

Document d'installation de Nginx



BACH Guillaume

Version 1.0

Table des matières

1.	Définition de NGINX	3
2.	Pourquoi NGINX et pas Apache ?	3
3.	Installation de NGINX	4
3.1.	Mise à jour	4
3.2.	Mise en place du logiciel	6

1. Définition de NGINX

Définition selon Wikipédia

NGINX est un logiciel libre de serveur Web (ou http) écrit par Igor Sysoev, dont le développement a débuté en 2002 pour les besoins d'un site russe à très fort trafic (Rambler qui est le moteur de recherche en Russie). Depuis avril 2019, c'est le serveur Web le plus utilisé au monde.

Aujourd'hui, il est plus qu'un serveur web ordinaire.

Ce que vous devez savoir sur NGINX :

- Au début, NGINX était utilisé comme un complément d'Apache
- Il a été usuellement sollicité pour traiter des fichiers statiques, mais actuellement, il a évolué et est devenu un serveur web qui traite les spectres complets des tâches du serveur.
- Désormais, NGINX est devenu comme un proxy inversé, un équilibreur de charges, un proxy de messagerie et un outil de mise en cache http.

2. Pourquoi NGINX et pas Apache ?

La principale différence entre Apache et NGINX se caractérise par leur design d'architecture. Apache utilise une approche axée sur les processus et crée un nouveau fil ou thread pour chaque requête.

Tandis que NGINX utilise une approche plutôt axée sur les événements pour traiter plusieurs requêtes pour un seul fil ou thread. (Fil ou Thread : composant du processus).

Nginx est plus efficace et moins exigeant pour les ressources système qu'Apache. Sur des contenus statiques, Nginx est 2,5 fois plus rapide, il peut traiter jusqu'à 1000 connexions simultanément.

Le comparatif d'Apache VS NGINX est encore un sujet de débat poignant même si les deux serveurs offrent une sécurité optimale grâce à leurs bases de code C.

Le code base de NGINX est toutefois significativement plus petit, donc celui-ci est plus avantageux en termes de sécurité. NGINX a aussi une liste de balises sécuritaires.

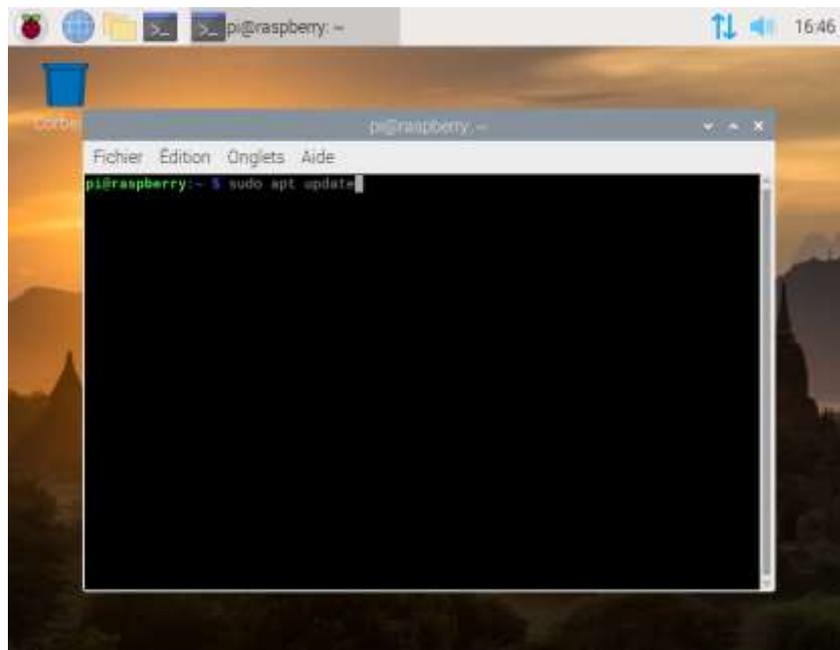
Le seul point négatif à NGINX est le fait de ne pas permettre de configuration additionnelle car pas de fichier « .htaccess », mais cela augmente quand même les performances.

3. Installation de NGINX

3.1. Mise à jour

Dans un premier temps, j'ouvre le terminal qui me permettra d'insérer les commandes et accéder aux applications.

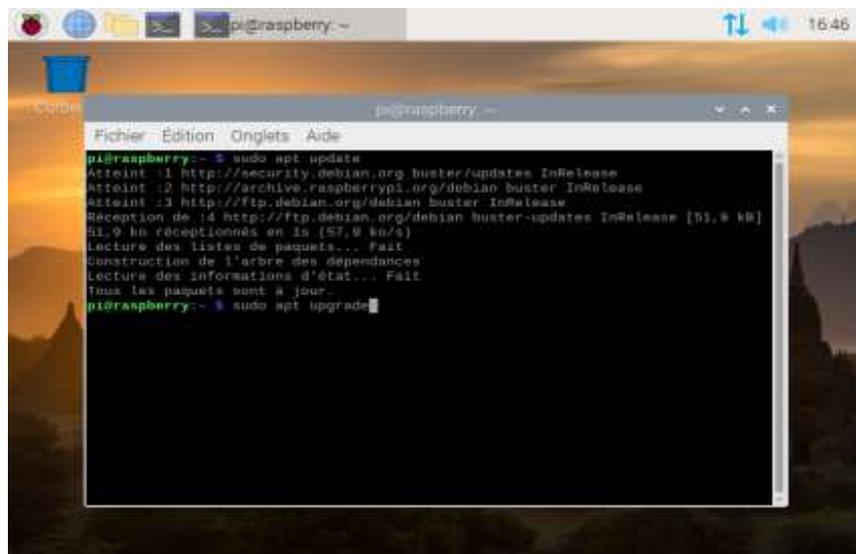
Ensuite, je commence par la commande « `sudo apt update` », où « `sudo` » est la commande qui permettra de lancer des applications avec les privilèges d'administration, « `apt` » est la commande pour la gestion des paquets « `Advanced Packaging Tools` » et « `update` » qui va permettre de rechercher les mises à jour des différents paquets.



La réception des paquets se fait et se met à jour.

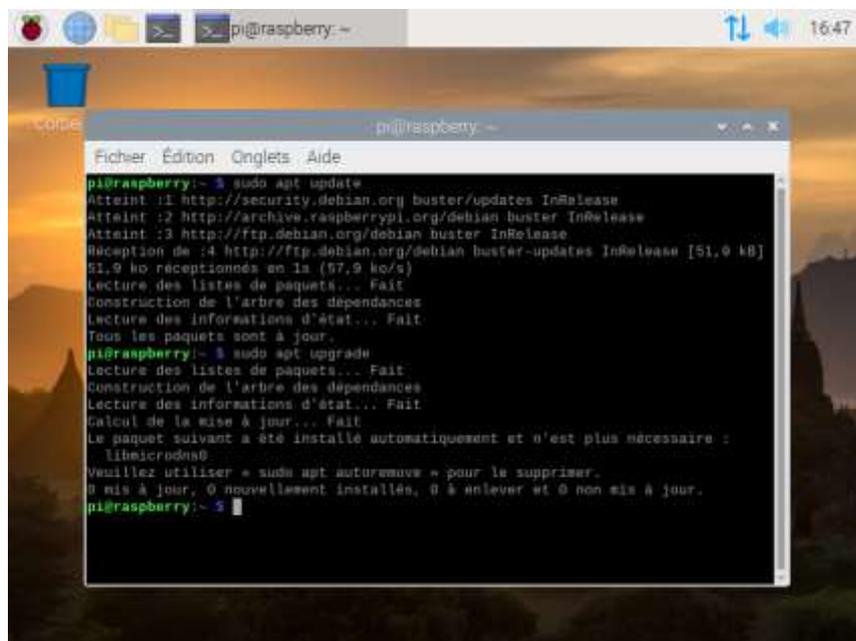


Je continue avec la commande « `sudo apt upgrade` » qui va installer de nouveaux paquets pour satisfaire des dépendances si nécessaires.



```
pi@raspberrypi:~$ sudo apt update
Atteint :1 http://security.debian.org/buster/updates InRelease
Atteint :2 http://archive.raspberrypi.org/debian buster InRelease
Atteint :3 http://ftp.debian.org/debian buster InRelease
Reception de :4 http://ftp.debian.org/debian buster-updates InRelease [51,9 kB]
51,9 ko réceptionnés en 1s (57,9 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
pi@raspberrypi:~$ sudo apt upgrade
```

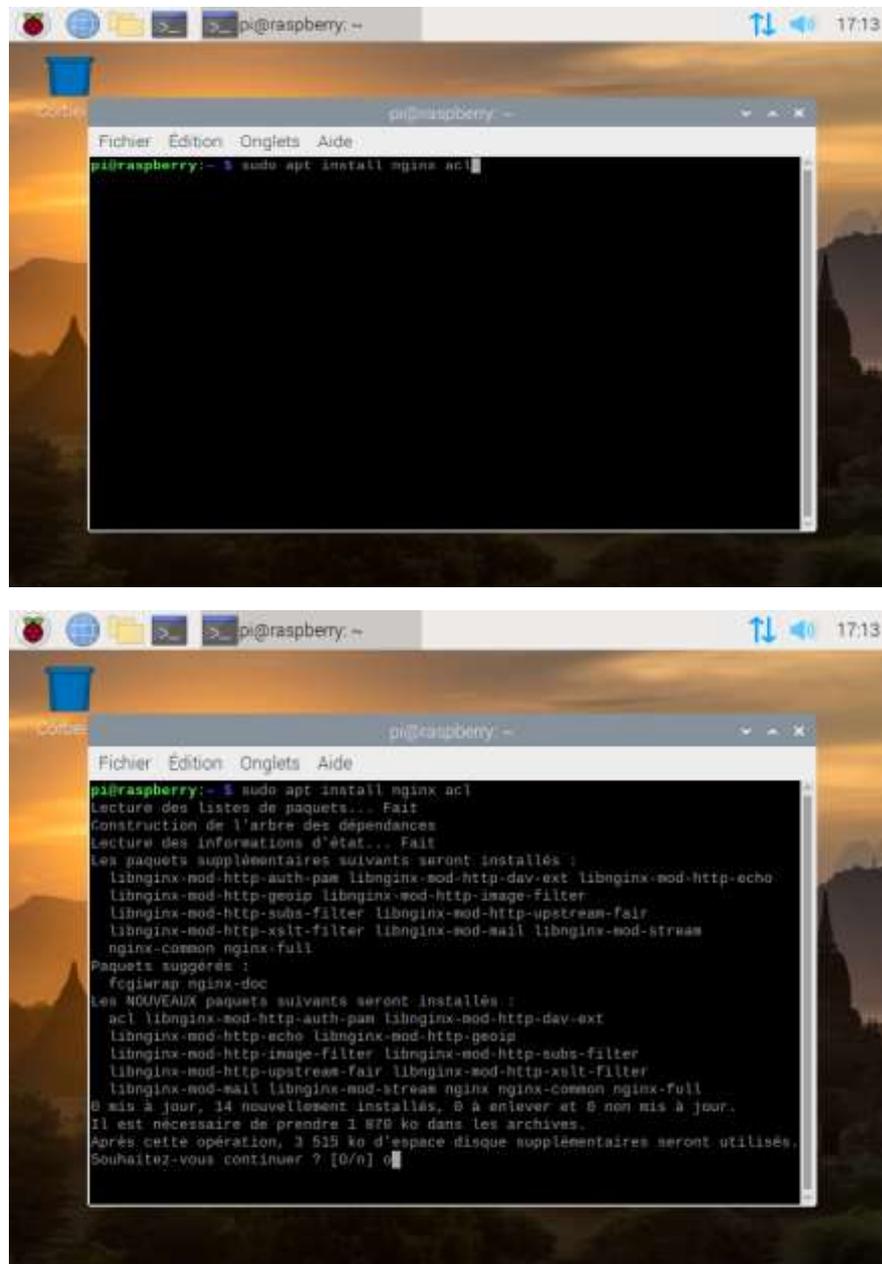
Les paquets sont à jour.



```
pi@raspberrypi:~$ sudo apt upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
Le paquet suivant a été installé automatiquement et n'est plus nécessaire :
libmicrodnss
Veuillez utiliser « sudo apt autoremove » pour le supprimer.
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
pi@raspberrypi:~$
```

3.2. Mise en place du logiciel

Je tape la commande « `sudo apt install nginx acl` ». Je vais donc installer l'appli Nginx avec une liste de contrôle d'accès afin de faire une gestion des droits d'accès aux fichiers.



Les paquets étant maintenant installés, nous allons voir via le navigateur si Nginx fonctionne bien en tapant dans celui-ci notre adresse IP.

Pour rappel, j'avais configuré l'adresse IP du Raspberry avec ma Freebox et obtenu l'adresse : 192.168.1.30.



Parfait notre serveur web fonctionne !

L'installation de NGINX s'est terminée avec succès, la suite consiste à installer PHP et Mariadb afin de pouvoir commencer notre serveur Web et implanter notre site internet.