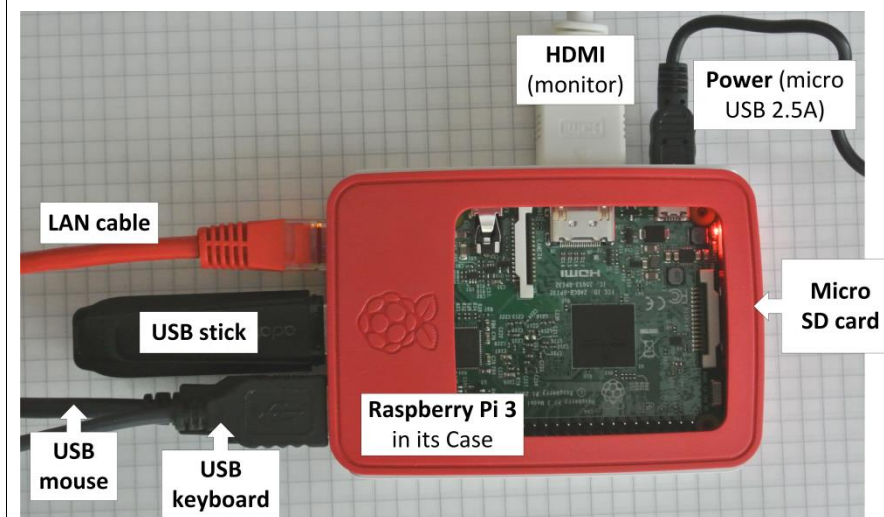


# Raspberry Pi 3: Apache2 webserver running websites off a USB stick

This is what you need:

Raspberry Pi 3	8GB micro SD-card + adapter	2.5A micro USB power supply
Monitor with HDMI connection	USB keyboard	USB mouse
HDMI-cable (male-male)	A network cable for connecting Raspberry Pi 3 to your router	
An Internet connection with static ip-address		



## A. Set up your Raspberry Pi 3

There is a wealth of information on the Internet. A good place to start is [www.raspberrypi.org](http://www.raspberrypi.org). Follow these points:

1. Download NOOBS from <https://www.raspberrypi.org/downloads/> to your computer and unzip the file to an empty folder
2. Insert the micro SD card in your computer (put the card in an adapter first). The SD card must be FAT32 formatted. (Check with Windows Explorer). If necessary, format the SD card.
3. Copy the unzipped NOOBS files from your computer to the SD card.
4. Better safe than sorry: now is the time to learn how to back-up your SD card. Google 'backup SD card', download program and follow instructions. (A good example of software is Win32DiskManager.)
5. Connect all peripherals, EXCEPT the POWER, see figure 1
6. Connect the power
7. In a few seconds you will see the Raspberry Pi desktop

## B. Set up your router

Usually computers, printers, etc, on a Local Area Network (LAN) are given ip-addresses by the router dynamically. We will change this for the Raspberry and give it a static ip-address.

1. Log on to your router
2. Under *device info*, or similar, you will see your Raspberry Pi as *raspberrypi*
3. Check your router's manual for how to make a *DHCP reservation*. For my D-Link DIR-655 I find this on my router at *Network settings, set up, DHCP reservations*.
4. Pick a free ip-address from the range that your router use and reserve that ip-address for your *raspberrypi*. From now on that is the ip-address for raspberrypi on your LAN. Verify by looking at device info, status. You should see raspberrypi there with the ip-address you gave it.
5. Make a note of the ip-address since you may want to use that later for connecting to the raspberrypi from other computers on your LAN.

6. You will want Internet conversations to go directly to your server only. Therefore set up *port forwarding*. I find this at *port forwarding, advanced*. Tick the box, fill in program name, ip, and set ports open to 80,443.

## C. Install Apache2

Open the command window by clicking on (a small black screen at the upper left corner) the command line write:



```
sudo apt-get install apache2 -y
```

Apache is by default configured to use directories `/var/www/html/` for website files. The apache2 files are in `/etc/apache2`. Use the Raspbian Filemanger (icon of a file cabinet, in the upper left corner of the screen) to browse the directory structure.

## D. Configure Apache2

Before we start, let's give the USB stick a short name, I used USBFD (for USB Flash Disk).

1. Connect the USB stick to your windows computer
2. In windows explorer right-click the USB stick and rename it to your taste
3. Reconnect the USB stick to Raspberry Pi

Linux is very particular about rights. To the system, Apache2 is a user named `www-data` that has no right to use the USB stick. To change this, we start by finding the USB id on the system. The USB stick is in directory `/media/pi`.

4. To find the USB stick's id on the system, change directory (the letters `ls -l` are small L, not the digit 1):

```
cd /dev/disk/by-label/
ls -l
```

This yields a table where I see that my USB stick, in my case called USBFD, is linked to file `sda1`.

5. Change to directory `/dev/by-uuid` and then print the files in that directory to screen:

```
cd /dev/disk/by-uuid
ls -l
```

I can see in the table that on my system `DC87-E3B0` is linked to `sda1` (an arrow  $\rightarrow$  pointing at `./././sda1`). This is the id of the USB stick. We will now give `www-data` (the Apache2 server program) rights on the USB stick, by adding a command line to `/etc/fstab`, a file that is loaded every time Raspberry Pi boots.

6. Change to directory `/etc` by typing

```
cd /etc
```

7. Open file `fstab` (filesystem table) in editor nano, by typing:

```
sudo nano fstab
```

8. At the end of the file, add this as one line (there is no space between `www-` and `data`)(exchange `DC87-E3B0` and `USBFD` for what is applicable in your case):

```
UUID=DC87-E3B0 /media/pi/USBDFD vfat notfail,uid=www-data,gid=www-data 0 0
```

9. Press `Ctrl+X`, `Y` and `Enter` to save the file. This will tell the operating system to mount the USB drive with a specific id to the same directory every time, and give ownership to the user `www-data`.

## E. Set up virtual servers

Although you only have one ip-address, you can still host several websites on your apache2 server. Because each website has a unique domain name and apache2 can be configured to handle virtual servers. Open apache2 configurations file:

```
cd /etc/apache2
sudo nano apache2.conf
```

Towards the end of the file, find the line `<Directory /var/www/>` and exchange that for `<Directory /media/pi/USBFD/>`. Now apache2 will find your USB stick. Now we shall create configuration files for your websites. I demonstrate for one website only.

Let's assume your domain name is `ronincircle.com` and that the site is located on your USB stick at `/etc/media/pi/RoninCircle`. Create file `ronincircle.com.conf` and open it in the `nano` editor by writing:

```
cd /etc/apache2/sites-available
sudo nano ronincircle.com.conf
```

While still in the editor write:

```
<VirtualHost *:80>
    DocumentRoot "/media/pi/USBFD/RoninCircle"
    ServerName www.ronincircle.com
    ServerAlias ronincircle.com
    <Directory /y>
        Options FollowSymLinks
    </Directory>
    Errorlog $/APACHE_LOG_DIR/error.log
    CustomLog $/APACHE_LOG_DIR/access.log combined
</VirtualHost>
```

Then save and exit (`Ctrl+X`, `Y`). Repeat for every website. Naming the file after the domain name of the site.

We now have all configuration files in `/etc/apache2/sites-available`. By using a tool called `a2ensite` we will create links to these files in `/etc/apache2/sites-enabled`.

```
cd /etc/apache2/sites-available
sudo a2ensite ronincircle.com.conf
```

Repeat `a2ensite` for every website. Now reboot raspberrypi.

```
sudo reboot
```

If Raspberry Pi should fail to boot, disconnect the power, remove the SD card and load the backed up copy onto your SD card using the appropriate software.

### Useful commands:

```
restart apache2:    sudo /etc/init.d/apache2 restart
shut down Raspberry Pi:  sudo poweroff
```

**Tip 1:** Raspberry has ssh activated by default. Download Putty.exe to your windows computer and you can control Raspberry from your windows machine. Nice.

**Tip 2:** Website user statistics is always interesting. For that purpose you can download and install for example `webalizer`.