

# Installer Nextcloud & Onlyoffice

Mise à jour : 01-26

Par Rémi Albertucci





# Table des matières

<b>I Intro.....</b>	<b>2</b>
<b>II Creation VM.....</b>	<b>3</b>
<b>III NPM.....</b>	<b>5</b>
<b>IV Installation de Nextcloud.....</b>	<b>7</b>
IV.1 Redis.....	9
<b>V Sécurisation.....</b>	<b>10</b>
V.1 Cron.....	11
<b>VI Only Office.....</b>	<b>12</b>
VI.1 Docker.....	12
VI.2 Configuration dans Nextcloud.....	13
<b>VII Utiles.....</b>	<b>16</b>
Chemins :	16
Logs :	16
Commandes :	16

# I Intro

Nextcloud est une plateforme de collaboration et de stockage de fichiers auto-hébergée.

Elle permet de remplacer des services propriétaires de type Google cloud Onedrive en gardant la souveraineté de ses données.

On y retrouve notamment le partage de fichiers, la synchronisation multi-appareils, et l'édition collaborative de documents.

Dans ce tutoriel, nous allons mettre en place une instance Nextcloud complète, auto-hébergée, sécurisée et utilisable par plusieurs utilisateurs.

L'installation repose sur une machine virtuelle Debian 13 (Trixie), un serveur web Apache, une base de données MariaDB, Redis pour les performances, et un accès sécurisé via un reverse proxy NGINX Proxy Manager. Nous ajouterons également OnlyOffice afin de permettre l'édition de documents en ligne.

Le guide suit une approche progressive et pragmatique :

- création et sécurisation de la machine virtuelle
- installation des dépendances et de Nextcloud
- configuration du reverse proxy et du HTTPS
- durcissement de la configuration
- ajout des services complémentaires (Redis, cron, OnlyOffice)

Tout au long de ce document, nous retrouverons le codes visuels suivants :

- à adapter à la config souhaitée
- commandes à rentrer
- fichiers de configuration (nano)
- sera-rempli **NE PAS MODIFIER** => attention à ne pas faire de copié coller sur cette partie

## II Creation VM

Nous créons une machine virtuelle tournant sous Debian 13 (trixie).

Nous lui attribuons 2 coeurs, ce qui sera suffisant, et 8192Mo (8Go) de RAM en prévision des multiples utilisateurs.

Nous suivons le processus d'installation classique et une fois dans la vm, nous commençons la configuration en se connectant ssh à la machine :

- création d'un utilisateur et ajout au groupe sudo

```
apt update && apt upgrade -y
apt install sudo -y
adduser <NOM>
usermod -aG sudo <NOM>
su <NOM>
```

- désactivation de la connexion à root en ssh :

```
sudo nano /etc/ssh/sshd_config
```

- dans le fichier de config on modifie :

```
PermitRootLogin no
```

```
sudo systemctl restart ssh
```

- installation et configuration de fail2ban

```
sudo apt install fail2ban -y
```

- Copier la conf par défaut

```
sudo cp /etc/fail2ban/jail.conf
/etc/fail2ban/jail.local
```

- Éditer jail.local

```
sudo nano /etc/fail2ban/jail.local
```

```
[sshd]
enabled = true
port = ssh
maxretry = 5
bantime = 3600

[nextcloud]
enabled = true
filter = nextcloud
logpath = /var/www/html/nextcloud/data/nextcloud.log
```

```
maxretry = 5  
bantime = 3600
```

```
sudo nano /etc/fail2ban/filter.d/nextcloud.conf
```

[Definition]

```
failregex = ^.*Login failed.*Remote IP.*<HOST>.*
```

- Redémarrer fail2ban

```
sudo systemctl restart fail2ban  
sudo systemctl enable fail2ban
```

```
sudo fail2ban-client reload  
sudo fail2ban-client status
```

doit nous renvoyer :

```
furet@nextcloud:/var/www/html/nextcloud/config$ sudo fail2ban-client reload  
sudo fail2ban-client status  
OK  
status  
- Number of jail:      2  
- Jail list:  nextcloud, sshd
```

## III NPM

On suppose NGINX proxy manager déjà installé et configuré.

Dans NPM, on se dirige dans Proxy Hosts et modifie l'entrée <https://NDD-DE-LA-VM>.

On remplit comme suit :

- Onglet Details :
  - Scheme : `http`
  - Forward Hostname/IP : `IP-DE-LA-VM`
  - Forward Port : 80
  - Block Common Exploits :  décoché pour le webdav
  - Websockets Support :  coché
- Onglet SSL :
  - Certificat Let's Encrypt
  - Force SSL :
  - HTTP/2 :
  - HSTS :

Exemple de configuration :

### Edit Proxy Host

Details Custom Locations SSL

Domain Names

nextcloud.remifuret.fr x

Scheme	Forward Hostname / IP	Forward Port
http	192.168.11.30	80

Access List

Publicly Accessible

**Options**

Cache Assets

Block Common Exploits

Websockets Support

Cancel Save

### Edit Proxy Host

Details Custom Locations SSL

SSL Certificate

nextcloud.remifuret.fr

Force SSL  HTTP/2 Support

HSTS Enabled  HSTS Sub-domains

Cancel Save



## IV Installation de Nextcloud

Maintenant que tout est à jour, que l'on a un utilisateur, et un nom de domaine avec certificat SSL, on peut s'attaquer à l'installation.

On commence par les paquets essentiels, apache et notre db :

```
apt install curl wget unzip apache2 mariadb-server  
-y
```

On démarre apache et mariadb :

```
sudo systemctl start apache2 && sudo systemctl  
enable apache2  
sudo systemctl start mariadb && sudo systemctl  
enable mariadb
```

On contrôle ensuite si tout est lancé :

```
sudo systemctl status apache2  
sudo systemctl status mariadb
```

On peut ensuite installer toutes les dépendances dont notre serveur LAMP aura besoin :

```
sudo apt install php libapache2-mod-php php-cgi  
php-mysqli php-pear php-phpseclib php-mysql php-  
mbstring php-zip php-gd php-curl php-common php-  
bcmath php-intl php-imagick php-bz2 php-gmp php-  
apcu php-redis -y
```

On vérifie la version de php installée, la latest étant au moment de la rédaction est 8.4 :

```
php -v
```

On rentre dans l'invité de commandes de mariadb :

```
sudo mariadb
```

On entre ensuite ces commandes en prenant soin de noter dans son gestionnaire de mot de passe les identifiants :

```
CREATE USER 'dbuser'@'localhost' IDENTIFIED BY
'Mot_De_Passe_Fort_Ici';
CREATE DATABASE nextdb;
GRANT ALL PRIVILEGES ON nextdb.* TO
'dbuser'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

On télécharge ensuite la dernière version de NextCloud Server dans le dossier /var/www/html, et on l'installe :

```
cd /var/www/html
wget
https://download.nextcloud.com/server/releases/lat
est.zip
unzip latest.zip
chown -R www-data:www-data nextcloud/
find nextcloud/ -type f -exec chmod 644 {} \;
find nextcloud/ -type d -exec chmod 755 {} \;
rm latest.zip
```

On modifie le fichier de configuration d'apache :

```
sudo nano
/etc/apache2/sites-available/nextcloud.conf
```

```
<VirtualHost *:80>
ServerName https://NDD-DE-LA-VM
DocumentRoot /var/www/html/nextcloud

<Directory /var/www/html/nextcloud>
AllowOverride All
Require all granted
</Directory>

<IfModule mod_headers.c>
# Header always set Strict-Transport-Security "max-
age=15552000; includeSubDomains"
Header always set X-Content-Type-Options "nosniff"
Header always set X-Frame-Options "SAMEORIGIN"
Header always set X-XSS-Protection "1; mode=block"
Header always set Referrer-Policy "no-referrer"
</IfModule>

ErrorLog ${APACHE_LOG_DIR}/nextcloud_error.log
```

```
CustomLog ${APACHE_LOG_DIR}/nextcloud_access.log
combined

</VirtualHost>
```

On redémarre ensuite les modules Apache :

```
sudo a2enmod rewrite headers
```

On active le site et désactive le default :

```
sudo a2ensite nextcloud.conf
sudo a2dissite 000-default.conf
```

Tester et redémarrer Apache :

```
apachectl -t
sudo systemctl restart apache2
```

On peut maintenant se rendre sur <http://IP-DE-LA-VM> que l'on a récupéré avec la commande `ip a` ou directement sur <https://NDD-DE-LA-VM>. On remplit :

- **Nom d'utilisateur admin** et **mot de passe**
- **Base de données** : MySQL/MariaDB
  - Utilisateur : `dbuser`
  - Mot de passe : `<MDP-DB>`
  - Nom de la base : `nextdb`
  - Hôte : `localhost`

On clique sur installer et on revient dans le shell de notre VM.

## IV.1 Redis

Redis sera utilisé pour la gestion du cache et des verrous applicatifs, afin d'améliorer les performances générales de Nextcloud et d'éviter certains problèmes de concurrence, notamment lors de l'accès simultané aux fichiers.

On l'installe comme suit :

```
sudo apt install redis-server php-redis -y
sudo systemctl enable redis-server
sudo systemctl start redis-server
```

## V Sécurité

```
sudo nano
/var/www/html/nextcloud/config/config.php
```

```
<?php
$CONFIG = array (
  'instanceid' => 'sera-rempli NE PAS MODIFIER',
  'passwordsalt' => 'sera-rempli NE PAS MODIFIER',
  'secret' => 'sera-rempli NE PAS MODIFIER',
  'version' => 'sera-rempli NE PAS MODIFIER',

  // Domaines et proxy
  'trusted_domains' =>
  array (
    0 => 'nextcloud.remifuret.fr',
  ),
  'trusted_proxies' =>
  array (
    0 => '192.168.11.252', // IP du NPM
  ),
  'forwarded_for_headers' =>
  array (
    0 => 'HTTP_X_FORWARDED_FOR',
  ),

  // HTTPS / reverse proxy
  'overwriteprotocol' => 'https',
  'overwritehost' => 'nextcloud.remifuret.fr',
  'overwritecli.url' => 'https://nextcloud.remifuret.fr',

  // Données et DB
  'datadirectory' => '/var/www/html/nextcloud/data',
  'dbtype' => 'mysql',
  'dbname' => 'nextdb',
  'dbhost' => 'localhost',
  'dbtableprefix' => 'oc_',
  'dbuser' => 'NOM-USER',
  'dbpassword' => 'MDP-USER',
  'mysql.utf8mb4' => true,

  'installed' => true,
```

```
// Cache
'memcache.local' => '\\OC\\Memcache\\APCu',
'memcache.locking' => '\\OC\\Memcache\\Redis',
'redis' =>
array (
  'host' => 'localhost',
  'port' => 6379,
),

'maintenance_window_start' => 3,
'maintenance' => false,
);
```

On configure maintenant php :

```
sudo nano /etc/php/8.4/apache2/php.ini
```

```
memory_limit = 512M
upload_max_filesize = 256M
post_max_size = 256M
max_execution_time = 30
date.timezone = Europe/Paris
opcache.enable = 1
opcache.memory_consumption = 128
opcache.interned_strings_buffer = 8
opcache.max_accelerated_files = 10000
opcache.revalidate_freq = 1
opcache.save_comments = 1
```

On redémarre bien évidemment le service apache :

```
sudo systemctl restart apache2
```

Enfin, on ajuste les permissions de config.php :

```
sudo chmod 640
/var/www/html/nextcloud/config/config.php
sudo chown www-data:www-data
/var/www/html/nextcloud/config/config.php
```

## V.1 Cron

```
sudo crontab -u www-data -e
```

Si c'est la première fois, choisis **\*\*1\*\*** (nano), puis ajoute à la fin :

```
*/5 * * * * /usr/bin/php8.4 -f
/var/www/html/nextcloud/cron.php
```

## VI Only Office

Pour installer docker sur debian 13, se référer à la documentation appropriée, présente sur ce site ou sur internet.

### VI.1 Docker

On crée un dossier /home/**user**/docker/onlyoffice et 4 sous dossiers, et on lance la création du fichier yaml :

```
mkdir /home/user/docker
cd /home/user/docker
mkdir onlyoffice
cd onlyoffice
mkdir data logs cache lib &&sudo nano docker-
compose.yaml
```

```
services:
  onlyoffice-documentserver:
    image: onlyoffice/documentserver:latest
    container_name: onlyoffice
    restart: always
  dns :
    - 1.1.1.1
    - 8.8.8.8
  ports:
    - "8082:80"
  environment:
    - JWT_ENABLED=true
    - JWT_SECRET=<MDP>
    - JWT_HEADER=Authorization
    # Limites de ressources (optionnel)
    - WOPI_ENABLED=false
    # Langue par défaut
    - DEFAULT_LANG=fr-FR
  volumes:
    - ./logs:/var/log/onlyoffice
    - ./data:/var/www/onlyoffice/Data
    - ./lib:/var/lib/onlyoffice
    - ./cache:/var/lib/postgresql
  networks:
    - onlyoffice-network
```

```
networks:  
onlyoffice-network:  
  driver: bridge
```

On lance ensuite la commande :

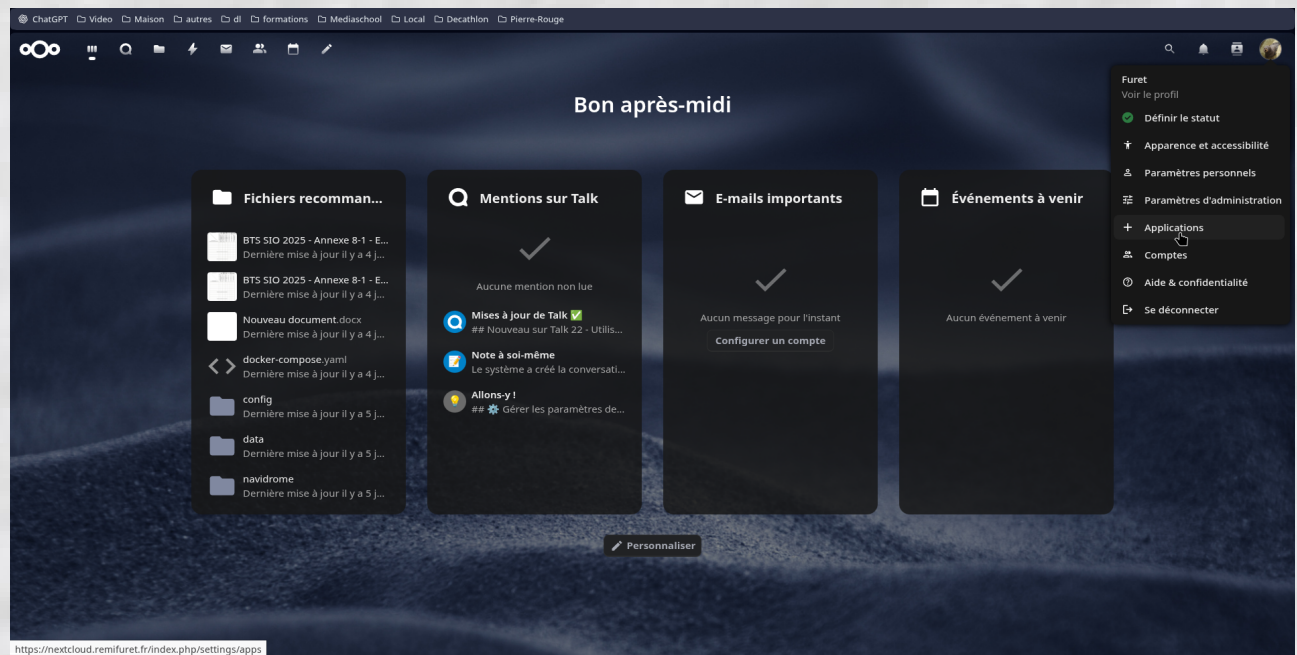
```
docker compose up -d
```

et on verifie qu'onlyoffice démarre bien avec la commande

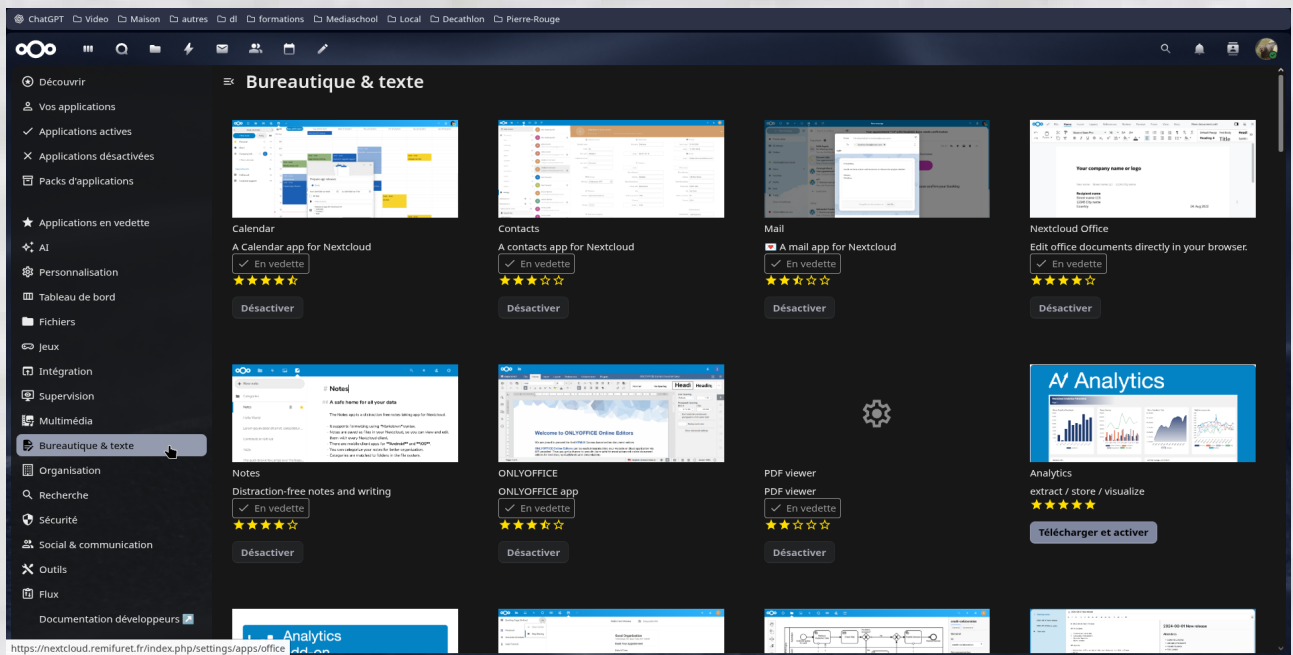
```
docker ps
```

## VI.2 Configuration dans Nextcloud

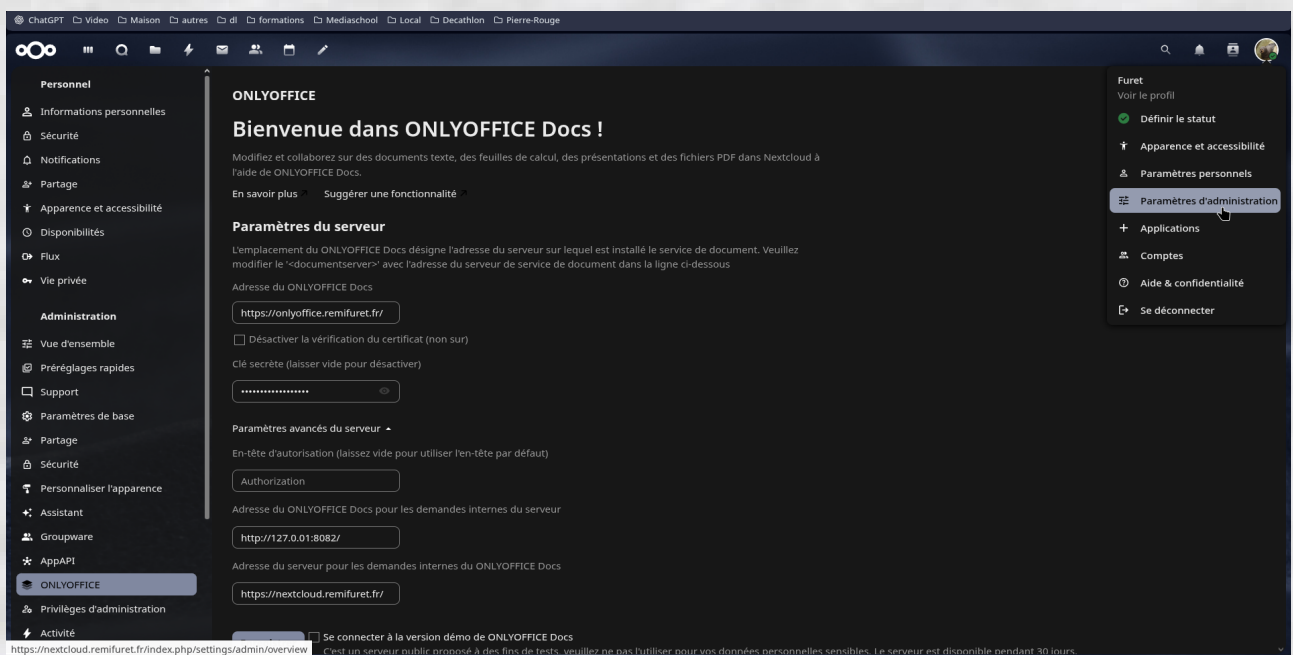
Dans NextCloud, on se rend sur notre compte en haut à droite et on va dans « Applications » :



Dans l'onglet des applications, on se dirige dans « bureautique et texte » et on y recherche only office. On clique alors sur « Télécharger et activer » :



Retour sur notre compte onglet « Paramètres d'Administration » puis dans le menu à gauche sur Only Office, ce qui nous amène sur cette page :

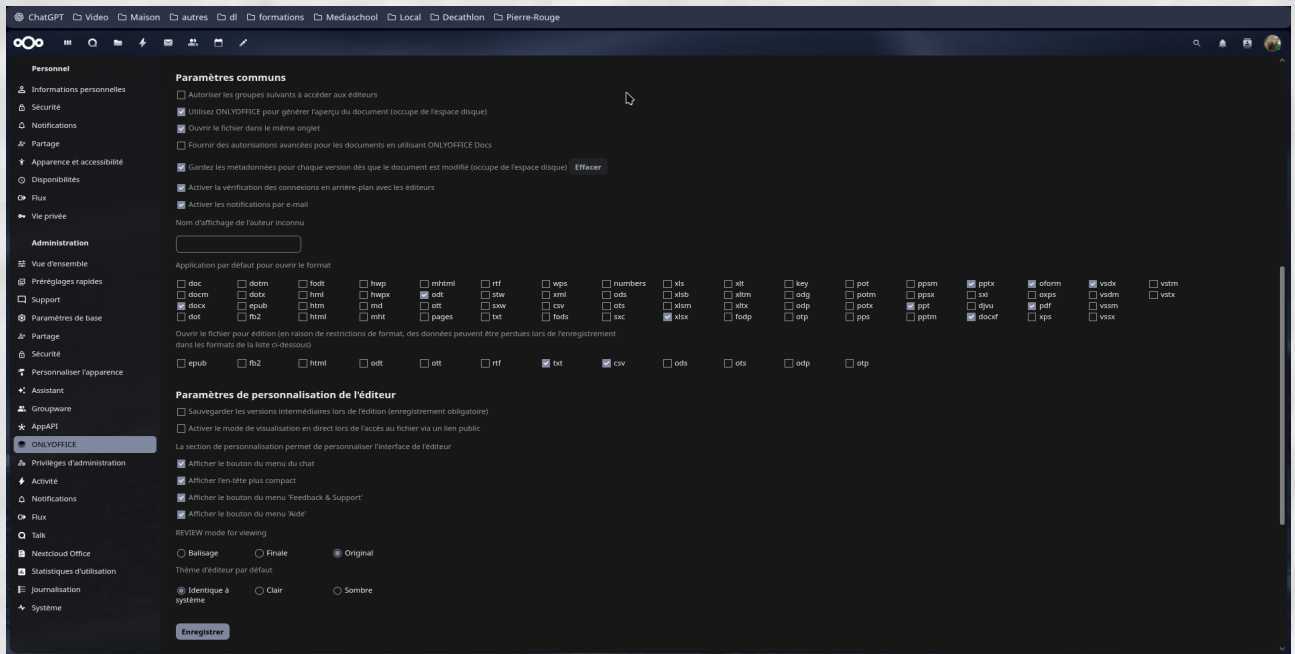


On remplit ensuite de cette façon :

- Adresse du ONLYOFFICE Docs : <https://IP-DE-LA-VM>
- Clé secrète : **définie dans le YAML**
- Adresse du ONLYOFFICE Docs pour les demandes internes du serveur : [127.0.0.1:8082](http://127.0.0.1:8082)
- Adresse du serveur pour les demandes internes du ONLYOFFICE Docs : <https://IP-DE-LA-VM>



Dans les paramètres communs, on peut définir quels types de documents only office ouvrira par défaut ainsi que des options utiles :



## VII Utiles

### Chemins :

- `/var/www/html/nextcloud/data/` : stockage des fichiers utilisateurs
- `/var/www/html/nextcloud/config/` : configuration de Nextcloud

### Logs :

- `/var/log/apache2/nextcloud_error.log` : erreurs Apache/Nextcloud
- `/var/log/apache2/nextcloud_access.log` : accès Apache
- `/var/www/html/nextcloud/data/nextcloud.log` : logs internes Nextcloud (fail2ban)

### Commandes :

- `sudo systemctl restart apache2` : redémarrer Apache
- `sudo systemctl restart mariadb` : redémarrer MariaDB
- `sudo systemctl restart redis-server` : redémarrer Redis
- `sudo fail2ban-client status` : vérifier l'état de fail2ban
- `php -v` : vérifier la version de PHP
- `sudo crontab -u www-data -l` : lister les tâches cron de Nextcloud