

Raspberry Pi Web Server with FTP access (update after 9 years).

I bought my first Raspberry Pi in 2012. One of my first Raspberry Pi projects was building a web server to host a single static website. That server has been running with the same hardware and system-software for almost 9 years, 24/7, with almost no interruption. The only failures have been peripherals – an SD memory card, and a power supply.

The site itself has been much expanded and regularly updated, entirely by 'live' updating using FTP. It was time for a rebuild in 2021, with a newer Raspberry Pi (Version 3) and the latest Pi OS software.

Looking back, I had built my first website (for business) in 2004, just about when adsl 'broadband' was being adopted and the internet was becoming established for those who had desktop computers. Huge expansion had taken place by 2012. In setting up a Raspberry Pi server I took the view that a very simple setup for a single static website would meet my needs, and indeed that has proved true. The site has become quite large, with some 250 pages, nearing 500Mb. Holding the master files on a Macbook and uploading with Filezilla has been easy. Hence my approach to the new configuration has not changed much. Details are as follows.

1. Set static address. The configuration file is now `/etc/dhcpd.conf` (changed since my 2012 configuration) and appropriate entries are made under static IP configuration.
2. After adding a user (`<username>`), create directories. Create `/home/<username>/ftp` (to be the ftp root) and `/home/<username>/ftp/files` (to contain the site. Deny writeability to the first, using terminal command **`sudo chmod a-w /home/<username>/ftp`**.
3. Edit the vsftpd configuration file `/etc/vsftpd.conf` to include the following directives **`anonymous_enable=NO ; local_enable=YES ; write_enable=YES ; chroot local_user = YES ; local_root=/home/<username>/ftp`**
note Don't use the directive `allow_writeable_chroot=YES` !
4. Test ftp e.g. with Filezilla
5. Make some changes in the nginx file `/etc/nginx/sites-enabled/default`. First, the root definition within the server block should be **`root /home/<username>/ftp/files`**. Next, consider the browser cache settings appropriate to the website; files which are liable to editing without name change should not be cached. So insert **`$expires $expires;`** after the root statement and then insert a 'map' for \$expires such as the following which caches all images but does not cache html files.

```
map $sent_http_content_type $expires {  
    default      off;  
    text/html    epoch;  
    ~image/     max;  
}
```

Place this just ahead of the server block.

6. Insert `<username>` in the first line of the `etc/nginx/nginx.conf` file
7. Load the site (or an index.html placeholder) into `/home/<username>/ftp/files` and test.

Notes A. It is wise to save a disc image regularly; 'Apple Pi Baker' makes this easy, and achieves resizing, which can otherwise be necessary and perhaps tricky.

B. Cache settings (para 5 above) are not mentioned in a number of other web pages about Raspberry Pi web servers. The default is to cache all files. If that is not changed, then repeat visitors to the site may not see updates unless they reload.