

INSTALL PLAYSMS AND SMSTOOLS ON DEBIAN 11

```
# install other sources to install php 7.2 on debian 11
```

```
apt -y install gnupg2 apt-transport-https ca-certificates software-properties-common
```

```
wget -O /etc/apt/trusted.gpg.d/php.gpg https://packages.sury.org/php/apt.gpg
```

```
echo "deb https://packages.sury.org/php/ $(lsb_release -sc) main" > /etc/apt/sources.list.d/php.list
```

```
apt update && apt install -y apache2 mariadb-server php7.2 php7.2-opcache php7.2-cli php7.2-mysqli  
php7.2-mysql php7.2-gd php7.2-mbstring php7.2-xml php7.2-curl php7.2-zip
```

```
apt-get install apache2 mariadb-server php php-cli php-mysql php-gd php-curl php-mbstring php-xml  
php-zip
```

```
update-alternatives --set php /usr/bin/php7.2
```

Install playSMS 1.4.6 and smstools 3 (compiling it) on Ubuntu 18.04 (ubuntu 20.04 and major cannot compile smstools)

Howto

playSMS version 1.4.6 has been released, and it is the recommended version as it contains fixes to several bugs and critical security vulnerability. This article is howto install playSMS 1.4.6 on Ubuntu 18.04.

I'm using DigitalOcean (DO) service to test the configuration and commands. Create new Droplet in DO account. Click [here](#) to register on DO if you don't have an account.

Choose Ubuntu 18.04 (currently 18.04.3 LTS) and select at least the cheapest service (USD 5). Create and wait for a minute or two for the SSH to be ready. You can then login via SSH and start playSMS installation.

Login to your CentOS droplet (later we will call droplet as server) using SSH and follow instructions below step by step. Read carefully why you need to do each step correctly. Please pay attention to details.

This article first published in: <https://antonraharja.com/2020/03/20/playsms-1-4-3-on-ubuntu-18-04/>

1. Prepare Ubuntu

1.1. Add Normal User

In DO you need to login as `root` first. But it is recommended to not login as `root` all the time, so we create a new normal Linux user.

As `root` create a new normal Linux user and set a strong password for it:

```
adduser playsms
```

Add user `playsms` to `sudo` group:

```
usermod -a -G sudo playsms
```

Q: Can I use other username beside `playsms` ?

Yes, you can. Just remember to adjust every reference of `playsms` in this article into your own chosen username.

1.2. Copy `authorized_keys`

This is additional and **optional** steps you need to do if you're login SSH as `root` using private key instead of password.

Advertisements

[REPORT THIS ADPRIVACY](#)

You need to copy `root`'s `authorized_keys` to `playsms`:

```
sudo mkdir -p /home/playsms/.ssh
```

```
sudo cp /root/.ssh/authorized_keys /home/playsms/.ssh/
```

```
sudo chown -R playsms.playsms /home/playsms
```

After this you can login SSH as user `playsms` using the same private key as `root`.

Q: Can I use different key for `playsms` ?

Yes, you can. Copy the public key (it's public key, not private key) to `playsms`'s authorized_keys` and remove root's public key from it.

1.3. Enable Ubuntu Firewall

Allow SSH first:

```
sudo ufw allow ssh
```

Enable UFW, activate it and make it starts on boot:

```
sudo ufw enable
```

Reload UFW:

```
sudo ufw reload
```

As of now only SSH allowed by server, later we will allow `http` and `https`. Don't forget to `ufw reload` after changing UFW rules.

1.4. Install `mc` , `zip` and `unzip`

Yes. Install `mc` and `unzip` :) I'm using `nano` as console text editor, and you might be checking files/folders frequently, for that I think `mc` helps. But you can always choose not to install it and stick with `nano` or `vi`.

Advertisements

[REPORT THIS ADPRIVACY](#)

You need to install `unzip`, `composer` will need it and `playSMS` will need `composer`.

Install `mc` , `zip` and `unzip`:

```
sudo apt update
```

```
sudo apt install mc zip unzip
```

1.5. Upgrade Server

Update and upgrade:

```
sudo apt update
```

```
sudo apt upgrade
```

Most likely after upgrade Ubuntu asks for server reboot, reboot it then:

```
sudo shutdown -r now
```

Re-login SSH using user `playsms` instead of `root`. Pass this point you need to login to the server as normal user `playsms`, and you will use `sudo` when you need to execute commands as `root`.

2. Install MySQL Server

We will use MariaDB as MySQL server.

If you have not logout out from `root` you need to logout now and re-login as normal user `playsms`.

Install MySQL server MariaDB:

```
sudo apt install mariadb-server
```

Starts MariaDB and enable it:

```
sudo systemctl start mariadb.service
```

```
sudo systemctl enable mariadb.service
```

Test your MySQL root access:

```
sudo mysql
```

You should now logged in to your MySQL server as MySQL user `root`. Type `quit` and <Enter> to exit MySQL console.

Advertisements

[REPORT THIS ADPRIVACY](#)

Note that you cannot login to MariaDB as MySQL user `root` if you are not Linux user `root`. Use `sudo` to access MySQL server as MySQL user `root`, you won't be asked for password.

We will not use MySQL user `root` in playSMS but we will create a new MySQL user just for playSMS database later.

3. Install Web Server and PHP 7.2

We will use Apache2 as the web server.

Install Apache2, PHP 7.2 and required PHP modules:

REMEMBER, IF INSTALL ON DEBIAN 11, pass step XXX

Have installed iet php 7.2?!?!?

((apt update && apt install -y apache2 mariadb-server php7.2 php7.2-opcache php7.2-cli php7.2-mysqli php7.2-mysql php7.2-gd php7.2-mbstring php7.2-xml php7.2-curl php7.2-zip)))

```
sudo apt install apache2 php php-cli php-mysql php-gd php-curl php-mbstring php-xml php-zip
```

*****STEP XXX*****

Start Apache2 and enable it:

```
sudo systemctl start apache2.service  
sudo systemctl enable apache2.service
```

Allow HTTP and HTTPS:

```
sudo ufw allow http  
sudo ufw allow https  
sudo ufw reload
```

Let's test the PHP:

```
cd /var/www/html  
sudo nano test.php  
<?php  
echo "Hello World";
```

Save `test.php` and browse the file, you should *Hello World* displayed.

Advertisements

[REPORT THIS ADPRIVACY](#)

Remove ``test.php`` after testing:

```
sudo rm -f /var/www/html/test.php
```

4. Supports HTTPS

HTTPS supports will be added to our web server by requesting, installing and configuring SSL certificate from [Let's Encrypt](#) on Apache2. Let's Encrypt provides a free SSL certificate for everyone.

4.1. Setup VirtualHost

This step is required for getting free SSL certificate for our HTTPS service from Let's Encrypt.

In this example I will be using `dm143.playsms.org` domain as my entry in VirtualHost setup. I also have set the DNS to point `dm143.playsms.org` to my CentOS server's public IP. Of course you will need your own domain/subdomain and point to your own CentOS server's public IP.

The example VirtualHost configuration will make Apache serve PHP file for domain `playsms` from our regular user (user `playsms`) Home Directory (`/home/playsms/public_html` to be exact).

Prepare user's Home Directory:

```
cd /home/playsms

mkdir -p public_html log

sudo chmod 775 /home/playsms public_html log

sudo chown playsms.playsms -R /home/playsms

sudo chown www-data.playsms -R /home/playsms/log

ls -l /home/playsms
```

Create VirtualHost configuration file for domain `dm143.playsms.org`:

```
sudo nano /etc/apache2/sites-enabled/000-default.conf
```

```
<VirtualHost *:80>
    ServerName playsmm
    DocumentRoot /home/playsms/public_html
    ErrorLog /home/playsms/log/httpd-error.log
    CustomLog /home/playsms/log/httpd-access.log combined
    <Directory /home/playsms/public_html>
        AllowOverride FileInfo AuthConfig Limit Indexes
        Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
        Require method GET POST OPTIONS
        php_admin_value engine On
    </Directory>
</VirtualHost>
```

Enable it:

```
sudo systemctl reload apache2.service
```

Switch user as user **playsms** and test VirtualHost by create a PHP file in `/home/playsms/public_html`.

```
nano /home/playsms/public_html/test.php
```

```
<?php
echo "<b>Welcome !!</b>";
```

Save the file and browse this file at your domain, in this example browse `http://ip_of_your_playsms_machine/test.php`

Advertisements

[REPORT THIS ADPRIVACY](#)

You know your VirtualHost is working when you see **Welcome !!** on your browser.

Remove `test.php` after testing:

```
rm -f /home/playsms/public_html/test.php
```

4.2. Install certbot (not needed...)

We will get the SSL certificate from Let's Encrypt and use `certbot` to install it on the server.

Install certbot:

```
sudo apt install python3-certbot-apache
```

4.3. Setup SSL Certificate

Run certbot for Apache:

```
sudo certbot --apache
```

Answer questions correctly. You will need to input your email address, choose **A** to Agree with the ToS and last choose **Redirect** (selection no. 2) to completely remove HTTP and just serve HTTPS by redirecting all HTTP requests to HTTPS.

Example of successful SSL certificate request and installation:

Visit ssllabs.com/ssltest and submit your domain to test your HTTPS configuration.

5. Install playSMS (needed!!!)

Now that we have a working web server with PHP and HTTPS supports, and MySQL server, we can then install playSMS 1.4.6.

Advertisements

[REPORT THIS ADPRIVACY](#)

From now on you must execute commands as normal Linux user. In this article playSMS will be installed under user `playsms` as mentioned before.

5.1. Prepare Directories

Here are some important directories that need to be ready before playSMS installation:

`public_html` and `log` is already exists and prepared, they are created previously on section 4.1 as part as VirtualHost configuration. So now we need to create the rest and set proper permission.

Then create directories:

```
cd /home/playsms  
  
mkdir -p bin etc lib src  
  
sudo chmod 775 bin etc lib src
```

Prepare log files too, this need to be done so that both web server Apache2 and playSMS daemon have write access to playSMS log files:


```
cd /home/playsms

sudo touch log/audit.log log/playsms.log

sudo chmod 664 log/audit.log log/playsms.log

sudo chown www-data:playsms -R log

ls -l log
```

5.2. Check PHP Modules

Required PHP modules should already be installed if you follow this article from the start, it is on section 3. But before proceeding with playSMS installation you need to make sure that required PHP modules are installed:

```
php -m
```

Make sure you see at least `curl`, `gd`, `mbstring`, `mysqli` and `xml` on the list. If they are not on the list then please install them, see section 3.

5.3. Prepare Database

Create MySQL database that will be used by playSMS:

```
sudo mysqladmin create playsms
```

Login as MySQL user root and create a new MySQL user for above database:

```
sudo mysql
```

```
1 CREATE USER 'playsms'@'localhost' IDENTIFIED BY 'strongpasswordhere';
2 GRANT ALL PRIVILEGES ON playsms.* TO 'playsms'@'localhost';
3 FLUSH PRIVILEGES;
4 exit
```

Do not copy-paste above SQL commands directly to MySQL console, you must use your own strong password, change the `strongpasswordhere` with your own strong password.

Advertisements

[REPORT THIS ADPRIVACY](#)

As of this section you will have a MySQL database named `playsms` and MySQL normal user `playsms` with your own strong password which only have access to database `playsms`.

5.4. Get playSMS Source Code

playSMS source code available on Github, you will need `git` to get them.

Go to src folder:

```
cd /home/playsms/src
```

Get playSMS version 1.4.6:

```
git clone -b 1.4.x --depth=1 https://github.com/antonraharja/playSMS
```

As of now your playSMS 1.4.6 source code is available at `/home/playsms/src/playSMS`.

5.5. Prepare install.conf

Go to playSMS source code directory, copy `install.conf.dist` to `install.conf` and then edit it.

Go to playSMS source code directory:

```
cd /home/playsms/src/playSMS
```

Edit `install.conf`:

```
cp install.conf.dist install.conf
```

```
nano install.conf
```

These are values I set on `install.conf`:

```
DBUSER="playsms"
DBPASS="strongpasswordhere"
DBNAME="playsms"
DBHOST="localhost"
DBPORT="3306"
WEBSERVERUSER="www-data"
WEBSERVERGROUP="www-data"
PATHSRC="/home/playsms/src/playSMS"
PATHWEB="/home/playsms/public_html"
PATHLIB="/home/playsms/lib"
PATHBIN="/home/playsms/bin"
PATHLOG="/home/playsms/log"
PATHCONF="/home/playsms/etc"
```

Values need to reflect your server configuration. If you follow this article from the start then above values should be correct, with exception your true database password (DBPASS) of course.

Save `install.conf` and ready to run install script.

5.6. Run playSMS Install Script

playSMS install script will download `composer` and download packages from `repo.packagist.org`. After that the script will copy necessary files from playSMS source code to `public_html` and `bin`.

Since theres requirement to be able to download from external site (`repo.packagist.org`), you have to make sure that external site is working and reachable.

But you can just start the install script, because you'll know if something not right, for example the script fail to download packages. When that happens you can fix the problem first, like fix your networking setup and perhaps firewall, or simply wait (theres a chance the external site down too), and then go back to re-run the install script.

Just to make sure that networking stuff is right, please see section 1.6.

OK, let's start the installation:

```
cd /home/playsms/src/playSMS
./install-playsms.sh
```

Verify installation:

Press **Y** (you will be asked twice, answer Y both) and proceed the installation.

Advertisements

[REPORT THIS ADPRIVACY](#)

Successful installation will show that all playSMS daemon is running:

Browse your playSMS, don't worry if the login page looks broken, it's because we haven't configure playSMS to enable HTTPS, we will do that after this. For now, check if you can see playSMS login page.

5.7. Adjust config.php

Edit playSMS `config.php` and adjust some value, or just one part, the HTTPS support.

```
nano /home/playsms/public_html/config.php
```

Inside `config.php`:

- Search for `logstate` and set it to `3`
- Search for `ishttps` and set it to `true`. (if prefer https and not only http)

Daemon result red color on web page: go in this file and modify:

`/home/playsms/public_html/plugin/feature/playsmslog/config.php`

From:

```
$plugin_config['playsmslog']['playsmsd']['bin'] = '/home/playsms/bin/playsmsd';
$plugin_config['playsmslog']['playsmsd']['conf'] = '/home/playsms/etc/playsmsd.conf';
```

to:

```
$plugin_config['playsmslog']['playsmsd']['bin'] = '/home/playsms/bin/playsmsd';
$plugin_config['playsmslog']['playsmsd']['conf'] = '/home/playsms/etc/playsmsd.conf';
```

save, all green now!!!

Edit also this file to remove (for me is +393xxx)

`/home/playsms/public_html/plugin/core/sendsms/fn.php`

After this line:

```
if (is_array($user)) {
    $prefix = ($user['replace_zero'] ? $user['replace_zero'] :
$core_config['main']['default_replace_zero']);
    $local_length = (int) $user['local_length'];
    _log('before prefix manipulation:[' . $number . ']', 3,
'sendsms_manipulate_prefix');
```

//add here//

```
    $number = str_replace('+', '', $number);
    if (substr($number, 0, 3) == '393') {
        _log('my own prefix manipulation not need:[' . $number . ']', 3,
'sendsms_manipulate_prefix');
    } else {
        _log('my own prefix manipulation needed changing it:[' . $number .
']', 3, 'sendsms_manipulate_prefix');
        $number = '39' . $number;
        _log('my own prefix manipulation needed:[' . $number . ']', 3,
'sendsms_manipulate_prefix');
    }
}
```

Save

Install Playams service:

on console write:

su root

(enter root password)

```
cat >> /etc/systemd/system/playsms.service << EOF
[Unit]
Description=playsms
After=mariadb.service
[Service]
Type=oneshot
RemainAfterExit=yes
ExecStart=/home/playsms/bin/playsmsd /home/playsms/etc/playsmsd.conf start
ExecStop=/home/playsms/bin/playsmsd /home/playsms/etc/playsmsd.conf stop
User=www-data
Group=www-data
[Install]
WantedBy=multi-user.target
EOF
```

Enable and execute playsms service:

```
chmod 755 /home/playsms/bin/playsmsd
systemctl daemon-reload
systemctl enable playsms
systemctl restart playsms
systemctl status playsms
```

5.8. Change Default Password

Go to your browser, browse the server and login as playSMS administrator, and change the default admin password immediately.

If you use release > ubuntu 18.04, for example debian 11, follow this instructions and then go to STEP “B”

Install smstools by command:

sudo apt install smstools, and then

```
sudo update-rc.d -f smstools disable
```

```
sudo update-rc.d -f smstools remove
```

go on my repository and download as zip the fix:

<https://github.com/pappicio/sfstools3-for-debian-11>

unzip and copy all folders/files in your root debian 11 system by winscp tool
(<https://winscp.net/download/WinSCP-6.1.1-Setup.exe>)

provide to set for all folders/subfolder/files, 0777 permission and as user/group: www-data, and at end execute:

```
sudo systemctl daemon-reload
```

```
sudo update-rc.d -f sms3 defaults
```

Go to step “B” now:

5.9 Install smstools last release

```
apt-get install build-essential libusb-1.0 libusb-1.0-0-dev build-essential manpages-dev
```

```
sudo apt-get update & sudo apt-get install usb-modeswitch
```

```
cd /tmp/
```

```
wget http://smstools3.kekekasvi.com/packages/sfstools3-3.1.21.tar.gz
```

```
tar -zxf smstools3-3.1.21.tar.gz -C /usr/local/src
```

```
ls -l /usr/local/src/
```

```
cd /usr/local/src/sfstools3/
```

make

make install

STEP “B”

5.8. compiled smstools, now configure:

```
mkdir -p /var/log/sms/stats
```

```
mkdir -p /var/spool/sms/ {checked,failed,incoming,outgoing,sent}
```

```
mkdir /var/spool/sms/modem1
```

```
chown www-data:www-data -R /var/spool/sms
```

```
chmod 777 -R /var/spool/sms
```

```
mv /etc/smsd.conf /etc/smsd.conf.dist
```

```
cd /tmp
```

```
wget -c
```

```
https://raw.githubusercontent.com/antonraharja/playSMS/master/contrib/smstools/smsd.conf
```

```
cp smsd.conf /etc/
```

configure your own, my is:

```
devices = modem1
loglevel = 7
# logfiles
stats = /var/log/sms/stats
logfile = /var/log/sms/smsd.log
# Default queue directory = /var/spool/sms
outgoing = /var/spool/sms/outgoing
checked = /var/spool/sms/checked
failed = /var/spool/sms/failed
incoming = /var/spool/sms/incoming
sent = /var/spool/sms/sent
delaytime = 2
errorsleeptime = 10
blocktime = 180
autosplit = 3
# Queue configurations
```

```
[queues]
modem1 = /var/spool/sms/modem1

[modem1]
device = /dev/ttyUSB1
init = AT^CURC=0
###init2 = AT+CPMS="ME", "ME", "ME"
#pin = 1234
report = yes
incoming = yes
queues = modem1
# mode = new
smc = 393770001016
baudrate = 115200 ###19200
memory_start = 0
decode_unicode_text = yes
#cs_convert = yes
report_device_details = no
```

...and finally...

```
sudo update-rc.d sms3 defaults
```

```
sudo reboot
```

and all works!!!

Block usb modem on same ttyUSBX for ever:

```
-----
Get deviceid for the dongle
```

```
sudo lsusb
```

Get to know properties of the device while it is switched in:

```
udevadm info -q all -p $(udevadm info -q path -n /dev/ttyUSB1)
```

(if your system is old, try instead with this command:)

```
( udevinfo -a -p $(udevinfo -q path -n /dev/ttyUSB0) )
```

Find some property that can identify the device (uniquely), for instance "serial"

Create a file called

`/etc/udev/rules.d/10-usb-serial`

which contains the line:

```
BUS=="usb", ATTR{serial}=="xxxx", NAME="ttyUSB1"
```

Note the two equal signs for properties that are tested, and one for that which is assigned to.