialInstallation](#), place the config file in `/etc/lighttpd/lighttpd.conf` and start the server with:

```
# /etc/init.d/lighttpd start
```

To stop it use:

```
# /etc/init.d/lighttpd stop
```

10 minutes

Conditionals, conditionals, conditionals:

The most important part in Lighty's configuration is the use of conditionals. Using simple or regular expression conditions, default setting can be overridden.

```
server.document-root = "/var/www/servers/www.example.org/pages/"

server.port = 80

server.username = "www"
server.groupname = "www"

mimetype.assign = (
    ".html" => "text/html",
    ".txt" => "text/plain",
    ".jpg" => "image/jpeg",
    ".png" => "image/png"
)

static-file.exclude-extensions = ( ".fcgi", ".php", ".rb", "~", ".inc" )
index-file.names = ( "index.html" )

$HTTP["host"] == "www2.example.org" {
    server.document-root = "/var/www/servers/www2.example.org/pages/"
}
```

Now we have a new virtual server, `www2.example.org`, which uses the same settings as the first server, only the document root is different.

The following server configuration adds a download area and enables the built-in directory listing feature:

```
server.document-root = "/var/www/servers/www.example.org/pages/"

server.port = 80

server.username = "www"
server.groupname = "www"

mimetype.assign = (
    ".html" => "text/html",
    ".txt" => "text/plain",
    ".jpg" => "image/jpeg",
    ".png" => "image/png"
)

static-file.exclude-extensions = ( ".fcgi", ".php", ".rb", "~", ".inc" )
index-file.names = ( "index.html" )

$HTTP["host"] == "www2.example.org" {
    server.document-root = "/var/www/servers/www2.example.org/pages/"
    $HTTP["url"] =~ "^/download/" {
        dir-listing.activate = "enable"
    }
}
```

As you can see, conditionals can be nested: only the download folder and its sub-folders have the directory listings enabled.

There's also the `else` clause for conditionals. Despite the name, it's an `else if` construct similar to some programming languages, as it has to be followed by another condition.

Here's an example of conditional-based vhosts. The `else` is being used to configure behavior that should be present only in "default" vhost.

```
$HTTP["host"] == "example.org" {
    # options specific to example.org
    expire.url = ( "" => "access plus 25 hours" )
} else $HTTP["host"] == "static.example.org" {
    # options specific to static.example.org
    expire.url = ( "" => "access plus 2 weeks" )
} else $HTTP["host"] =~ "" {
    # options applied to any other vhosts present on this ip
    # ie. default options
    expire.url = ( "" => "access plus 3 hours" )
}
```

Now that we've covered the basics, you're ready to learn some more advanced topics like [includes](#) and configuring PHP with [FastCGI](#)

After 30 minutes

Now you know the basic set up, include files, and maybe even how to set up PHP or Ruby. But there's so much more to find out.

Most of it is described in examples in the default configuration file that you can find in the doc directory of the tarball or in the repository (<source:trunk/doc/lighttpd.conf>).

For the **configuration file syntax** and the complete **list of configuration options**, visit the reference section of the wiki: [Reference Documentation](#)

Also check out our [community](#).

[lighttpd.conf](#)  (11.2 KB) Anonymous, 2006-09-18 21:34