



# **PPE : Installation NextCloud.**

# Sommaire

Détails de la solution et de ses dépendances :.....	3
NextCloud .....	3
Apache.....	3
UFW.....	4
PHP.....	4
MariaDB.....	4
Pré-requis logiciel et matériel :.....	5
Installation de la solution NextCloud :.....	6
1. Apache.....	6
2. UFW.....	7
3. PHP.....	8
4. MariaDB.....	10
5. Configuration Apache.....	12

# Détails de la solution et de ses dépendances :

## NextCloud

**Nextcloud** est une solution de cloud privé open source qui **permet le stockage, la synchronisation et le partage de fichiers de manière sécurisée**. Contrairement aux services cloud publics, Nextcloud offre un contrôle total sur les données, car il s'installe sur un serveur que l'on gère soi-même. Les utilisateurs peuvent y accéder depuis un navigateur, un ordinateur ou un appareil mobile, et retrouver leurs fichiers synchronisés en permanence.

Au-delà du simple partage de documents, Nextcloud intègre des fonctionnalités de collaboration comme l'édition en ligne de fichiers, la messagerie instantanée, la visioconférence, la gestion de calendriers et de contacts. Il peut être connecté à un annuaire Active Directory pour gérer les utilisateurs, ce qui le rend adapté aux environnements professionnels.

Nextcloud repose sur des technologies web classiques comme Linux, Apache ou Nginx, PHP et une base de données comme MariaDB. Il est également compatible avec les certificats SSL pour garantir la sécurité des échanges. Grâce à sa modularité, il peut évoluer selon les besoins, grâce à un large catalogue d'applications.

C'est une alternative solide aux services comme Google Drive ou Dropbox, particulièrement appréciée pour sa transparence, sa sécurité et le respect de la confidentialité.

## Apache

**Apache HTTP Server**, souvent simplement appelé **Apache**, est un serveur web open source très répandu, utilisé pour héberger des sites et des applications web. Il fonctionne sur la majorité des systèmes d'exploitation, notamment Linux, Windows et macOS. Apache repose sur une architecture modulaire, ce qui permet d'ajouter ou de désactiver des fonctionnalités selon les besoins, comme le support de PHP, SSL, la réécriture d'URL, l'authentification, ou encore les journaux personnalisés.

Ce serveur est très apprécié pour sa stabilité, sa flexibilité et la richesse de sa documentation. Il permet de gérer des sites statiques ou dynamiques, de configurer des hôtes virtuels pour héberger plusieurs sites sur un même serveur, et d'appliquer des règles d'accès ou de redirection via des fichiers

# UFW

**UFW** permet de définir rapidement quelles connexions sont autorisées ou bloquées, en se basant sur les ports, adresses IP ou protocoles. Il est particulièrement utile pour sécuriser un serveur en limitant l'accès aux services essentiels comme **SSH**, **HTTP** ou **HTTPS**, tout en bloquant le reste du trafic non sollicité. Il peut aussi être utilisé avec des profils d'applications (comme **Apache** ou **OpenSSH**) pour simplifier les règles.

# PHP

**PHP** (Hypertext Preprocessor) est un langage de script open source, principalement utilisé côté serveur pour le développement web. Il permet de générer dynamiquement du contenu **HTML**, de gérer des sessions utilisateurs, d'interagir avec des bases de données, et de traiter des formulaires. **PHP** est largement utilisé pour faire fonctionner des systèmes de gestion de contenu (**CMS**) comme **WordPress**, ainsi que des applications comme **Nextcloud**.

PHP s'intègre facilement avec des serveurs web comme **Apache** via des modules. I

l est conçu pour être rapide, flexible et relativement simple à apprendre.

Sa grande communauté, son support natif pour de nombreuses bases de données, et sa compatibilité avec des centaines d'extensions en font un outil très répandu pour les applications web.

# MariaDB

**MariaDB** est un système de gestion de base de données relationnelle open source, dérivé de MySQL. Il a été créé pour garantir la continuité du projet MySQL en restant libre et communautaire. Totalement compatible avec MySQL au niveau des commandes, des structures et des bibliothèques, MariaDB est souvent utilisé comme moteur de base de données pour des applications web comme WordPress, Nextcloud

# Pré-requis logiciel et matériel :

## Une machine Debian 12 déployée (se référer au PPE précédent)

Voici les pré-requis logiciel afin d'installer la solution **NextCloud** :

- OS : Linux (ici Debian v.12)
- Serveur web : Apache 2
- PHP : Version 8.2 avec les extensions suivantes ( )
- Base de donnée : MariaDB
- Client web

Voici maintenant les pré-requis matériel afin d'installer cette dernière :

- CPU : 1.4 GHz 32x ou x64
- RAM : 2 Go à 4 Go
- Stockage : 20 Go (favoriser du SSD pour un transfert optimal)

# Installation de la solution NextCloud :

Pour des raisons de praticité, j'élève mes privilèges en **root**.

Je mets ensuite à jour les sources des dépôts et lance les mises à jour si nécessaire avec les commandes suivantes ;

```
su -                               elliot@deb12speed:~$ su -
apt update                          root@deb12speed:~# apt update
apt upgrade                          root@deb12speed:~# apt upgrade
```

```
elliot@deb12speed:~$ su -
Mot de passe :
root@deb12speed:~# apt update
Atteint :1 http://deb.debian.org/debian bookworm InRelease
Atteint :2 http://security.debian.org/debian-security bookworm-security InRelease
Atteint :3 http://deb.debian.org/debian bookworm-updates InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
root@deb12speed:~# apt upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
root@deb12speed:~#
```

## 1. Apache

Premièrement, nous allons procéder à l'installation de notre serveur apache avec la commande suivante ;

```
apt-get install apache2
```

```
root@deb12speed:~# apt-get install apache2
```

Puis grâce à la commande ;

```
systemctl status apache2
```

Vérifiez si le service est bien en cour d'exécution. Voici ce que la commande devrait vous retourner ;

```
root@deb12speed:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Sun 2025-04-13 16:31:42 CEST; 32s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 3969 (apache2)
    Tasks: 55 (limit: 2241)
   Memory: 11.1M
      CPU: 64ms
   CGroup: /system.slice/apache2.service
           └─3969 /usr/sbin/apache2 -k start
             └─3970 /usr/sbin/apache2 -k start
               └─3971 /usr/sbin/apache2 -k start
```

Notre serveur apache est correctement installé, passons à l'installation d'un pare-feu **UFW**

## 2. UFW

Installez le paquet ufw ;

```
apt install ufw
```

```
root@deb12speed:~# apt install ufw
```

une fois effectué, vous devrez installer OpenSSH ;

```
apt install --reinstall openssh-server
```

```
root@deb12speed:~# apt install --reinstall openssh-server
```

Puis autoriser **OpenSSH** sur le pare-feu **UFW** :

```
root@deb12speed:~# ufw allow OpenSSH
Rules updated
Rules updated (v6)
```

Vous pouvez maintenant activer le pare-feu avec la commande ci-dessous, celle ci devrait vous retourner le même message que sur la capture d'écran ;

```
root@deb12speed:~# ufw enable
Firewall is active and enabled on system startup
```

Nous allons aussi autoriser WWW Full afin d'ajouter le port HTTP et HTTPs pour notre serveur web ;

```
root@deb12speed:~# ufw allow "WWW Full"
Rule added
Rule added (v6)
root@deb12speed:~#
```

La configuration UFW est désormais achevée, nous passons maintenant à l'installation de PHP ainsi que de ses dépendances nécessaires.

### 3. PHP

PHP est installé nativement sur notre système d'exploitation **Debian**. Nous aurons alors uniquement besoin d'installer les dépendances nécessaires avec la commande suivante ;

```
apt install -y php php-curl php-cli php-mysql php-gd php-common php-xml php-json php-intl php-pear php-imagick php-dev php-common php-mbstring php-zip php-soap php-bz2 php-bcmath php-gmp php-apcu libmagickcore-dev
```

```
root@12deb:~# sudo apt install -y php php-curl php-cli php-mysql php-gd php-common php-xml php-json php-intl php-pear php-imagick php-dev php-common php-mbstring php-zip php-soap php-bz2 php-bcmath php-gmp php-apcu libmagickcore-dev
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
```

Désormais nous devons ouvrir le fichier de configuration de php afin d'apporter des modifications sur quelques lignes. Le fichier en question se trouve dans ; /etc/php/8.2/apache2/php.ini

Dans notre cas nous l'éditerons avec **nano** et modifierons les lignes suivantes avec les correspondances suivantes ;

979

```
; https://php.net/date.timezone
date.timezone = Europe/amsterdam
```

435

```
memory_limit = 512M
```

855

```
upload_max_filesize = 500M
```

703

```
max_execution_time = 300
```

409

```
post_max_size = 600M
```

966

```
zend_extension=opcache
```

Ajoutez les paramètres recommandés par **NextCloud pour Debian** ;

```
zend_extension=opcache

; Module Settings ;

opcache.enable = 1
opcache.interned_strings_buffer = 8
opcache.max_accelerated_files = 10000
opcache.memory_consumption = 128
opcache.save_comments = 1
opcache.revalidate_freq = 1
```

Vous pouvez désormais relancer le service apache pour qu'il prenne les modifications en compte ;

```
root@12deb:~# systemctl restart apache2
root@12deb:~#
```

**Nous allons, pour suivre, procéder à l'installation de MariaDB afin de créer notre base de données**

## 4. MariaDB

Saisissez simplement la commande suivante :

```
apt install mariadb-server
```

Suite à quoi vous pourrez contrôler la mise en service de mariadb avec la commande suivante ;

```
systemctl status mariadb
```

Elle devrait vous retourner ceci ;

```
root@12deb:~# systemctl is-enabled mariadb
enabled
root@12deb:~# systemctl status mariadb
● mariadb.service - MariaDB 10.11.11 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; preset: enal
   Active: active (running) since Sat 2025-04-12 18:51:36 CEST; 25min ago
     Docs: man:mariadbd(8)
           https://mariadb.com/kb/en/library/systemd/
   Main PID: 37678 (mariadbd)
   Status: "Taking your SQL requests now..."
```

Nous allons alors créer un mot de passe root pour maria db ainsi qu'effacer la base test et les utilisateurs anonymes existants ;

```
root@12deb:~# mariadb-secure-installation
```

nous répondrons non à cette question uniquement,

```
Switch to unix_socket authentication [Y/n] N
```

vous pourrez répondre **Y** aux suivantes ;

```
Change the root password? [Y/n] Y
```

```
Remove anonymous users? [Y/n] Y
```

```
Disallow root login remotely? [Y/n] Y
```

```
Remove test database and access to it? [Y/n] Y
```

```
Reload privilege tables now? [Y/n] Y
```

L'installation est terminée, **MariaDB** nous retourne ceci ;

```
All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.
```

```
Thanks for using MariaDB!
root@12deb:~#
```

Nous pouvons maintenant créer notre base de données avec la suite de commande suivante ;

```
Thanks for using MariaDB!
root@deb12speed:~# mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 39
Server version: 10.11.11-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE elliot_db;
Query OK, 1 row affected (0,000 sec)

MariaDB [(none)]> CREATE USER 'speed'@'localhost' IDENTIFIED BY 'votre mot de
passé';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON elliot_db.* TO 'speed'@'localhost';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]>
```

Avant de passer à l'étape suivante, nous allons télécharger une version récente du code source de **NextCloud** afin d'assurer la compatibilité de l'OS.

```
root@deb12speed:~# apt install curl unzip -y
root@deb12speed:/var/www# curl -o nextcloud.zip https://download.nextcloud.com/s
erver/releases/latest.zip
```

Une fois téléchargé, vous pouvez extraire le code source et ainsi donner les droits correspondants ;

```
root@deb12speed:/var/www# unzip nextcloud.zip
root@deb12speed:/var/www# chown -R www-data:www-data nextcloud
```

## 5. Configuration Apache

Nous allons maintenant créer un fichier de configuration pour NextCloud.  
J'utilise toujours la commande nano pour ouvrir mon éditeur de texte ;

```
root@12deb:/var/www# nano /etc/apache2/sites-available/nextcloud.conf
```

et voici la configuration que je rentre ; ( attention à bien remplacer mon ip par la votre ! )

```
GNU nano 7.2 /etc/apache2/sites-available/nextcloud.conf
<VirtualHost *:80>
  ServerName nextcloud.192.168.5.140
  DocumentRoot /var/www/nextcloud/
                /var/www/nextcloud/

# log files
  ErrorLog /var/log/apache2/files.192.168.5.140-error.log
  CustomLog /var/log/apache2/files.192.168.5.140-access.log
  combined

  <Directory /var/www/nextcloud/>
    Options +FollowSymlinks
    AllowOverride All

    <IfModule mod_dav.c>
      Dav off
    </IfModule>

    SetEnv HOME /var/www/nextcloud
    SetEnv HTTP_HOME /var/www/nextcloud
  </Directory>
</VirtualHost>

[ Lecture de 25 lignes ]
```

Vous pouvez alors activer la configuration et essayer la configuration test avec les commandes suivantes ;

```
a2ensite nextcloud.conf
```

```
apachectl configtest
```

Ce qui vous retourne alors ;

```
| Syntax OK
```

Vous êtes désormais prêts à vous lancer dans la configuration de celui-ci par l'interface web de votre navigateur en saisissant l'ip associé.