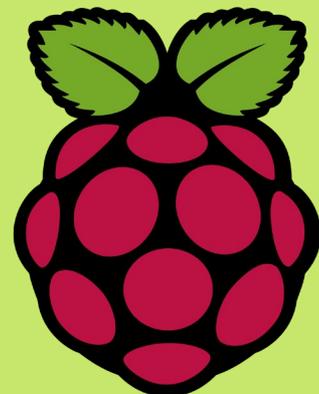


# Installation d'un worker sur un Raspberry



POWERED BY  
Raspberry Pi

## OBJECTIF :

Mettre en place de A à Z un worker avec mod\_gearman sur un Raspberry Pi 4b

REDACTION : mars 2020

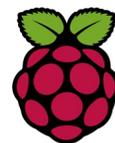
VERSION : 1.0

PAR : GUENAT Lilian (Stagiaire)



# Sommaire

1	PREPARATION .....	3
1.1	Montage .....	3
1.2	L'ISO .....	3
2	CONFIGURATION .....	4
3	INSTALLATION .....	6
3.1	Nagios .....	6
3.2	Mod_Gearman.....	7
3.3	Mises à jour .....	8
3.4	Aide .....	9



# 1 PREPARATION

## 1.1 Montage

Avant tout de chose, il faut d'abord préparer notre Raspberry. Ici, nous sommes en possession d'un Raspberry Pi 4b avec 2 Go de RAM.



On place la carte à l'intérieur de la boîte puis on la fixe à l'intérieur. Ensuite, on fixe le ventilateur sur la partie supérieur de la boîte.

On branche le ventilateur sur la carte puis on referme la boîte.

Notre Raspberry Pi est maintenant prêt à l'emploi. Branchons-lui un câble réseau.



## 1.2 L'ISO

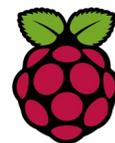
Il est maintenant temps de préparer notre carte micro SD avec le bon ISO. Pour des raisons de simplicité, nous allons utiliser ici un ISO de Raspbian avec quelques applications installées dessus.

Pour se faire, rendez-vous sur le site de Raspberry Pi ici : <https://www.raspberrypi.org/downloads/raspbian/>.

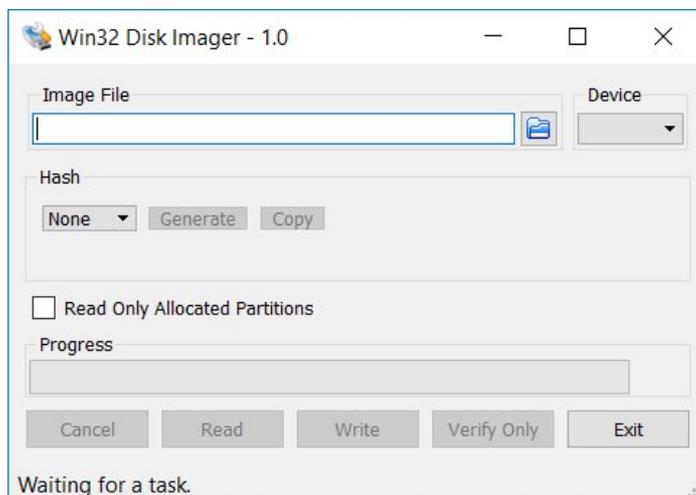
Prenez la version « Raspbian Buster with desktop and recommended software », téléchargez-là.

Maintenant, munissez-vous du logiciel « Win32 Disk Imager » et d'une carte SD (de 16 Go de préférence).

Voici le lien pour le télécharger : <https://sourceforge.net/projects/win32diskimager/>.



Pour installer l'ISO sur la carte micro SD, lancez Win32 Disk Imager et choisissez l'ISO qui vient d'être téléchargé (celle de Raspbian). Sélectionner à présent votre carte micro SD au niveau du périphérique puis cliquez sur « Ecrire ».



Une fois l'écriture de votre carte terminée, vous pouvez l'insérer dans votre Raspberry. Vous pouvez maintenant mettre sous tension votre Raspberry avec l'alimentation (et le câble réseau, déjà fait précédemment normalement).

## 2 CONFIGURATION

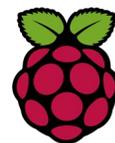
Lors du premier démarrage, vous avez besoin de relier votre Raspberry à un écran ainsi qu'un clavier et une souris car le SSH n'est pas disponible dès le départ.

Une fois cela fait, vous allez donc arriver sur le bureau de Raspbian. Un premier assistant va apparaître au centre, suivez les quelques étapes demandées, choix de la langue, région et disposition du clavier. A la fin de l'assistant, il vous sera demandé si vous souhaitez oui ou non faire les dernières mises à jour. Cliquez sur oui, elles ne seront plus à faire après.

On va donc vous demander de redémarrer le Raspberry, faites-le.

Maintenant, il est conseillé de faire une chose, ajouter la place restante de votre carte micro SD à la partition principale de Raspbian. Pour se faire, exécutez la commande suivante dans la console (que vous pouvez ouvrir depuis la barre de tâche au haut) :

- ▶ `sudo raspi-config`



Vous allez arriver sur ce menu :

```
Raspberry Pi 4 Model B Rev 1.1

Raspberry Pi Software Configuration Tool (raspi-config)

1 Change User Password Change password for the 'pi' user
2 Network Options      Configure network settings
3 Boot Options         Configure options for start-up
4 Localisation Options Set up language and regional settings to match your location
5 Interfacing Options  Configure connections to peripherals
6 Overclock            Configure overclocking for your Pi
7 Advanced Options    Configure advanced settings
8 Update              Update this tool to the latest version
9 About raspi-config  Information about this configuration tool

<Select>                <Finish>
```

Rendez-vous dans « 7 Advanced Options » puis aller sur « A1 Expand Filesystem ».

```
Raspberry Pi Software Configuration Tool (raspi-config)

A1 Expand Filesystem Ensures that all of the SD card storage is available to the OS
A2 Overscan          You may need to configure overscan if black bars are present on display
A3 Memory Split      Change the amount of memory made available to the GPU
```

On va également vous demandez de redémarrer mais juste avant vous pouvez changer le hostname du Raspberry. Retournez au menu principal du raspi-config puis maintenant rendez-vous dans « 2 Network Options » puis vous pouvez modifier le hostname en allant sur « N1 Hostname ». Ici, le hostname est maintenant 'Wrk-AxBB'.

```
Raspberry Pi Software Configuration Tool (raspi-config)

N1 Hostname          Set the visible name for this Pi on a network
N2 Wi-fi             Enter SSID and passphrase
N3 Network interface names Enable/Disable predictable network interface names
N4 Network proxy settings Configure network proxy settings
```

Maintenant, redémarrez-le avec un 'reboot' dans la console, ça ira plus vite.

Par ailleurs, il est également conseillé de mettre une IP fixe sur votre Raspberry afin d'éviter le changement d'IP car s'il change, ça va être compliqué de garder notre worker stable.

Vous pouvez le faire via la commande suivante :

- ▶ `sudo nano /etc/dhcpd.conf`



Ici, vous devez modifier cette partie, commencez par la décommenter puis modifier la configuration IP à votre sauce.

```
# Example static IP configuration:
interface eth0
static ip_address=10.153.54.201/23
static routeurs=10.153.54.253
static domain_name_servers=8.8.8.8 8.8.4.4
```

Une fois fait, sauvegardez le fichier (ctrl+x, O puis Entrée) puis il nous faut redémarrer le service 'networking' avec cette commande :

- ▶ `sudo /etc/init.d/networking restart`

Notre Raspberry est maintenant prêt à accueillir notre worker. Nous allons pouvoir attaquer.

## 3 INSTALLATION

### 3.1 Nagios

Nous allons commencer par installer Nagios avec toutes les dépendances pour le bon fonctionnement des différents scripts.

Pour commencer, refaites cette commande afin de bien s'assurer que tous nos paquets sont à jour :

- ▶ `sudo apt-get update && apt-get upgrade`

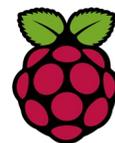
Maintenant, commençons à installer quelques dépendances :

- ▶ `sudo apt-get install -y autoconf automake gcc libc6 libmcrypt-dev make libssl-dev wget bc gawk dc build-essential snmp snmpd libnet-snmp-perl gettext`

Passons maintenant à l'installation des différents paquets nécessaires à la bonne exécution des différents scripts de Nagios :

- ▶ `sudo apt-get install -y nagios-plugins libpqxx3-dev libdbi-dev libfreeradius-client-dev libldap2-dev libmariadbclient-dev libmariadbclient-dev-compat dnsutils smbclient qstat fping`

On peut remarquer par exemple le 'fping' qui sera utile au script 'check\_fping'.



Passons à présent à l'installation de Nagios, suivez simplement les commandes suivantes une par une :

- ▶ `cd /tmp`
- ▶ `sudo wget --no-check-certificate -O nagios-plugins.tar.gz https://github.com/nagios-plugins/nagios-plugins/archive/release-2.3.2.tar.gz`
- ▶ `sudo tar xzf nagios-plugins.tar.gz`
  
- ▶ `cd nagios-plugins-release-2.3.2`
- ▶ `sudo ./tools/setup`
- ▶ `sudo ./configure`
- ▶ `sudo make`
- ▶ `sudo make install`

Afin de voir si votre installation a bien été faite, vous pouvez vous rendre ici :

➔ `cd /usr/local/nagios/libexec` (puis commande 'ls')

Vous devriez avoir tous les scripts nécessaires au bon fonctionnement :

```
pi@Wrk-AxBB:/usr/local/nagios/libexec $ ls
check_apt      check_disk      check_game      check_ircd      check_mrtgtraf  check_ntp_peer  check_radius    check_ssh        check_uptime
check_breeze   check_disk_smb  check_hpjd      check_jabber    check_mysql      check_ntp_time  check_real      check_ssl_validity  check_users
check_by_ssh   check_dns       check_http      check_ldap      check_mysql_query  check_nwstat    check_rpc       check_ssmtp     check_wave
check_clamd    check_dummy     check_icmp      check_ldaps     check_nagios    check_oracle    check_sensors   check_swap      negate
check_cluster  check_file_age  check_ide_smart  check_load     check_ntp       check_overcr    check_simap     check_tcp       remove_perfdata
check_dbi      check_flexlm    check_ifoperstatus  check_log      check_nntp      check_ping      check_smtp      check_time      urlize
check_dhcp     check_fping     check_ifstatus  check_mailq     check_nt        check_pop       check_snmp      check_udp       utils.pm
check_dig      check_ftp       check_imap      check_mrtg     check_ntp       check_procs     check_spop      check_ups       utils.sh
```

⚠ Ne pas oublier de changer la communauté SNMP dans le fichier de configuration `snmpd.conf` (cf. Doc. Eyes Of Network.docx).

### 3.2 Mod\_Gearman

Il est maintenant temps d'installer `mod_gearman`. Pour se faire nous devons installer quelques paquets avant de le configurer.

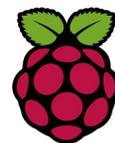
- ▶ `sudo apt-get install mod-gearman-worker gearman-server gearman mod-gearman-module mod-gearman-tools`

Une fois la commande effectuée, nous pouvons déjà vérifier si l'installation est bonne avec la commande :

- ▶ `gearman_top`

A présent, il nous faut tout de même configurer notre worker, éditez le fichier suivant :

- ▶ `sudo nano /etc/mod-gearman/worker.conf`



```
GNU nano 3.2 /etc/mod-gearman/worker.conf
# Example configuration file for the mod gearman worker
#
server=82.243.228.62:4730
key=EONaxBBKey
hosts=yes
services=yes

hostgroups=HG_AxiansBB

pl_file=/usr/share/mod-gearman/mod_gearman_pl.pl
```

Modifiez le fichier en fonction de vos besoins en indiquant l'IP du serveur distant, la clé et les hostgroups que le worker doit gérer.

Maintenant, sauvegardez bien le fichier et pensez à redémarrer le service :

- ▶ `sudo service mod-gearman-worker restart`

Vous devriez pouvoir voir via la commande `mod_gearman` que votre worker fonctionne.

### 3.3 Mises à jour

Pour mettre à jour les MIB, il faut commencer par supprimer tous les fichiers dans le dossier 'mibs' avec cette commande :

- ▶ `rm -rf /usr/share/snmp/mibs/*`

A présent, il faut copier tous les fichiers du dossier 'mibs' du serveur EON dans le dossier 'mibs' de notre worker :

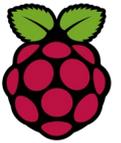
- ▶ `scp -r -p /usr/share/snmp/mibs/ root@192.168.3.62:/usr/share/snmp/mibs/*`

On va ici vous demander un mot de passe pour accéder à serveur EON, le mot de passe a indiqué ici est « Axians2590 ».

Vous pouvez également mettre à jour les plugins de Nagios afin de récupérer les scripts personnalisés ajoutés manuellement sur le serveur EON. Commencez donc par supprimer tous les fichiers du dossier 'plugins' de notre worker :

- ▶ `rm -rf /usr/lib/nagios/plugins/*`

Maintenant il nous reste à copier tous les fichiers du dossier 'plugins du serveur EON dans le dossier 'plugin de notre worker :



► `scp -r -p /usr/lib/nagios/plugins/ root@192.168.3.62:/usr/lib/nagios/plugins/`\*

On va encore vous demander ici le mot de passe du serveur EON (« Axians2590 »).

Si cette façon ne fonctionne pas, vous pouvez transférer tous les fichiers dans /plugins à la main depuis le serveur sur le worker.

Si toutes les étapes présentes ont bien été suivies votre worker devrait pour voir communiquer sans problème avec le serveur.

### 3.4 Aide

Lors de l'installation manuelle de mod\_gearman et Nagios, les fichiers/scripts présents dans /plugins ne se situent pas au même endroit que sur le serveur. Par conséquent, il est probable que rien ne fonctionne puisque le worker ne peut pas trouver les scripts.

Pour se faire, vous pouvez copier tous les dossiers et en le réécrit comme sur le serveur EON :

► `cp -r /usr/lib/nagios/plugins/* /srv/eyesofnetwork/nagios/plugins/`

Normalement, vous devriez plus rencontrer de problème ! 😊