

INSTALLATION D'UN SERVEUR LIBRENMS SOUS RASPBERRY PI

Raspberry Pi - Debian Bullseye
Configuration de base

Tutoriel **LIBRENMS** - RASPBERRY PI

David GOÏTRÉ

Table des matières

Introduction	1
1. Pré requis	1
2. Connexion au serveur	1
3a. Paramétrage Ethernet du serveur	2
3b. Paramétrage Wifi du serveur	3
4. Installer LibreNMS	3
5. Créer et configurer MariaDB	4
6. Configurer PHP-PFM	5
7. Configurer le serveur WEB	5
8. Activer l'achèvement de la commande Inms.....	6
9. Activer la gestion des journaux de LibreNMS	6
10. Vérification et dépannage	6
11. Installer LibreNMS	7
12. Sauvegarde et restauration de la base de données.....	9
13. Optimisation des performances	10
14. Aides et support	10
15. Mise à jour LibreNMS	10
16. Commandes RaspberryPi.....	10
17. Conclusion	10

Introduction

LibreNMS est un système de supervision réseaux qui possèdent les fonctionnalités suivantes :

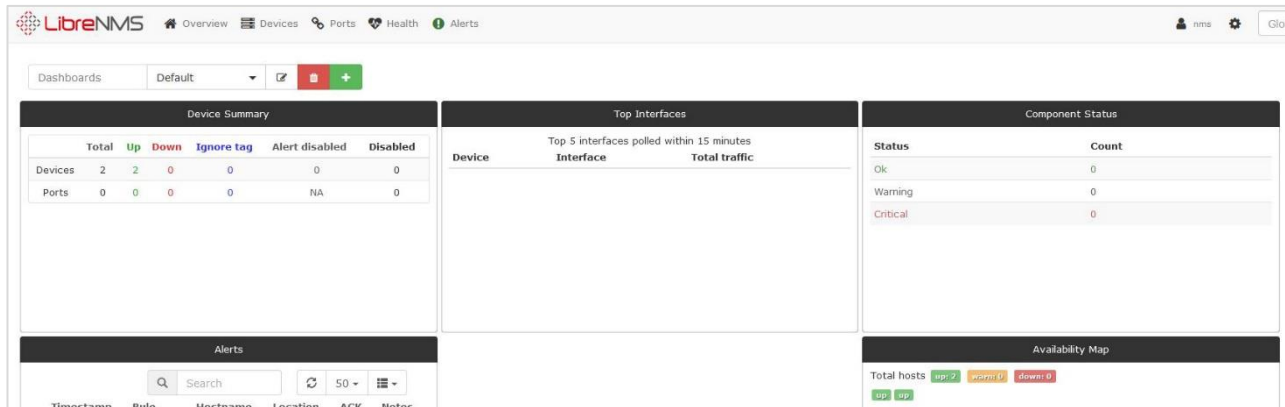
- Une découverte automatique du réseau en utilisant CDP, FDP, LLDP, OSPF, BGP, SNMP et ARP.
- Un système d'alerte flexible.
- Une interface très complète pour gérer, représenter et extraire les données (graphiques, tableaux ...)
- Des mises à jour automatique pour corriger des bugs ou ajouter de nouvelles fonctionnalités
- Et de nombreuses autre possibilités.

1. Pré requis

On a besoin des différents matériels et logiciels pour la création d'un Serveur LIBRENMS avec un RaspberryPi.

- Un ou des PC client sous Windows
- Une Box (Free, Orange, Sfr...)
- Un Raspberry 3B+ avec l'OS Raspian Bullseye installé avec [Etcher](#)
- Le logiciel [Putty](#) pour se connecter en SSH au serveur VPN
- Connaitre l'interface réseau (eth0, br0, ens3...) via la commande : **ip a**
Pour notre test c'est **l'interface eth0** qui sera utilisée

Voici l'interface que l'on doit obtenir une fois connecter au serveur **LibreNMS** mise en place



2. Connexion au serveur

a) Activer le **SSH** sur le serveur. Pour ce faire, ouvrir la carte SD du RaspberryPi via l'explorateur de Windows et créer un fichier **ssh** (sans extension) à sa racine.

b) Ouvrir **Putty** et se connecter au serveur LibreNMS avec les identifiants (par défaut **pi/raspberry**)

c) Mettre à jour les packages du système vers la dernière version. Exécuter la commande suivante pour mettre à jour et mettre à niveau les packages de votre système :

```
# apt-get update -y  
# apt-get upgrade -y
```

3a. Paramétrage Ethernet du serveur

Avant d'aller plus loin, il nous faut connaître l'interface réseau de notre serveur **RaspberryPI** et lui attribuer une adresse IP fixe.

a) Lister les interfaces

```
$ ip link | awk '{ print $2}' # liste les interfaces
# ethtool <interface> | grep detected # détecte l'interface connectée
```

b) Définir une adresse IP fixe

```
# nano /etc/network/interfaces # ouvre le fichier des interfaces
```

c) Copier le texte ci-dessous dans le fichier **interfaces**

```
# Interface reseau de bouclage
auto lo
iface lo inet loopback
# Interface reseau principale
allow-hotplug eth0
iface eth0 inet static
address 192.xxx.xxx.xxx
netmask 255.255.255.0
gateway 192.xxx.xxx.xxx
```

d) Rebooter le serveur

```
# /etc/init.d/networking restart
# reboot
```

e) Paramétrer le serveur

```
$ raspi-config # ouvre l'utilitaire, sélectionner le menu System Options
```

```
-----| Raspberry Pi Software Configuration Tool (raspi-config) |-----
1 System Options          Configure system settings
2 Display Options         Configure display settings
```

Sélectionner le menu **S3 Password** pour modifier le mot de passe et **S4 Hostname** pour modifier le nom du serveur.

```
-----| Raspberry Pi Software Configuration Tool (raspi-config) |-----
S1 Wireless LAN          Enter SSID and passphrase
S2 Audio                  Select audio out through HDMI or 3.5mm jack
S3 Password               Change password for the 'pi' user
S4 Hostname               Set name for this computer on a network
```

3b. Paramétrage Wifi du serveur

Par défaut le Wifi est désactivé. Il faut créer un fichier **wpa_supplicant.conf** et le copier à la racine de la carte SD, permettant à Raspberry Pi OS de lire le fichier au prochain démarrage et d'appliquer la configuration directement.

a) ouvrir un éditeur de texte et copier le texte suivant

```
country=FR
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="NOM_RESEAU"
    scan_ssid=1 #nécessaire quand le ssid n'est pas diffuser
    psk="MOTDEPASSE"
    key_mgmt=WPA-PSK
}
```

b) Modifier les champs du **SSID** et **PSK**

c) enregistrer le fichier sous le nom **wpa_supplicant.conf** et copier-le à la racine de la carte SD

4. Installer LibreNMS

Par défaut, le paquet **LibreNMS** n'est pas disponible dans le référentiel de Debian 11. Il faut :

a) installer les packages nécessaires avec la commande suivante

```
# apt install acl curl composer fping git graphviz imagemagick mariadb-client mariadb-server mtr-tiny nginx-full nmap php7.4-cli php7.4-curl php7.4-fpm php7.4-gd php7.4-gmp php7.4-json php7.4-mbstring php7.4-mysql php7.4-snmp php7.4-xml php7.4-zip python3-dotenv python3-pymysql python3-redis python3-setuptools python3-systemd rrdtool git nagios-plugins snmp snmpd whois
```

b) Ajouter un utilisateur à librenms (ex : adminnms)

```
# useradd nomutilisateur -d /opt/librenms -M -r -s "$(which bash)"
```

c) Télécharger librenms

```
# cd /opt
# git clone https://github.com/librenms/librenms.git
```

b) Définir des autorisations

```
# chown -R nomutilisateur:nomutilisateur /opt/librenms
# chmod 771 /opt/librenms
# setfacl -d -m g::rwx /opt/librenms/rrd /opt/librenms/logs /opt/librenms/bootstrap /cache/ /opt/librenms/storage/
# setfacl -R -m g::rwx /opt/librenms/rrd /opt/librenms/logs /opt/librenms/bootstrap /cache/ /opt/librenms/storage/
```

b) Installer les dépendances

```
# sudo su - nomutilisateur
./scripts/composer_wrapper.php install --no-dev
exit
```

c) Parfois, lorsqu'un proxy est utilisé pour accéder à Internet, le script ci-dessus peut échouer. La solution de contournement consiste à installer le composer package manuellement. Pour une installation globale.

```
# wget https://getcomposer.org/composer-stable.phar
# mv composer-stable.phar /usr/bin/composer chmod +x /usr/bin/composer
```

d) Définir le fuseau horaire en modifiant le champ **date.timezone** du fichier .ini et celui du système

```
# nano /etc/php/7.4/fpm/php.ini - (date.timezone = Europe/paris)
# nano /etc/php/7.4/cli/php.ini - (date.timezone = Europe/paris)
# timedatectl set-timezone Etc/UTC
```

5. Créer et configurer MariaDB

Dans la dernière version de Linux Debian 11, on utilise plus Apache2 mais **Nginx** et **MariaDB**

a) Editer le fichier **server.cnf** et ajouter dans la section **[MySqlId]** les lignes

```
# nano /etc/mysql/mariadb.conf.d/50-server.cnf
- innodb_file_per_table = 1 - ligne à rajouter
- lower_case_table_names = 0 - ligne à rajouter
```

b) Redémarrer mariadb

```
# systemctl enable mariadb
# systemctl restart mariadb
```

c) Ouvrir Mysql language

```
# mysql -u root
```

d) Créer la base et le mot de passe utilisateur (modifier les chaînes **nomutilisateur** et **motdepasse**)

```
> CREATE DATABASE nombdd CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
> CREATE USER 'nomutilisateur'@'localhost' IDENTIFIED BY 'motdepasse';
> GRANT ALL PRIVILEGES ON librenms.* TO 'nomutilisateur'@'localhost';
> FLUSH PRIVILEGES;
> exit
```

6. Configurer PHP-PFM

a) Editer le fichier **librenms.conf** et modifier **[www]** par **[nomutilisateur]**

```
# cp /etc/php/7.4/fpm/pool.d/www.conf /etc/php/7.4/fpm/pool.d/librenms.conf
# nano /etc/php/7.4/fpm/pool.d/librenms.conf
```

b) Modifier les lignes **user**, **group** et **listen** comme ci-dessous

```
- user = nomutilisateur
- group = nomutilisateur
- listen = /run/php-fpm-nomutilisateur.sock
```

c) Supprimer le fichier **www.conf** pour économiser des ressources, s'il n'y a pas d'autres applications Web sur ce serveur. Ne pas hésitez pas à ajuster les paramètres de performances dans le fichier **librenms.conf** pour répondre aux besoins.

7. Configurer le serveur WEB

a) Créer le fichier **librenms.vhost**

```
# nano /etc/nginx/sites-enabled/librenms.vhost
```

b) Ajouter le texte ci-dessous au fichier de conf et saisir une valeur pour le champ **server_name** et le **nomutilisateur**.

```
server {
    listen 80;
    server_name @ip ou nom du serveur;
    root /opt/librenms/html;
    index index.php;

    charset utf-8;
    gzip on;
    gzip_types text/css application/javascript text/javascript application/x-javascript
    image/svg+xml text/plain text/xsd text/xsl text/xml image/x-icon;
    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }

    location ~ [^/]\.php(/|$) {
        fastcgi_pass unix:/run/php-fpm-nomutilisateur.sock;
        fastcgi_split_path_info ^(.+\.(php))(/.+)$;
        include fastcgi.conf;
    }
    location ~ /\.(!well-known).* {
        deny all;
    }
}
```

c) Effacer la configuration des sites par défaut et relancer les services **Nginx** et **PHP**

```
# rm /etc/nginx/sites-enabled/default
# systemctl reload nginx
# systemctl restart php7.4-fpm
```

8. Activer l'achèvement de la commande Inms

Cette fonctionnalité vous donne la possibilité d'utiliser l'onglet pour la complétion sur les commandes **Inms** comme on le ferait pour les commandes Linux normales.

```
# ln -s /opt/librenms/Inms /usr/bin/Inms
# cp /opt/librenms/misc/Inms-completion.bash /etc/bash_completion.d/
```

a) Configurer **SNMPD** et modifier la chaîne de caractère **RANDOMSTRINGGOESHERE** et définir son propre nom de communauté (ex : WORKSGROUP).

```
# cp /opt/librenms/snmpd.conf.example /etc/snmp/snmpd.conf
# nano /etc/snmp/snmpd.conf
```

b) Donner des droits et relancer le service **snmpd**

```
# curl -o /usr/bin/distro https://raw.githubusercontent.com/librenms/librenms-agent/master/snmp/distro
# chmod +x /usr/bin/distro
# systemctl enable snmpd
# systemctl restart snmpd
```

c) Créer une tâche planifiée

```
# cp /opt/librenms/librenms.nonroot.cron /etc/cron.d/librenms
```

9. Activer la gestion des journaux de LibreNMS

LibreNMS conserve les logs **/opt/librenms/logs**. Pour activer la gestion de ceux-ci utiliser la commande ci-dessous :

```
# cp /opt/librenms/misc/librenms.logrotate /etc/logrotate.d/librenms
```

10. Vérification et dépannage

b) Vérifier l'installation finale

```
# sudo su - librenms
# ./validate.php
```


On doit obtenir la sortie suivante :

```
librenms@raspberrypi:~$ ./validate.php
=====
Component | Version
----- | -----
LibreNMS  | 21.11.0-31-gcdd1ae0d1
DB Schema | Not Connected (0)
PHP       | 7.4.25
Python   | 3.9.2
MySQL    | ?
RRDTool  | 1.7.2
SNMP     | 5.9
=====

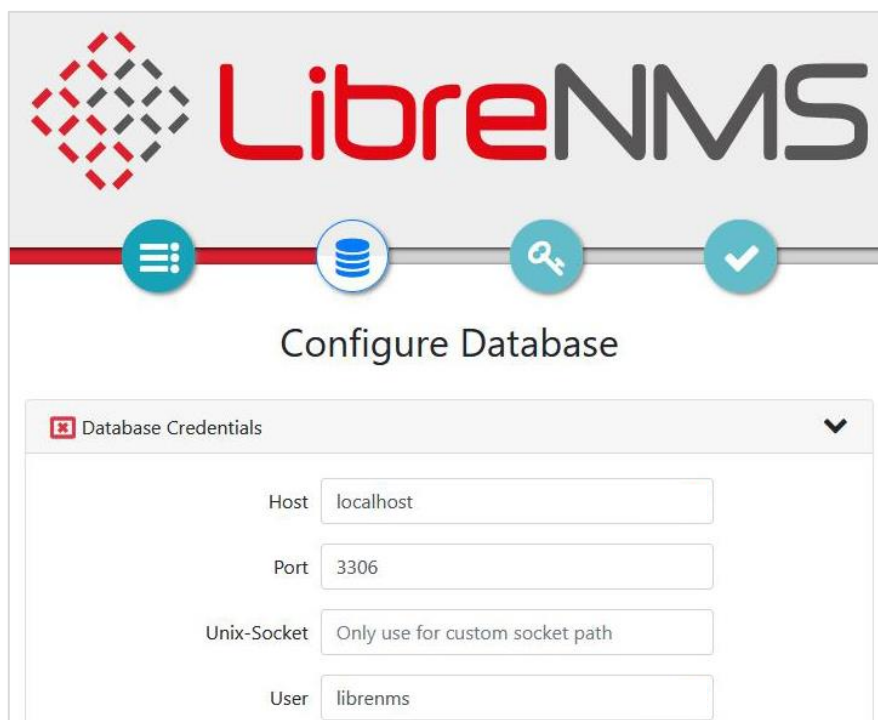
[OK] Composer Version: 2.1.12
[OK] Dependencies up-to-date.
Could not connect to database, check logs/librenms.log.
```

11. Installer LibreNMS

Le programme Web créer automatiquement un fichier **config.php** à la fin de l'installation à l'emplacement **/opt/librenms/config.php**. Ne pas oublier pas de définir les autorisations sur ce fichier. Au besoin, définir le plugin Nagios avec la 2^{ème} commande.

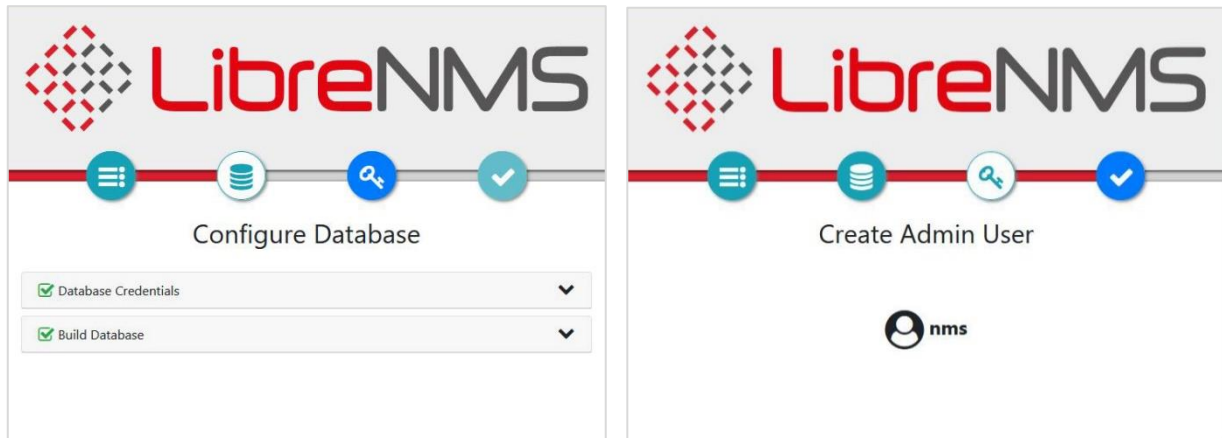
```
# chown nomutilisateur:nomutilisateur /opt/librenms/config.php
# $config['nagios_plugins'] = '/usr/lib/nagios/plugins';
```

a) Ouvrir la page avec **@IPduServeur/install.php** dans le navigateur pour effectuer l'installation de **LibreNMS**. L'écran ci-dessous s'affiche. Saisir les identifiants créer pendant la création de la base.

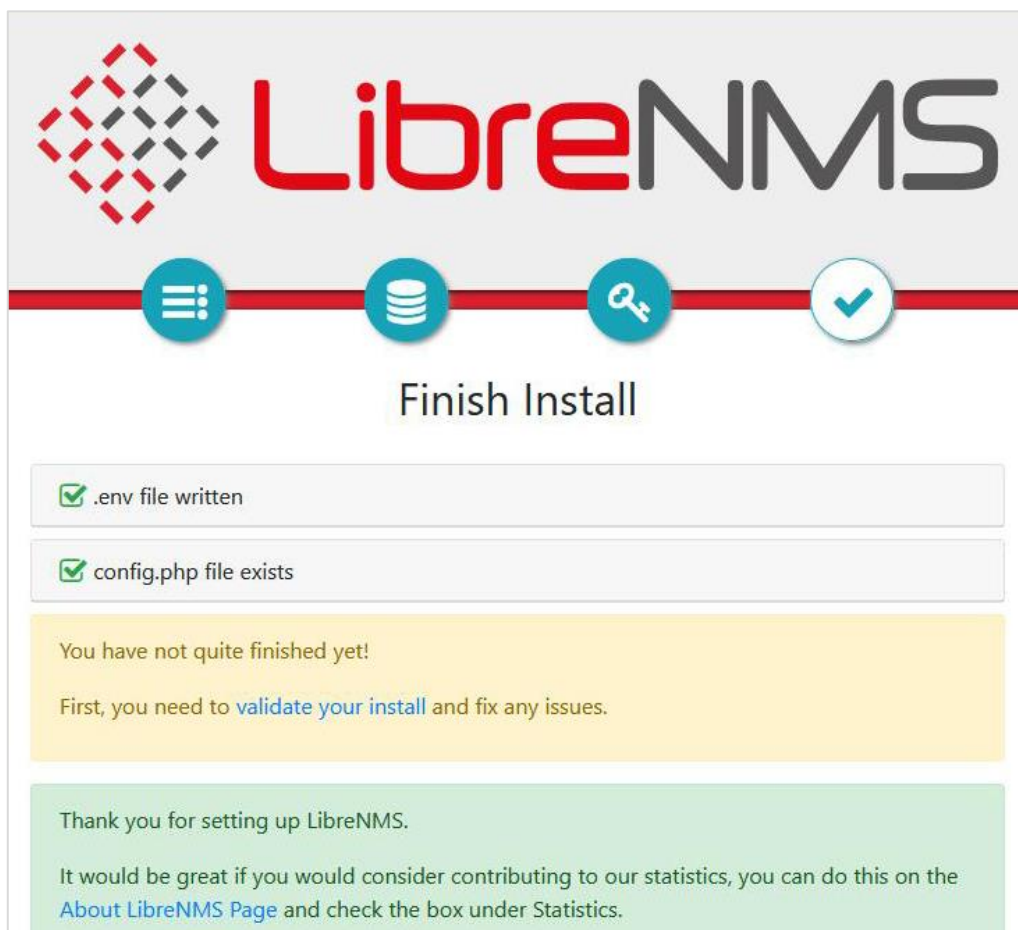


The screenshot shows the LibreNMS web interface for configuring the database. At the top, there is the LibreNMS logo and a navigation bar with four icons: a menu, a database, a key, and a checkmark. Below the navigation bar, the title "Configure Database" is centered. A form titled "Database Credentials" is displayed, containing four input fields: "Host" (localhost), "Port" (3306), "Unix-Socket" (Only use for custom socket path), and "User" (librenms).

b) La base a été construite, cliquer sur l'icône de la clé pour continuer, puis sur l'icône de la coche pour finir



c) L'écran final s'affiche. Cliquer sur le lien **Validate your install** pour finaliser l'installation.



d) L'écran de connexion s'affiche. Saisir les **identifiants** pour se connecter à LibreNMS



e) Ajouter **localhost** comme premier appareil à partir de l'interface Web (recommandé).

12. Sauvegarde et restauration de la base de données

Il peut être utile de sauvegarder sa base de données à des fins de restauration sur une nouvelle machine par exemple.

a) Ouvrir une connexion **ssh** au serveur Librenms

b) saisir les commandes ci-dessous :

```
# mysql -u root
# show databases; # permet de lister les bases de données
# exit
# mysqldump -u nomutilisateur -p --databases librenms --complete-insert --result-
file=/var/www/librenmsbdd.sql # sauvegarde la base
```

c) Récupérer le fichier sql via Filezilla

d) Pour restaurer la base de données

```
# mysql -u root
# create database nombdd; # crée la base de données
# use nombdd;
# exit
# mysql -u root -p nombdd < /var/ww/nombdd.sql #réinjecte le sql dans la base
```

13. Optimisation des performances

a) Désactiver le Wifi et le bluetooth

```
# echo "dtoverlay=disable-wifi" | sudo tee -a /boot/config.txt
# echo "dtoverlay=disable-wifi" | sudo tee -a /boot/config.txt
# systemctl disable hciuart
```

b) Vérifier le fonctionnement

```
# systemctl status bluetooth - (bluetooth, doit renvoyer inactive)
# ip a - (wifi, n'affiche que les interfaces ethernet)
```

14. Aides et support

- Support : <https://www.librenms.org.translate.goog#support>
- Optimisation et performance : <https://docs.librenms.org.translate.goog/Support/Performance>
- Découverte automatique : <https://docs.librenms.org/Extensions/Auto-Discovery>
- Dépannage déverrouillage de fichier : <https://itsfoss.com/could-not-get-lock-error>
- Configurations complémentaires : <https://ressources.labomedia.org>

15. Mise à jour LibreNMS

Pour mettre à jour LibreNMS manuellement, exécuter les commandes suivantes

```
# su - librenms
# git pull
# ./daily.sh
```

16. Commandes RaspberryPi

a) Liste des commandes basique à la gestion du serveur RaspberryPi

```
# shutdown -h now # éteint le serveur en toute sécurité
# shutdown -r now # redémarre le serveur en toute sécurité
# apt install openssh-server # installe le SSH
# systemctl enable sshd.service # active le service SSH au démarrage
##### Désactive la mise en veille #####
# systemctl mask sleep.target suspend.target hibernate.target hybrid-sleep.target
```

17. Conclusion

LibreNMS est installé et configuré avec succès sur le serveur **RaspberryPi Debian 11**. On peut désormais accéder à Internet en toute sécurité et protéger son identité.

Destiné au RaspberryPi (Raspbian), **LibreNMS** fonctionne aussi parfaitement sur une distribution Ubuntu, Debian 10, CentOS.

Pour **Debian 10**, voir le lien : <https://installation-librenms-debian-10-apache>

Pour **CentOS**, voir le lien : <https://websetnet.net/fr/how-to-install-librenms-with-nginx-on-centos7>

Autre lien pour **Ubuntu** : <https://lucasvidelaine.com/supervision-librenms>