

MoodleBox building reference*

Version 1.9.3

Nicolas Martignoni

December 28, 2021

Contents

1	Introduction	1
2	Raspberry Pi preparation	2
3	Wireless access point (AP) feature	6
4	Web server and database installation (LEMP stack)	11
5	Moodle installation and configuration	14
6	Accès depuis Moodle aux fichiers d'une clef USB ou déposés par SFTP	17
7	Configurations additionnelles de Moodle	18
8	Configuration de Adminer	19
9	Optimisation	20
10	Redimensionnement automatique de la partition	22
11	Nettoyage de la distribution	23
12	Acknowledgements	24

1 Introduction

This article documents how to build a MoodleBox. It is intended for developers or system administrators to provide background information on how a MoodleBox is built.

*This work is licensed under a [Creative Commons Attribution - Non Commercial - Share Alike 4.0 International](https://creativecommons.org/licenses/by-nc-sa/4.0/) license.

This is not meant to be an end-user documentation. For end-user documentation, please consult [MoodleBox website](#).

The actual building of MoodleBox is now done using [Ansible](#). Using Ansible enables most of the build to be automated, as well as ensuring that it is reproducible. Though complete, the instructions in this document should not be understood as a method to obtain a MoodleBox that is totally equivalent to the officially published MoodleBox images.

What MoodleBox does

- Configurable wireless access point with SSID: *MoodleBox* and password: *moodlebox*.
- DHCP server for wireless clients.
- Moodle server (<http://moodlebox.home/>). This is a totally standard Moodle installation.
- MoodleBox Moodle administration plugin¹ providing a GUI to manage most aspects of the MoodleBox.
- Internet access: If MoodleBox is connected via ethernet or wireless LAN to an Internet-connected network, it acts as a router and gives its wireless clients access to the Internet.

Specific Moodle features

- Moodle version 3.11.x in its basic configuration, with no content (no courses). The only Moodle user account is an administrator account (username: *moodlebox*, password: *Moodlebox4\$*). The server is configured to accept clients from the official Moodle app.² *cron* service is launched every minute.
- When a USB stick is inserted into the MoodleBox, its files are available to users in Moodle's *File system* repository.
- Ability to upload files via SFTP directly to the MoodleBox (username: *moodlebox*, password: *Moodlebox4\$*); these files are then available to users in Moodle's *File system* repository.
- Adminer, a full-featured database management tool, is installed (<http://moodlebox.home/adminer.php>).

What MoodleBox doesn't do

- Email server: MoodleBox is intended to be used “in the field”, independent of any network infrastructure; email server functionality is not relevant for this purpose.
- Coffee machine.

2 Raspberry Pi preparation

Although MoodleBox works fine on other models, a Raspberry Pi Model 3B, 3B+ or 4B is recommended to **build** it.

¹https://github.com/moodlebox/moodle-tool_moodlebox.

²<https://download.moodle.org/mobile/>.

Copy last version of Raspberry Pi OS Lite on a microSD card

Download last version of *Raspberry Pi OS Lite* from Raspberry Pi web site,³ copy it on your (good quality!) microSD card. Complete description of the process is available on Raspberry Pi web site.⁴

Enable SSH access

For security reasons, SSH access isn't enabled by default.⁵ To enable it, we copy a file with name `ssh` in the boot directory (root of boot partition of microSD card). The content of the file doesn't matter. The following commands will do.

```
$ cd <mounting point of "boot" partition>
$ > ssh
```

Eject the microSD card and insert it into the Raspberry Pi.

Connect the power supply of the Raspberry Pi. Plug the Raspberry Pi with an Ethernet cable into a network with a DHCP server and wait 20-30 seconds. Your Raspberry Pi is now reachable on the network using the address `raspberrypi.local`.⁶

Log into the Raspberry Pi via SSH

From now on, all operations are done via the command line, using `ssh` (a regular terminal on macOS and Linux, or Putty on Windows).

Login to the Raspberry Pi. Username is *pi* with password *raspberry* (this password will be changed later, as will the user account).

```
$ ssh pi@raspberrypi.local
$ pi@raspberrypi.local's password:
```

Upgrade Raspberry Pi OS

Upgrade the Raspberry Pi's operating system, with these commands:

```
$ sudo apt-get update -y
$ sudo apt-get full-upgrade -y
```

This step can take several minutes, depending on your Internet connection speed.

³<https://www.raspberrypi.com/software/operating-systems>.

⁴<https://www.raspberrypi.org/documentation/computers/getting-started.html>.

⁵See <https://www.raspberrypi.org/blog/a-security-update-for-raspbian-pixel/>

⁶This network address is provided by the *zeroconf* standard protocol. Some older Android and Windows devices do not understand this protocol. With such devices, it is necessary to get access to the Raspberry Pi via its numerical IP address, which must be discovered manually.

Configure some Raspberry Pi important settings

Launch *raspi-config* utility:

```
$ sudo raspi-config
```

With *raspi-config*, configure following settings:

- Expand Filesystem.
- Change password for the *pi* user. As the disk image being produced is intended to be openly distributed, we choose MoodleBox default password *Moodlebox4\$*.
- Install following *locales*:

- *de_DE.UTF-8*
- *en_AU.UTF-8*
- *en_GB.UTF-8*
- *es_ES.UTF-8*
- *fr_FR.UTF-8*
- *it_IT.UTF-8*

and set *en_GB.UTF-8* as default *locale*.

- Set time zone and WLAN country. This will also unblock wireless use. MoodleBox default settings are respectively Europe/Zurich and CH.
- Set the hostname to *moodlebox*.

From now on, to connect to the Raspberry Pi (via SSH or SFTP), use address *moodlebox.local*, username *pi* and password *Moodlebox4\$*.

Reboot and log into the MoodleBox with these new credentials.

Change default username⁷

There's no problem to use default username *pi* for MoodleBox usage. However, for the convenience of the end-user, we change the default username to *moodlebox*.

We start by creating a temporary user account, *tempuser*, and set its password. As this account will be deleted later, the quality of the password is not important.

```
$ sudo useradd -m tempuser  
$ sudo passwd tempuser
```

User account *tempuser* is added to *sudoers* user group, to allow it to perform the appropriate operations. The current session of user *pi* is then exited.

```
$ sudo usermod -a -G sudo tempuser  
$ exit
```

⁷This section is based on <http://unixetc.co.uk/2016/01/07/how-to-rename-the-default-raspberry-pi-user/>.

Log now into the MoodleBox with username *tempuser* and its previously set password.

```
$ ssh tempuser@moodlebox.local
$ tempuser@moodlebox.local's password:
```

We first save the files we're going to modify as a backup, in case a problem occurs.

```
$ cd /etc
$ sudo tar -czf /home/pi/authfiles.tgz passwd group shadow gshadow subuid
subgid sudoers sudoers.d/010_pi-nopasswd systemd/system/autologin@.
service
```

We change now the account name to *moodlebox* in the relevant files, then change its home directory name. A symbolic link to the old directory is created, in order to avoid some unlikely (but possible) side effects due to the account name change.

```
$ sudo sed -i 's/\bpi\b/moodlebox/g' passwd group shadow gshadow subuid
subgid sudoers sudoers.d/010_pi-nopasswd systemd/system/autologin@.
service
$ sudo mv /etc/sudoers.d/010_pi-nopasswd /etc/sudoers.d/010_moodlebox-
nopasswd
$ sudo mv /home/pi /home/moodlebox
$ sudo ln -s /home/moodlebox /home/pi
$ exit
```

We can now log into the MoodleBox with the new username *moodlebox* and password *Moodlebox4\$* (which was not changed).

```
$ ssh moodlebox@moodlebox.local
$ moodlebox@moodlebox.local's password:
```

Temporary user account *tempuser* can be now deleted, as well as the backup file:

```
$ sudo userdel tempuser
$ sudo rm -rf /home/tempuser/
$ sudo rm authfiles.tgz
```

From now on, we will log into the MoodleBox with username *moodlebox* and password *Moodlebox4\$*.

Increase system available memory

To increase the system available memory, the memory reserved for the graphics chip is reduced to 16 MB. This is of no consequence, as our system does not have a graphical interface. We do this by adding the following line at the end of the file `/boot/config.txt`:

```
gpu_mem=16
```

Enable shutdown/startup hardware button

We enable shutdown/startup hardware button feature by further editing file `/boot/config.txt`. Insert following line after line beginning with `# Additional overlays` to get:

```
# Additional overlays and parameters are documented /boot/overlays/README
dtoverlay=gpio-shutdown
```

3 Wireless access point (AP) feature

Enable wireless interface for access point mode

We want to be able to use standard wireless interface `wlan0` to optionally connect to a wireless LAN, so we need another interface for our AP.

We use a *udev* rule for this purpose, creating a new file named `90-wireless.rules` in directory `/etc/udev/rules.d/`.

Here's the content of `/etc/udev/rules.d/90-wireless.rules` on the MoodleBox:

```
# File /etc/udev/rules.d/90-wireless.rules
# Add wireless interface for AP mode
ACTION=="add", SUBSYSTEM=="ieee80211", KERNEL=="phy0", \
    RUN+="/sbin/iw phy %k interface add uap0 type __ap"
```

Switch to alternative wireless chip firmware

Standard Raspberry Pi wireless firmware has several additional features that are not very useful for a MoodleBox. However, SRAM on the wireless chip is used both for data storage whilst in use, but also for additional features and bug fixes. Each time a feature is added or a bug fix made, the amount of SRAM available for runtime variables decreases, and so the number of clients in AP mode.

An alternative firmware is now available that has been tuned to maximise the number of clients in AP mode while still supporting STA mode. This is the firmware we choose to use in MoodleBox. Following commands switch the firmware to the alternative one.

```
$ cd /lib/firmware/brcm/
$ sudo ln -sf ../cyprss/cyfm43455-sdio-minimal.bin \
    brcmfmac43455-sdio.bin
```

Wireless connection as a client (optional)

If MoodleBox is intended to be used as a client (STA mode) as well as an access point, a valid configuration file named `wpa_supplicant.conf` should be provided to `wpa_supplicant` service, in directory `/etc/wpa_supplicant/`.

Following listing shows an example of such file:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=CH

network={
    scan_ssid=1
    ssid="<Name of your wireless LAN>"
    psk="<Password for your wireless LAN>"
    proto=RSN
    key_mgmt=WPA-PSK
    pairwise=CCMP
    auth_alg=OPEN
}
```

Don't forget to edit the file and put the SSID and password of your wireless network.

Install required packages for access point feature

Access point feature requires packages `hostapd` and `dnsmasq`. Let's install them.

```
$ sudo apt-get install -y hostapd dnsmasq
```

Static IP address configuration

MoodleBox gets dynamically via DHCP its IP address on the ethernet interface `eth0` or as a wireless client on the `wlan0` interface.

With its access point feature, it will also act as a DHCP provider on its wireless interface `uap0`. We force then a static IP address `10.0.0.1` on `wlan0` interface. Any other private IP address can alternatively be used.

To this end, we add at the very end of file `/etc/dhcpd.conf` the lines

```
interface wlan0
# Uncomment following line to disable AP+STA mode
# nohook wpa_supplicant

interface uap0
static ip_address=10.0.0.1/24
nohook wpa_supplicant
```

Access point configuration

To configure the access point, we edit `hostapd.conf`. This is where the access point SSID and password are set, as well as other options, such as the broadcast channel and regulatory country. We define *MoodleBox* as SSID and *moodlebox* as password.

Here's the content of `/etc/hostapd/hostapd.conf` on the MoodleBox:

```
# Set country code
country_code=CH
# Name of the Wi-Fi interface
interface=uap0
# Use the nl80211 driver
driver=nl80211
# Wi-Fi network name (SSID)
ssid=MoodleBox
# Show or hide SSID
ignore_broadcast_ssid=0
# Use the 2.4GHz band
hw_mode=g
# The Wi-Fi channel
channel=11
# Enable 802.11n
ieee80211n=1
# Enable WMM
wmm_enabled=1
# Enable 40 MHz channels with short guard interval for 20 Mhz
ht_capab=[HT40][SHORT-GI-20][DSSS_CCK-40]
# Accept all MAC addresses
macaddr_acl=0
# Use WPA authentication
auth_algs=1
# Use WPA2
wpa=2
# Use a pre-shared key
wpa_key_mgmt=WPA-PSK
# The network passphrase
wpa_passphrase=moodlebox
# Use AES, instead of TKIP
rsn_pairwise=CCMP
# Enable hostapd_cli
ctrl_interface=/var/run/hostapd
ctrl_interface_group=0
```

We now need to define where hostapd gets its configuration. This is set in file `/etc/default/hostapd`. In this file we replace `#DAEMON_CONF=""` with `DAEMON_CONF="/etc/hostapd/hostapd.conf"`.

Finally, we enable hostapd service:

```
$ sudo systemctl unmask hostapd
$ sudo systemctl enable hostapd
```

DHCP and DNS server configuration

DHCP and DNS server configuration on uap0 interface are provided through dnsmasq.⁸ We edit `/etc/dnsmasq.conf` to have this content:

⁸DNS configuration is based on <https://www.linux.com/learn/dnsmasq-easy-lan-name-services>.


```

interface=uap0          # Use interface uap0
listen-address=127.0.0.1 # Explicitly specify the address to listen on
listen-address=10.0.0.1 # Explicitly specify the address to listen on
bind-interfaces          # Make sure we aren't sending things elsewhere
server=9.9.9.9           # Forward DNS requests to external public DNS
server=149.112.112.112  # Forward DNS requests to external public DNS
domain-needed           # Don't forward short names
bogus-priv              # Don't forward addresses in the non-routed spaces
domain=home             # Set private domain name to 'home'
local=/home/            # Don't forward queries for private domain 'home'
address=/home/10.0.0.1  # Resolve subdomains '*.home'
expand-hosts            # Add private domain name to hostnames
dhcp-range=wifi,10.0.0.10,10.0.0.254,255.255.255.0,1h # Assign IP addresses
                        with 1h lease, subnet name 'wifi'
dhcp-option=wifi,6,10.0.0.1 # Set DNS server for subnet wifi
txt-record=moodlebox.home,"MoodleBox by Nicolas Martignoni"
log-facility=/var/log/dnsmasq.log # Enable log

```

We edit now the file `/etc/hosts`, replacing last line, beginning with `127.0.1.1`, with this line:

```
10.0.0.1    moodlebox
```

This configuration enables any device to access the MoodleBox via its URL <http://moodlebox.home/>, even those which do not implement *zeroconf*⁹ standard protocol.

Finally, we fix a race condition between `dhcpcd` and `dnsmasq` by editing `dnsmasq` service file `/lib/systemd/system/dnsmasq.service`. We add following lines just before `[Install]`:

```
RestartSec=5
Restart=on-failure
```

mDNS services publication

In order to make MoodleBox services visible on the network, we create the file `/etc/avahi/services/moodlebox.service`, with following content.

```

<?xml version="1.0" standalone='no'?>
<!DOCTYPE service-group SYSTEM "avahi-service.dtd">
<service-group>
  <name replace-wildcards="yes">%h</name>
  <service>
    <type>_device-info._tcp</type>
    <port>0</port>
    <txt-record>model=MoodleBox</txt-record>
  </service>
  <service>
    <type>_ssh._tcp</type>
    <port>22</port>
  </service>
</service-group>

```

⁹https://en.wikipedia.org/wiki/Zero-configuration_networking

```
</service>
<service>
  <type>_sftp-ssh._tcp</type>
  <port>22</port>
</service>
<service>
  <type>_http._tcp</type>
  <port>80</port>
</service>
</service-group>
```

Routing configuration

We configure routing so that wireless clients can browse the Internet when MoodleBox is connected via ethernet or wireless to an Internet router.

We edit the file `/etc/sysctl.conf`, uncommenting or adding line

```
net.ipv4.ip_forward=1
```

Installation of package `iptables-persistent` enables routing rules to survive MoodleBox reboot or shutdown.

```
$ sudo apt-get install -y iptables-persistent
```

We can now define the routing rules:

```
$ sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
$ sudo iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE
$ sudo iptables -A FORWARD -i eth0 -o uap0 \
  -m state --state RELATED,ESTABLISHED -j ACCEPT
$ sudo iptables -A FORWARD -i wlan0 -o uap0 \
  -m state --state RELATED,ESTABLISHED -j ACCEPT
$ sudo iptables -A FORWARD -i uap0 -o eth0 -j ACCEPT
$ sudo iptables -A FORWARD -i uap0 -o wlan0 -j ACCEPT
```

And we reboot.

```
$ sudo reboot
```

Captive portal

We can now install the captive portal, based on free software Nodogsplash.¹⁰ Nodogsplash needs first to be compiled for the Raspberry Pi platform. Its compilation is not covered in this guide. Refer to [Nodogsplash documentation](https://nodogsplashdocs.readthedocs.io/) for any support about its compilation.

Let's install the Nodogsplash package we got after compilation:

¹⁰<https://nodogsplashdocs.readthedocs.io/>.

```
$ sudo dpkg -i nodogsplash_5.0.0-1_armhf.deb
```

Nodogsplash configuration file `/etc/nodogsplash/nodogsplash.conf` should read

```
# Nodogsplash Configuration File
GatewayInterface uap0
FirewallRuleSet authenticated-users {
    FirewallRule allow all
}
FirewallRuleSet preauthenticated-users {
}
FirewallRuleSet users-to-router {
    FirewallRule allow udp port 53
    FirewallRule allow tcp port 53
    FirewallRule allow udp port 67
    FirewallRule allow tcp port 22
    FirewallRule allow tcp port 80
    FirewallRule allow tcp port 443
}
GatewayName MoodleBox
GatewayAddress 10.0.0.1
RedirectURL http://moodlebox.home/
GatewayPort 2050
MaxClients 50
SessionTimeout 360
```

We can now edit Nodogsplash splash page at our convenience. The files to edit are located in directory `/etc/nodogsplash/htdocs/`.

In the official MoodleBox image, Nodogsplash captive portal is not enabled. If we want to disable it too, we type following commands in our shell:

```
$ sudo systemctl stop nodogsplash.service
$ sudo systemctl disable nodogsplash.service
```

4 Web server and database installation (LEMP stack)

The LEMP software stack is a group of software that can be used to serve dynamic web pages and web applications. The term LEMP is an acronym that represents a Linux operating system with an Nginx (pronounced “engine-x”, hence the E in the acronym) web server. The backend data stored in a MariaDB database and the dynamic processing is handled by PHP.

Nginx and PHP installation et configuration

We install first Nginx and all needed PHP packages, notably those required by Moodle.

```
$ sudo apt-get install -y nginx php7.4-fpm php7.4-cli php7.4-common \
    php7.4-json php7.4-mbstring php7.4-opcache php7.4-readline \
```

```
php7.4-xmlrpc php7.4-curl php7.4-gd php7.4-intl php7.4-soap \
php7.4-mysql php7.4-xml php7.4-zip php7.4 php-apcu
```

SSL certificate and key

We can now copy the SSL certificate `moodlebox.pem` and its key `moodlebox.key` to the directory `/etc/nginx/`.

SSL certificate generation is not covered in this guide. Refer to adequate documentation on SSL certificate generation, e.g. on <https://stackoverflow.com/a/10176685>.

Web server configuration

Nginx web server configuration is done in file `/etc/nginx/sites-available/default`, which content should be like below.

```
# Default server configuration
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    listen 443 ssl;
    listen [::]:443 ssl;
    ssl_certificate /etc/nginx/ssl/moodlebox.pem;
    ssl_certificate_key /etc/nginx/ssl/moodlebox.key;

    root /var/www/moodle;

    index index.php index.html index.htm index.nginx-debian.html;

    server_name moodlebox;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ [^/]\.php(/|$) {
        include fastcgi_params;
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_read_timeout 300;
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param PATH_INFO $fastcgi_path_info;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param PHP_VALUE "max_execution_time=300\n
            upload_max_filesize=50M\n post_max_size=50M\n max_input_vars
            =5000";
        client_max_body_size 50M;
    }
}
```

Last `fastcgi_param` line, with line `client_max_body_size`, increases to 50 MB the max upload file size, as well as script maximum execution time to 300 s. It also sets `max_input_vars` to 5000.

We then relaunch the web server:

```
$ sudo systemctl restart nginx php7.4-fpm
```

The ownership, group and access rights of Nginx and PHP should now be tweaked so that the default user *moodlebox* can easily edit the files, while allowing Moodle updates and plugin installation via the web interface.

To do this, we edit the two files `/lib/systemd/system/nginx.service` and `/lib/systemd/system/php7.4-fpm.service`, adding after line `[Service]` the line `UMask=0002`, to get something like

```
...
[Service]
UMask=0002
Type=...
...
```

We then edit the file `/etc/php/7.4/fpm/pool.d/www.conf` to set the correct group ownership for PHP, replacing line `group = www-data` with `group = moodlebox`, so we have now

```
...
user = www-data
group = moodlebox
...
```

If needed, variables in `/etc/php/7.4/fpm/pool.d/www.conf` can be tweaked, e.g. for performance gains.

MariaDB installation and configuration

We install now MariaDB package. PHP MariaDB support was already installed with PHP (see section 4).

```
$ sudo apt-get install -y mariadb-server
```

During the installation, we define the password of the main user *moodlebox* of the database. For this installation, we set the password to *Moodlebox4\$*.

In order to allow flexible access to the databases, a new database user is created in MariaDB.

```
$ sudo mysql
> CREATE USER 'moodlebox'@'localhost' IDENTIFIED BY 'Moodlebox4$';
> GRANT ALL PRIVILEGES ON *.* TO 'moodlebox'@'localhost';
> FLUSH PRIVILEGES;
> QUIT;
```

If needed, variables in `/etc/mysql/mariadb.conf.d/50-server.cnf` can be tweaked, e.g. for performance gains.

5 Moodle installation and configuration

Moodle database creation

We first create the database which Moodle will use.

```
$ sudo mysql
> CREATE DATABASE moodle;
> QUIT;
```

Moodle download

We can now download Moodle; we do this with Git in order to facilitate future updates. We first install Git.

```
$ sudo apt-get install -y git
```

We get now Moodle source in the appropriate location, specifying the current stable branch of Moodle. A shallow clone is used to later save space on the MoodleBox image.

```
$ sudo git clone --depth=1 -b MOODLE_311_STABLE \
git://git.moodle.org/moodle.git /var/www/moodle
```

Creation of Moodle data directories

Moodle data and several other directories are now created.

```
$ sudo mkdir -p /var/www/moodledata/repository /var/www/moodledata/temp \
/var/www/moodledata/backup /var/cache/moodle /var/cache/moodle-cache-
backup
```

Adequate permissions and ownership are set on these directories, including SGID permission on Moodle data directory. We also set the correct permissions to Moodle source directory:

```
$ sudo chown -R www-data:moodlebox /var/www/moodle /var/www/moodledata/ \
/var/cache/moodle /var/cache/moodle-cache-backup
$ sudo chmod -R ug+w,o-w /var/www/moodle /var/www/moodledata/ \
/var/cache/moodle /var/cache/moodle-cache-backup
$ sudo chmod -R g+s /var/www/moodledata/
```

We can now launch Moodle installation, either by loading URL <http://moodlebox.home/> in a browser and follow the instructions on screen, or by using Moodle command line interface.¹¹ For MoodleBox, the administrator account is defined with *moodlebox* as username and the password *Moodlebox4\$*.

The installation takes at least 10 minutes. So be patient!

Tweaking the Moodle configuration

Moodle course backup directory

As we use for Moodle course backup a different directory than the data directory, we set the following line in Moodle configuration file `/var/www/moodle/config.php`:

```
$CFG->backuptempdir = '/var/www/moodledata/backup';
```

X-Sendfile setup

We set *X-Sendfile* in `/var/www/moodle/config.php` to allow files in the Moodle data directory to be uploaded faster through the web server:

```
$CFG->xsendfile = 'X-Accel-Redirect';  
$CFG->xsendfilealiases = array ('/dataroot/' => $CFG->dataroot);
```

For *X-Sendfile* to be active, we must add following lines to Nginx configuration file `/etc/nginx/sites-available/default`, inside server block:

```
location /dataroot/ {  
    internal;  
    alias /var/www/moodledata/  
}
```

Custom filetype definition

Defining a custom file type in Moodle facilitates the web certificate from MoodleBox home page. The definition is coded in `/var/www/moodle/config.php` too:

```
$CFG->customfiletypes = array(  
    (object)array(  
        'extension' => 'crt',  
        'icon' => 'sourcecode',  
        'type' => 'application/x-x509-ca-cert',  
        'customdescription' => 'X.509 CA certificate'  
    )  
);
```

¹¹See https://docs.moodle.org/311/en/Installing_Moodle#Command_line_installer.

Hiding content that doesn't make sense

We hide Moodle campaign and Moodle services and support sections from notifications page. These content don't make sense for MoodleBox, which is off-line most of the time.

```
$CFG->showcampaigncontent = false;
$CFG->showservicesandsupportcontent = false;
```

Final content of the file `/var/www/moodle/config.php` is then:

```
<?php // Moodle configuration file

unset($CFG);
global $CFG;
$CFG = new stdClass();

$CFG->dbtype      = 'mariadb';
$CFG->dblibrary   = 'native';
$CFG->dbhost      = 'localhost';
$CFG->dbname      = 'moodle';
$CFG->dbuser      = 'moodlebox';
$CFG->dbpass      = 'Moodlebox4$';
$CFG->prefix      = 'mdl_';
$CFG->dboptions   = array (
    'dbpersist' => 0,
    'dbport'    => '',
    'dbsocket'  => '',
    'dbcollation' => 'utf8mb4_general_ci',
);

$CFG->wwwroot     = 'http://moodlebox.home';
$CFG->dataroot     = '/var/www/moodledata';
$CFG->admin        = 'admin';

$CFG->backuptempdir = '/var/www/moodledata/backup';
$CFG->xsendfile     = 'X-Accel-Redirect';
$CFG->xsendfilealiases = array('/dataroot/' => $CFG->dataroot);
$CFG->customfiletypes = array(
    (object)array(
        'extension' => 'crt',
        'icon'      => 'sourcecode',
        'type'      => 'application/x-x509-ca-cert',
        'customdescription' => 'X.509 CA certificate'
    )
);
$CFG->showcampaigncontent = false;
$CFG->showservicesandsupportcontent = false;

$CFG->directorypermissions = 02777;

require_once(__DIR__ . '/lib/setup.php');

// There is no php closing tag in this file,
// it is intentional because it prevents trailing whitespace problems!
```


Cron configuration

On lance le cron de Moodle toutes les 3 minutes. Pour ce faire, lancer la commande

```
$ sudo crontab -e
```

et ajouter à la table des crons la ligne

```
*/3 * * * * nice -n 10 ionice -c2 /usr/bin/php /var/www/moodle/admin/cli/cron.php
```

6 Accès depuis Moodle aux fichiers d'une clef USB ou déposés par SFTP

Fichiers sur une clef USB

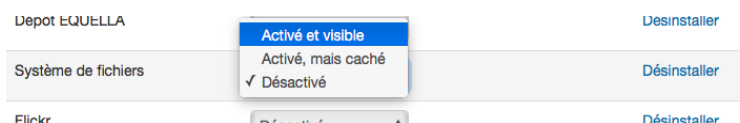
On configure le montage automatique de clefs USB (quel que soit leur format), ainsi que l'accès à tous les fichiers de clefs USB branchées via le dépôt *Système de fichiers* de Moodle.¹²

Pour ce faire, on installe le paquet `usbmount`, puis on crée un dossier dans le dossier de données de Moodle, et l'on crée enfin un lien depuis le point de montage de la clef USB vers ce dossier.

```
$ sudo apt-get install -y usbmount ntfs-3g exfat-fuse
$ sudo mkdir -p /var/www/moodledata/repository
$ sudo chown -R www-data:moodlebox /var/www/moodledata/
$ sudo ln -s /media/usb /var/www/moodledata/repository
```

Il faut ensuite configurer Moodle de façon adéquate:¹³ connecté comme administrateur sur la plateforme, on visite *Administration du site > Plugins > Dépôts > Gérer les dépôts*.

Dans la ligne *Système de fichiers*, on sélectionne dans le menu déroulant *Activé et visible*.



On clique ensuite sur *Enregistrer*, puis sur *Paramètres* (même ligne), puis sur le bouton *Créer une instance de dépôt*. Enfin, on sélectionne *usb* dans le menu déroulant, et on indique *USB Drive* dans le champ obligatoire *Nom*.

Fichiers déposés par SFTP

On crée un dossier dans lequel les fichiers devront être déposés pour être accessibles depuis Moodle, ainsi qu'un lien vers le dossier de données de Moodle. Les permissions adéquates sont

¹²Si plusieurs clefs USB sont branchées à la MoodleBox, seuls les fichiers de la première clef branchée sont accessibles via le dépôt de fichiers.

¹³Voir https://docs.moodle.org/en/File_system_repository.

Configuration pour dépôt système de fichiers

Nom*

Ces dossiers sont situés dans le dossier `/var/www/moodledata/repository/`.

Permettre les fichiers relatifs ☐ Ce réglage permet l'accès à tous les fichiers du dépôt au moyen de liens relatifs.

données.

```
$ mkdir -p /home/moodlebox/files
$ sudo chown -R moodlebox:www-data files/
$ sudo chmod g+s files/
$ sudo ln -s /home/moodlebox/files /var/www/moodledata/repository
```

On effectue la configuration d'un dépôt *Système de fichiers*, de façon similaire au dépôt « USB Drive » ci-dessus, en indiquant le dossier *files* et en indiquant *SFTP Documents* comme nom de dépôt.

Pour déposer des fichiers, on se connecte au moyen d'un logiciel SFTP¹⁴ sur la MoodleBox, avec le nom d'utilisateur *moodlebox* et le mot de passe *Moodlebox4\$*. Les fichiers seront déposés dans le dossier *files*.

Installation de ghostscript et de poppler-utils

7 Configurations additionnelles de Moodle

Activation de l'accès via l'app mobile de Moodle

Après s'être connecté à la plateforme avec le compte administrateur, on visite dans le Moodle *Administration du site > Plugins > Services web > Mobile*. On coche la case *Activer les services web pour appareils mobiles* et l'on enregistre les modifications.

Pour cette plateforme dont la destination n'est pas d'être publiée sur Internet, l'avertissement concernant le certificat SSL peut être ignoré sans risque.

Installation du plugin Administration MoodleBox

Afin de permettre le redémarrage et l'arrêt de la MoodleBox sans risque de corruption de la carte microSD, la modification du mot de passe, le réglage de l'heure, le changement du mot de passe du réseau Wi-Fi, ainsi que pour donner des informations utiles sur son fonctionnement, on installe le plugin d'administration *MoodleBox* pour Moodle.¹⁵

Le plus simple est de l'installer au moyen de l'interface graphique de Moodle, en visitant *Administration du site > Plugins > Dépôts > Installer des plugins*. On clique sur le bouton *Installer des*

¹⁴Par exemple : [FileZilla](#), [Cyberduck](#), [WinSCP](#).

¹⁵Plugin MoodleBox <https://github.com/martignoni/moodlebox-plugin>.

plugins à partir du répertoire des plugins Moodle et choisit le plugin d'administration MoodleBox (Admin tools).

Il est aussi possible de l'installer via Git.¹⁶

```
$ cd /var/www/moodle/admin/tool/  
$ sudo git clone https://github.com/martignoni/moodlebox-plugin.git  
moodlebox
```

On termine dans ce cas l'installation du plugin en visitant la page <http://moodlebox.home/admin>.

Pour finaliser l'installation du plugin, il est nécessaire de créer quelques fichiers et d'en régler les autorisations.

```
$ cd /var/www/moodle/admin/tool/moodlebox  
$ sudo touch .reboot-server; touch .shutdown-server; touch .set-server-  
datetime; touch .newpassword; touch .wifisettings  
$ sudo chown -R www-data:moodlebox /var/www/moodle/admin/tool/moodlebox  
$ sudo chmod -R ug+w,o-w /var/www/moodle/admin/tool/moodlebox
```

TODO: Replace incron with direvent

On installe finalement le paquetage *incron*.

```
$ sudo apt-get install -y incron
```

On autorise l'utilisation de *incron* par *root*, puis modifie la table des tâches de *incron*.

```
$ echo root | sudo tee -a /etc/incron.allow  
$ sudo incrontab -e
```

en y ajoutant les lignes

```
/var/www/moodle/admin/tool/moodlebox/.reboot-server IN_CLOSE_WRITE /sbin/  
shutdown -r now  
/var/www/moodle/admin/tool/moodlebox/.shutdown-server IN_CLOSE_WRITE /sbin/  
shutdown -h now  
/var/www/moodle/admin/tool/moodlebox/.set-server-datetime IN_MODIFY /bin/  
bash /var/www/moodle/admin/tool/moodlebox/.set-server-datetime  
/var/www/moodle/admin/tool/moodlebox/.newpassword IN_CLOSE_WRITE /bin/bash  
/var/www/moodle/admin/tool/moodlebox/bin/changepassword.sh  
/var/www/moodle/admin/tool/moodlebox/.wifisettings IN_CLOSE_WRITE /bin/bash  
/var/www/moodle/admin/tool/moodlebox/bin/changewifisettings.sh
```

8 Configuration de Adminer

¹⁶Le code source du plugin MoodleBox est disponible à l'adresse https://moodle.org/plugins/tool_moodlebox.

```
$ sudo apt-get install -y -t stretch phpmyadmin
$ sudo ln -s /usr/share/phpmyadmin /var/www/moodle/phpmyadmin
```

Définir un mot de passe fort. Pour cette installation, le mot de passe choisi est *Moodlebox4\$*. Pour accéder à la totalité des fonctionnalités de PhpMyAdmin, on accèdera à l'interface au moyen de l'utilisateur *moodlebox* et du mot de passe *Moodlebox4\$*.

9 Optimisation

Pour que la MoodleBox soit utilisable en pratique, il est nécessaire de prendre soin à son optimisation. On configure ainsi le cache de Moodle, ainsi que sa gestion des dépôts et téléchargements de fichiers.

Disque RAM pour certains dossiers de Moodle¹⁷

Créer un dossier comme point de montage pour le disque RAM.

```
$ sudo mkdir /var/cache/moodle
$ sudo chown www-data:moodlebox /var/cache/moodle/
$ sudo chmod ug+w,o-w /var/cache/moodle/
```

On utilise également des disques RAM pour le dossier temporaire et le dossiers des sessions de Moodle. Ces deux dossiers sont situés dans le dossier *moodledata*.

On définit dans la table des partitions du Raspberry les disques RAM. Pour ce faire, on ajoute au fichier */etc/fstab*, les lignes suivantes

```
tmpfs /var/cache/moodle tmpfs size=64M,mode=775,uid=www-data,gid=www-data 0
0
tmpfs /var/www/moodledata/temp tmpfs size=64M,mode=775,uid=www-data,gid=www
-data 0 0
tmpfs /var/www/moodledata/sessions tmpfs size=32M,mode=775,uid=www-data,gid
=www-data 0 0
```

Le contenu du fichier */etc/fstab* sera alors :

```
proc /proc proc defaults 0 0
PARTUUID=2e60d480-01 /boot vfat defaults 0 2
PARTUUID=2e60d480-02 / ext4 defaults,noatime 0 1
tmpfs /var/cache/moodle tmpfs size=64M,mode=775,uid=www-data,gid=www-data 0
0
tmpfs /var/www/moodledata/temp tmpfs size=64M,mode=775,uid=www-data,gid=www
-data 0 0
tmpfs /var/www/moodledata/sessions tmpfs size=16M,mode=775,uid=www-data,gid
=www-data 0 0
```

¹⁷Cette section est en partie inspirée de <https://www.leading-interactive.de/e-learning/moodle-performance-tuning-mit-tmpfs/>.

```
# a swapfile is not a swap partition, no line here
# use dphys-swapfile swap[on|off] for that
```

Après un redémarrage de la Raspberry, le cache peut être configuré dans Moodle.

On se connecte à Moodle avec le compte administrateur (créé plus haut), puis dans le Moodle on visite *Administration du site > Plugins > Cache > Configuration*. On crée deux nouvelles instances de dépôt, en cliquant sur *Ajouter une instance* dans la section *Entrepôts de cache installés* (en haut de la page).

Ajouter entrepôt pour Cache de fichiers

Nom de l'entrepôt* ?

Verrouillage ?

▼ Configuration entrepôt

Chemin d'accès du cache ?

Auto-crédation de dossiers ? ☒

Dépôt dossier unique ? ☐

Pré-scanner le dossier ? ☐

Ajouter entrepôt pour Cache de fichiers

Nom de l'entrepôt* ?

Verrouillage ?

▼ Configuration entrepôt

Chemin d'accès du cache ?

Auto-crédation de dossiers ? ☒

Dépôt dossier unique ? ☐

Pré-scanner le dossier ? ☐

1. Nom de l'entrepôt : *TMPFS application*, chemin d'accès du cache : */var/cache/moodle/ap-
plication*, cocher la case *Auto-crédation de dossiers*;
2. Nom de l'entrepôt : *TMPFS sessions*, chemin d'accès du cache : */var/cache/moodle/sessions*,
cocher la case *Auto-crédation de dossiers*.

Pour terminer, il reste à associer ces nouvelles instances de cache à leur destination, en cliquant sur *Modifier les correspondances* tout en bas de la page, dans le domaine *Entrepôts utilisés en l'absence de correspondance*.

Administration du cache

Application

Session

Requête

Le cache en disque RAM a un défaut: les données qu'il contient disparaissent à chaque redémarrage de la MoodleBox. Moodle doit donc à chaque fois reconstruire le cache. Si l'on veut conserver le cache entre les redémarrages, on copie à intervalle régulier le contenu du disque RAM sur la carte microSD, et, lors de chaque démarrage, on effectue l'opération inverse.

On crée le dossier de sauvegarde, puis on définit le cron

```
$ sudo mkdir /var/cache/moodle-cache-backup/
$ sudo crontab -e
```

On ajoute à la table des crons les deux lignes suivantes pour effectuer la sauvegarde du cache toutes les 20 minutes et pour restaurer le cache au démarrage:

```
*/20 * * * * rsync -a --delete /var/cache/moodle/ /var/cache/moodle-cache-
backup/
@reboot cp -Rpf /var/cache/moodle-cache-backup/* /var/cache/moodle/
```

Optimisation de MariaDB

On modifie les valeurs de quelques variables de MariaDB dans le fichier `/etc/mysql/mariadb.conf.d/50-server.cnf`. Pour ce faire, on ouvre le fichier en question et on y modifie les lignes adéquates, à savoir:

```
table_cache           = 512
table_definition_cache = 512
max_connections        = 100
query_cache_size       = 16M
query_cache_type        = 0
```

10 Redimensionnement automatique de la partition

Afin d'éviter à l'utilisateur de devoir redimensionner manuellement la partition de travail, on configure le redimensionnement automatique au premier démarrage. La méthode est identique à celle utilisée lors de la construction de l'image-disque de *Raspbian Stretch Lite*.¹⁸

On copie le fichier `resize2fs_once` ci-dessous dans le dossier `/etc/init.d/`, et on lui donne les permissions adéquates pour être lancé au redémarrage.

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          resize2fs_once
# Required-Start:
# Required-Stop:
# Default-Start: 3
# Default-Stop:
# Short-Description: Resize the root filesystem to fill partition
# Description:
### END INIT INFO
. /lib/lsb/init-functions
case "$1" in
    start)
        log_daemon_msg "Starting resize2fs_once"
```

¹⁸<https://github.com/RPi-Distro/pi-gen>.

```

ROOT_DEV=`grep -Eo 'root=[[:graph:]]+' /proc/cmdline | cut -d '=' -f
2-` &&
resize2fs $ROOT_DEV &&
update-rc.d resize2fs_once remove &&
rm /etc/init.d/resize2fs_once &&
log_end_msg $?
;;
*)
echo "Usage: $0 start" >&2
exit 3
;;
esac

```

Finalement, avant la dernière extinction précédant le clonage et le redimensionnement de l'image disque, on termine en lançant la commande

```
$ sudo systemctl enable resize2fs_once
```

puis on ajoute à la fin de la ligne du fichier `/boot/cmdline.txt` les instructions ci-dessous, immédiatement après le texte `rootwait` (sans oublier un espace).

```
quiet init=/usr/lib/raspi-config/init_resize.sh
```

Il est indispensable de ne pas redémarrer la MoodleBox, sans quoi l'opération décrite dans cette section devra être intégralement recommencée.

11 Nettoyage de la distribution

Les commandes ci-dessous permettent de nettoyer la MoodleBox et de diminuer l'espace disque qui lui est nécessaire, avant de la cloner et de la distribuer.

```

$ sudo rm -r /var/www/moodledata/cache/*
$ sudo rm -r /var/www/moodledata/localcache/*
$ sudo rm -r /var/www/moodledata/temp/*
$ sudo rm -r /var/www/moodledata/trashdir/*
$ sudo rm -r /var/www/moodledata/sessions/*
$ sudo rm -r /var/cache/moodle/*
$ sudo rm -r /var/cache/moodle-cache-backup/*
$ sudo mysql moodle -e "truncate table moodle.mdl_logstore_standard_log"
$ sudo mysql moodle -e "truncate table moodle.mdl_config_log"
$ sudo mysql moodle -e "truncate table moodle.mdl_upgrade_log"
$ sudo apt-get clean
$ sudo rm -r /var/lib/apt/lists/*
$ sudo rm -r /var/cache/debconf/*
$ sudo rm -r /tmp/*
$ sudo rm -r /var/tmp/*
$ rm ~/.mysql_history
$ sudo bash -c 'for logs in `find /var/log -type f`; do > $logs; done'
$ cat /dev/null > ~/.bash_history && history -c && sudo shutdown -h now

```

Si tout s'est bien passé, la totalité a une taille d'environ 1.6 Go. L'image-disque, une fois tronquée, a une taille d'environ 1.9 Go. Compressée, elle tient dans environ 550 Mo.

12 Acknowledgements

MoodleBox utilise certaines idées de la première version de Christian Westphal,¹⁹ qui mérite pour cela une gratitude particulière.

¹⁹Christian Westphal, Académie de Strasbourg, voir <https://moodle.org/user/view.php?id=1378197&course=20>.