

# TP — Installation et Configuration de Nextcloud

BTS SIO — SISR | Ubuntu 22.04 | Nextcloud 30

Travaux Pratiques

## Installation et Configuration de Nextcloud

Nextcloud 30 · Apache · MariaDB · PHP 8.3 · Redis · Let's Encrypt

Étudiant	Touil Ayoub
Formation	BTS SIO – option SISR
Système cible	Ubuntu 22.04 LTS
Version Nextcloud	30.0.9.2
Date	Avril 2025

## Table des matières

1. Préparation du système
2. Installation des prérequis
3. Déploiement de Nextcloud
4. Configuration d'Apache
5. Configuration de la base de données MariaDB
6. Finalisation de l'installation
7. Activation des applications
8. Optimisation des performances
9. Certificat SSL avec Let's Encrypt
10. Création des utilisateurs et groupes
11. Sauvegarde de Nextcloud
12. Mise à jour de Nextcloud

## 1. Préparation du système

### 1.1 Configuration du fuseau horaire

Avant toute installation, il est essentiel de s'assurer que le fuseau horaire du serveur est correctement configuré. Un décalage horaire peut provoquer des erreurs de synchronisation, notamment avec les certificats SSL et les tâches planifiées (cron).

```
sudo timedatectl set-timezone Europe/Paris
sudo hwclock -w
timedatectl status
```

```
root@modele:/home/test# sudo timedatectl set-timezone Europe/Paris
root@modele:/home/test# sudo hwclock -w
root@modele:/home/test# timedatectl status
          Local time: mar. 2025-04-15 15:35:04 CEST
          Universal time: mar. 2025-04-15 13:35:04 UTC
             RTC time: mar. 2025-04-15 13:35:04
             Time zone: Europe/Paris (CEST, +0200)
System clock synchronized: yes
              NTP service: active
             RTC in local TZ: no
```

Figure 1 — Vérification du fuseau horaire Europe/Paris (NTP actif, horloge synchronisée)

### 1.2 Mise à jour du système

La mise à jour préalable de l'ensemble des paquets garantit la stabilité du système et évite les conflits de dépendances lors de l'installation.

```
sudo apt -y update && sudo apt -y upgrade
```

## 2. Installation des prérequis

### 2.1 Ajout du dépôt PHP 8.3

PHP 8.3 n'est pas inclus dans les dépôts officiels d'Ubuntu 22.04. Il est nécessaire d'ajouter le dépôt **ppa:ondrej/php**, maintenu par Ondřej Surý, afin d'obtenir cette version.

```
sudo apt install -y software-properties-common
sudo add-apt-repository ppa:ondrej/php
sudo apt update -y
```

### 2.2 Installation de la pile LAMP et des extensions PHP

Nextcloud 30 requiert Apache, MariaDB et un ensemble précis d'extensions PHP. La variable **DEBIAN\_FRONTEND=noninteractive** permet d'éviter les interruptions interactives lors de l'installation automatisée.

```
export DEBIAN_FRONTEND=noninteractive
sudo apt install -y apache2 mariadb-server libapache2-mod-php8.3 \
php8.3 php8.3-gd php8.3-mysql php8.3-curl php8.3-mbstring php8.3-intl \
php8.3-imagick php8.3-xml php8.3-zip php8.3-apcu php8.3-redis \
php8.3-ldap php8.3-smbclient php8.3-bcmath php8.3-gmp
```

### 2.3 Téléchargement et extraction de Nextcloud

L'archive officielle de Nextcloud 30 est téléchargée depuis le site officiel, puis extraite dans le répertoire web d'Apache.

```
wget https://download.nextcloud.com/server/releases/latest-30.tar.bz2
sudo tar -xvf latest-30.tar.bz2 -C /var/www/
```

## 3. Configuration d'Apache

### 3.1 Création du VirtualHost

Un fichier de configuration dédié est créé pour le site Nextcloud dans Apache. Il définit l'alias racine, les permissions d'accès, et active les en-têtes de sécurité HSTS (HTTP Strict Transport Security).

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

```
Alias / "/var/www/nextcloud/"
<Directory /var/www/nextcloud/>
    Require all granted
    AllowOverride All
    Options FollowSymLinks MultiViews
    <IfModule mod_dav.c>
        Dav off
    </IfModule>
    <IfModule mod_headers.c>
        Header always set Strict-Transport-Security "max-age=15552000; includeSubDomains"
    </IfModule>
</Directory>
```

Figure 2 — Contenu du VirtualHost nextcloud.conf avec directives de sécurité HSTS

### 3.2 Activation du site et des modules Apache

Les modules nécessaires (réécriture d'URL, en-têtes, SSL, etc.) sont activés, puis Apache est redémarré pour prendre en compte la configuration.

```
sudo a2ensite nextcloud.conf
sudo a2enmod rewrite headers env dir mime ssl
sudo a2ensite default-ssl
sudo systemctl restart apache2
```

### 3.3 Attribution des droits au répertoire Nextcloud

Apache s'exécute sous l'utilisateur **www-data**. Il est impératif de lui attribuer la propriété récursive du répertoire Nextcloud pour que le serveur web puisse lire et écrire les fichiers.

```
sudo chown -R www-data:www-data /var/www/nextcloud/
```

### 3.4 Vérification de la version installée

La commande occ (outil CLI de Nextcloud) permet de vérifier les informations de version avant de procéder à la configuration.

```
sudo -u www-data php /var/www/nextcloud/occ status
```

```
Nextcloud is not installed - only a limited number of commands are available
Nextcloud 30.0.9
root@modele:/home/test# sudo -u www-data php /var/www/nextcloud/occ status
Nextcloud is not installed - only a limited number of commands are available
- installed: false
- version: 30.0.9.2
- versionstring: 30.0.9
- edition:
- maintenance: false
- needsDbUpgrade: false
- productname: Nextcloud
- extendedSupport: false
root@modele:/home/test#
```

Figure 3 — Statut Nextcloud 30.0.9.2 (non encore installé à ce stade)

## 4. Configuration de la base de données MariaDB

### 4.1 Sécurisation initiale de MariaDB

Un script SQL héritaire permet de sécuriser l'installation MariaDB : définition du mot de passe root, suppression des utilisateurs anonymes et de la base de test.

```
sudo mysql --user=root <<-EOF
SET PASSWORD FOR 'root'@'localhost' = PASSWORD('VotreMotDePasse');
DELETE FROM mysql.user WHERE User='root'
AND Host NOT IN ('localhost','127.0.0.1','::1');
DELETE FROM mysql.user WHERE User='';
DELETE FROM mysql.db WHERE Db='test' OR Db='test_%';
FLUSH PRIVILEGES;
EOF
```

■ ■ Remplacer 'VotreMotDePasse' par un mot de passe fort avant toute mise en production.

### 4.2 Installation via la CLI occ

Plutôt que d'utiliser l'assistant web (qui peut générer une erreur 500 en cas de mauvaise configuration PHP), l'installation est effectuée directement via la commande **occ maintenance:install**.

```
sudo -u www-data php /var/www/nextcloud/occ maintenance:install \
--admin-user admin \
--admin-pass \
--database 'mysql' \
--database-name 'nextcloud' \
--database-user 'nextcloud' \
--database-pass \
--database-host 'localhost'
```

■ ■ Cette méthode CLI est recommandée : elle évite les timeouts et offre un retour d'erreur plus précis que l'assistant web.

## 5. Finalisation de l'installation

### 5.1 Ajout du domaine de confiance

Lors du premier accès web, Nextcloud affiche un avertissement « Accès depuis un domaine non approuvé ». Il faut ajouter l'adresse IP du serveur dans le tableau **trusted\_domains** du fichier `config.php`.

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

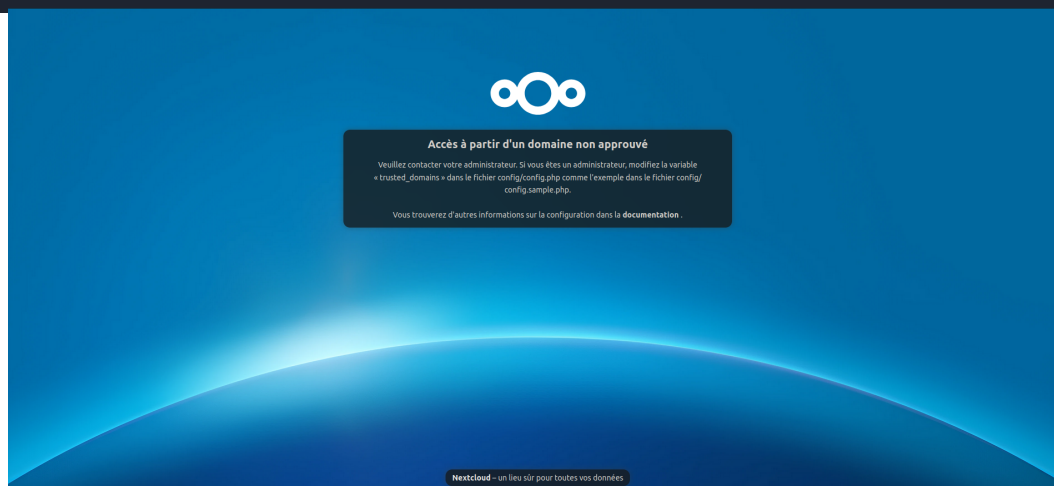


Figure 4 — Fichier `config.php` avec ajout de l'entrée `trusted_domains[1] = '192.168.0.38'`

### 5.2 Accès à l'interface web

Après rechargement de la page, l'interface de connexion Nextcloud est accessible. La connexion s'effectue avec le compte administrateur défini lors de l'installation.

```
<?php
$CONFIG = array (
  'instanceid' => 'ocymj38kscdt',
  'passwordsalt' => '89Y+Aml2Kb7hJjqAGrBkbl58z+N/Ra',
  'secret' => 'Sd3bcmY/h/8c3sYIXq/HyGY82+iCGDhCWF0Y0QWISBJN/QHs',
  'trusted_domains' =>
  array (
    0 => 'localhost',
    1 => '192.168.0.38',
  ),
  'datadirectory' => '/var/www/nextcloud/data',
  'dbtype' => 'mysql',
  'version' => '30.0.9.2',
  'overwrite.cli.url' => 'http://localhost',
  'dbname' => 'nextcloud',
  'dbhost' => 'localhost',
  'dbport' => '',
  'dbtableprefix' => 'oc_',
  'mysql.utf8mb4' => true,
  'dbuser' => 'nextcloud',
  'dbpassword' => 'mariokart8',
  'installed' => true,
);
```

Figure 5 — Page de connexion Nextcloud accessible après correction des `trusted_domains`

### 5.3 Configuration post-installation

Plusieurs paramètres système sont à configurer immédiatement après la première connexion : région téléphonique, pretty URL et redirection HTTPS.

```
# Région par défaut (numéros de téléphone)
sudo -u www-data php /var/www/nextcloud/occ config:system:set \
```

```

default_phone_region --value='FR'

# Pretty URL (suppression de /index.php dans les chemins)
sudo -u www-data php /var/www/nextcloud/occ config:system:set \
  htaccess.RewriteBase --value='/'
sudo -u www-data php /var/www/nextcloud/occ maintenance:update:htaccess

```

## 5.4 Forcer la redirection HTTPS

Une règle de réécriture est ajoutée dans le fichier `.htaccess` de Nextcloud afin de rediriger automatiquement toute connexion HTTP vers HTTPS (code 301).

```

sudo nano /var/www/nextcloud/.htaccess
# Insérer avant :
RewriteCond %{HTTPS} !=on
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]

```

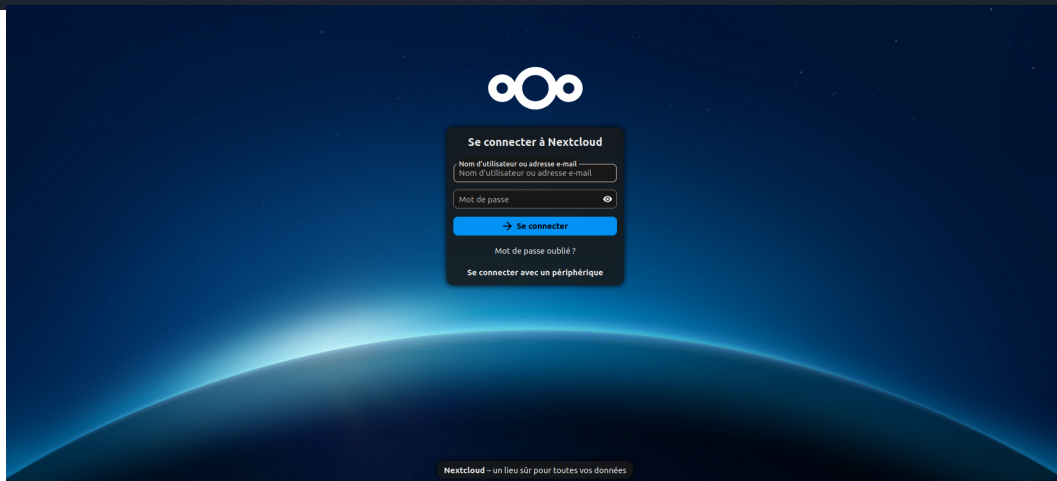


Figure 6 — Règles de réécriture dans `.htaccess` pour la redirection HTTPS

## 5.5 Nettoyage des fichiers skeleton

Par défaut, Nextcloud peuple les comptes de fichiers exemples (documents, photos, vidéo d'introduction). Ces fichiers sont supprimés pour offrir un espace vierge aux nouveaux utilisateurs.

```

sudo rm -r "/var/www/nextcloud/core/skeleton/Documents/"
sudo rm -r "/var/www/nextcloud/core/skeleton/Photos/"
sudo rm "/var/www/nextcloud/core/skeleton/Nextcloud intro.mp4"
sudo rm "/var/www/nextcloud/core/skeleton/Nextcloud.png"

```

## 6. Activation des applications

Nextcloud dispose d'un gestionnaire d'applications accessible via la CLI occ. Dans ce TP, les applications **files external** (stockage externe) et **user ldap** (annuaire LDAP/Active Directory) sont activées.

```
# Activation (si déjà installées)
sudo -u www-data php /var/www/nextcloud/occ app:enable files_external
sudo -u www-data php /var/www/nextcloud/occ app:enable user_ldap

# Installation si nécessaire
sudo -u www-data php /var/www/nextcloud/occ app:install user_ldap
sudo -u www-data php /var/www/nextcloud/occ app:install files_external

# Mise à jour de toutes les applications
sudo -u www-data php /var/www/nextcloud/occ app:update --all
```

### 6.1 Politique de rétention de la corbeille

La durée de conservation des fichiers supprimés dans la corbeille est configurée à 15 jours. Des commandes manuelles permettent de forcer l'expiration si nécessaire.

```
# Retention : automatique avec maximum 15 jours
sudo -u www-data php /var/www/nextcloud/occ config:system:set \
  trashbin_retention_obligation --value='auto, 15'

# Forcer l'expiration des fichiers hors rétention
sudo -u www-data php /var/www/nextcloud/occ trashbin:expire

# Vider définitivement la corbeille de tous les utilisateurs
sudo -u www-data php /var/www/nextcloud/occ trashbin:cleanup --all-users
```

### 6.2 Configuration des tâches planifiées (Cron)

Nextcloud requiert l'exécution régulière de tâches d'arrière-plan (indexation, nettoyage, notifications). Le planificateur cron système est configuré pour lancer ces tâches toutes les 5 minutes.

```
sudo crontab -u www-data -e
# Ajouter à la fin du fichier :
*/5 * * * * php -f /var/www/nextcloud/cron.php
```

```
GNU nano 6.2 /tmp/crontab.qz6vaB/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
*/5 * * * * php -f /var/www/nextcloud/cron.php
```

Figure 7 — Crontab de www-data avec la tâche Nextcloud configurée toutes les 5 minutes

```
# Déclarer cron comme gestionnaire de tâches dans Nextcloud
sudo -u www-data php /var/www/nextcloud/occ background:cron

# Fenêtre de maintenance : 1h du matin (heure UTC)
sudo -u www-data php /var/www/nextcloud/occ config:system:set \
  maintenance_window_start --type=integer --value=1
```

## 7. Paramètres personnels et mise à jour

Dans l'interface web, accéder au menu utilisateur (en haut à droite) puis à **Paramètres personnels** pour vérifier et ajuster la langue et les paramètres régionaux.

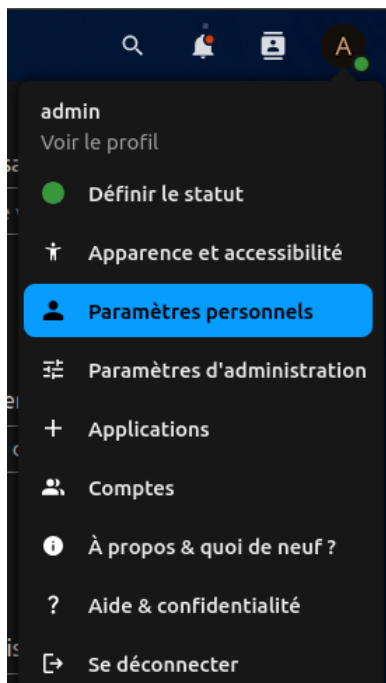


Figure 8 — Menu principal : accès aux paramètres personnels et d'administration

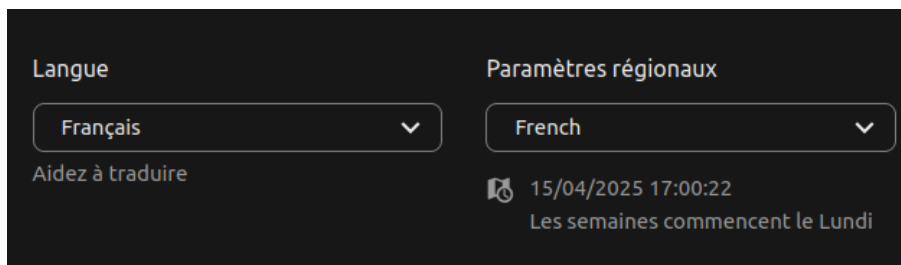


Figure 9 — Paramètres régionaux : langue Français, semaines commençant le lundi

### Vérification des mises à jour disponibles

Le panneau d'administration indique la disponibilité d'une version plus récente. Dans ce TP, Nextcloud 31.0.3 est proposé. La procédure de mise à jour est traitée en section 12.

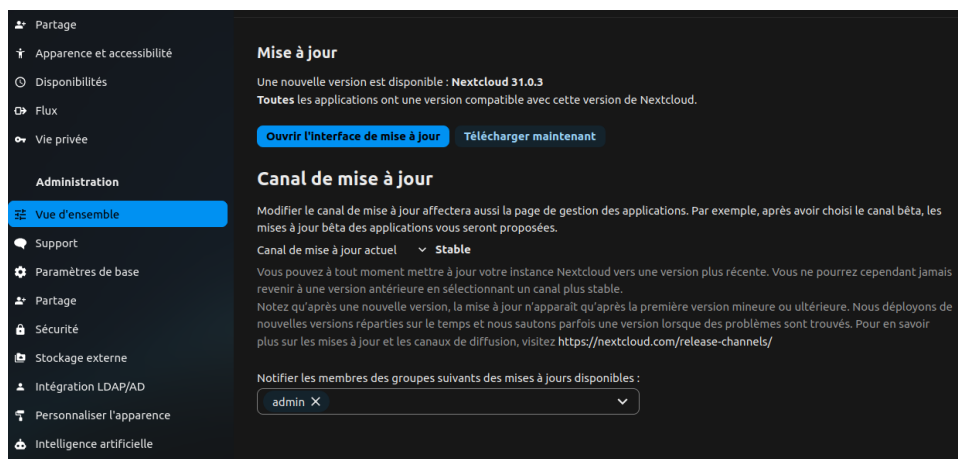


Figure 10 — Notification de mise à jour disponible (Nextcloud 31.0.3)

## 8. Création des utilisateurs et groupes

La gestion des comptes s'effectue depuis le menu **Comptes** de l'interface d'administration. Il est possible de créer des groupes (ex. : admins, utilisateurs, DSI) et d'y affecter les utilisateurs.

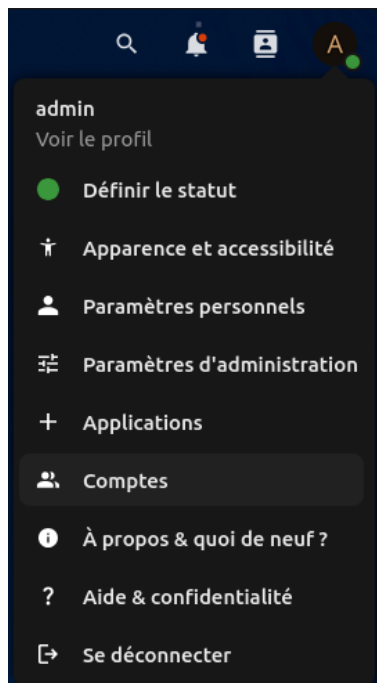


Figure 11 — Menu Comptes : accès à la gestion des utilisateurs

La création d'un groupe s'effectue via le bouton **+** dans le panneau Groupes. Pour créer un utilisateur, cliquer sur **+ Nouveau compte** et renseigner le nom d'affichage, le nom de compte, le mot de passe et l'e-mail.

■ ■ Ne pas oublier d'ajouter l'utilisateur au groupe approprié afin de lui attribuer les rôles d'administrateur si nécessaire.

## 9. Optimisation des performances

Le serveur Nextcloud est à présent fonctionnel. Pour garantir des performances optimales en environnement de production, plusieurs ajustements sont recommandés par la documentation officielle.

### 9.1 Optimisation MariaDB (InnoDB)

```
sudo nano /etc/mysql/conf.d/mysql.cnf
# Ajouter sous [mysqld] :
[mysqld]
innodb_buffer_pool_size = 512M
innodb_io_capacity = 4000

sudo nano /etc/mysql/my.cnf
# Ajouter sous [mysqld] :
[mysqld]
transaction_isolation = READ-COMMITTED
binlog_format = ROW

sudo systemctl restart mariadb.service
```

■ *innodb\_buffer\_pool\_size doit être adapté à la RAM disponible (généralement 50–70 % de la mémoire totale en production).*

### 9.2 PHP — Mémoire et output buffering

La limite mémoire PHP est augmentée à 512 Mo pour les opérations intensives (synchronisation, prévisualisation). L'output buffering est désactivé pour améliorer la réactivité des requêtes.

```
sudo sed -i -e 's/^memory_limit \+= \+.*$/memory_limit = 512M/g' \
/etc/php/8.3/apache2/php.ini

sudo sed -i -e 's/^output_buffering \+= \+.*$/output_buffering = 0/g' \
/etc/php/8.3/apache2/php.ini

sudo systemctl restart apache2.service

# Vérification
cat /etc/php/8.3/apache2/php.ini | egrep 'memory_limit|^output_buffering'
```

### 9.3 Configuration de Redis (cache distribué et file locking)

Redis est utilisé comme système de cache mémoire distribué et pour le verrouillage de fichiers (file locking), ce qui améliore considérablement les performances en accès concurrent. Redis est configuré en mode socket Unix pour éviter la surcharge TCP.

```
# Ajouter www-data au groupe redis
sudo usermod -a -G redis www-data

# Configuration socket Unix et désactivation du port TCP
sudo sed -i -e 's/^#* *port +*6379$/port 0/g' /etc/redis/redis.conf
sudo sed -i -e 's/^#* *unixsocket +*.*$/unixsocket \/var\/run\/redis\/redis-server.sock/g' /
etc/redis/redis.conf
sudo sed -i -e 's/^#* *unixsocketperm+*.*$/unixsocketperm 770/g' /etc/redis/redis.conf
```

```
sudo systemctl restart redis-server.service
```

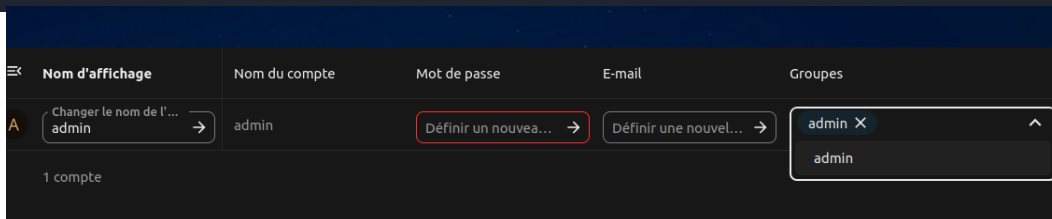


Figure 12 — Vérification de la configuration Redis (socket Unix, port 0)

## 9.4 File Locking et Memory Cache dans Nextcloud

```
# Activation du file locking
sudo -u www-data php /var/www/nextcloud/occ config:system:set \
  filelocking.enabled --value='true'

# Configuration Redis
sudo -u www-data php /var/www/nextcloud/occ config:system:set redis host \
  --value='/var/run/redis/redis-server.sock'
sudo -u www-data php /var/www/nextcloud/occ config:system:set redis port \
  --value='0'
sudo -u www-data php /var/www/nextcloud/occ config:system:set redis timeout \
  --value='0.0'

# Memory cache
sudo -u www-data php /var/www/nextcloud/occ config:system:set \
  memcache.distributed --value='\\OC\\Memcache\\Redis'
sudo -u www-data php /var/www/nextcloud/occ config:system:set \
  memcache.local --value='\\OC\\Memcache\\Redis'
sudo -u www-data php /var/www/nextcloud/occ config:system:set \
  memcache.locking --value='\\OC\\Memcache\\Redis'
```

```
# The MariaDB configuration file
#
# The MariaDB/MySQL tools read configuration files in the following order:
# 0. "/etc/mysql/my.cnf" symlinks to this file, reason why all the rest is read.
# 1. "/etc/mysql/mariadb.cnf" (this file) to set global defaults,
# 2. "/etc/mysql/conf.d/*.cnf" to set global options.
# 3. "/etc/mysql/mariadb.conf.d/*.cnf" to set MariaDB-only options.
# 4. "./.my.cnf" to set user-specific options.
#
# If the same option is defined multiple times, the last one will apply.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# If you are new to MariaDB, check out https://mariadb.com/kb/en/basic-mariadb-articles/
#
# This group is read both by the client and the server
# use it for options that affect everything
#
[client-server]
# Port or socket location where to connect
# port = 3306
socket = /run/mysqld/mysqld.sock

# Import all .cnf files from configuration directory
!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mariadb.conf.d/

[mysqld]
transaction_isolation = READ-COMMITTED
binlog_format = ROW
```

Figure 13 — Vérification config.php : file locking et memcache Redis correctement configurés

## 9.5 Activation de PHP OPcache

OPcache améliore significativement les performances de PHP en compilant et mettant en cache le bytecode des scripts, évitant leur réinterprétation à chaque requête.

```
sudo sed -i -e 's/^;* *opcache\.enable *= *.*$/opcache.enable=1/g' /etc/php/8.3/apache2/php.ini
sudo sed -i -e 's/^;* *opcache\.memory_consumption *= *.*$/opcache.memory_consumption=128/g' /etc/php/8.3/apache2/php.ini
sudo sed -i -e 's/^;* *opcache\.max_accelerated_files *= *.*$/opcache.max_accelerated_files=10000/g' /etc/php/8.3/apache2/php.ini
sudo sed -i -e 's/^;* *opcache\.revalidate_freq *= *.*$/opcache.revalidate_freq=60/g' /etc/php/8.3/apache2/php.ini
sudo sed -i -e 's/^;* *opcache\.save_comments *= *.*$/opcache.save_comments=1/g' /etc/php/8.3/apache2/php.ini
sudo sed -i -e 's/^;* *opcache\.interned_strings_buffer *= *.*$/opcache.interned_strings_buffer=16/g' /etc/php/8.3/apache2/php.ini

sudo systemctl restart apache2.service
```

```
root@modele:/home/test# sudo cat /var/www/nextcloud/config/config.php |
> egrep "'filelocking\.enabled|'memcache\.local\
> |'memcache\.locking|'host|'port|'timeout|'memcache\.distributed"
'filelocking.enabled' => 'true',
'host' => '/var/run/redis/redis-server.sock',
'port' => '0',
'timeout' => '0.0',
'memcache.distributed' => '\\OC\\Memcache\\Redis',
'memcache.local' => '\\OC\\Memcache\\Redis',
'memcache.locking' => '\\OC\\Memcache\\Redis',
root@modele:/home/test#
```

Figure 14 — Valeurs OPcache vérifiées dans php.ini

## 9.6 Taille limite de téléchargement

Par défaut, PHP limite la taille des fichiers uploadés à 2 Mo. Pour un usage professionnel, cette limite est portée à 16 Go afin de permettre le transfert de fichiers volumineux.

```
sudo sed -i -e 's/^;* *upload_max_filesize *= *.*$/upload_max_filesize=16G/g' /etc/php/8.3/apache2/php.ini
sudo sed -i -e 's/^;* *post_max_size *= *.*$/post_max_size=16G/g' /etc/php/8.3/apache2/php.ini

sudo systemctl restart apache2.service
```

```
> |^;* *opcache\.interned_strings_buffer*="
opcache.enable=1
opcache.memory_consumption=128
opcache.max_accelerated_files=10000
opcache.revalidate_freq=60
opcache.save_comments=1
root@modele:/home/test#
```

Figure 15 — Vérification : upload\_max\_filesize et post\_max\_size à 16G

## 10. Certificat SSL avec Let's Encrypt

Let's Encrypt permet d'obtenir gratuitement un certificat TLS valide et reconnu par les navigateurs. Certbot automatise la génération, la configuration Apache et le renouvellement automatique.

### 10.1 Installation de Certbot

```
sudo apt update -y
sudo apt install -y software-properties-common
sudo add-apt-repository universe
sudo apt update -y
sudo apt install -y certbot python3-certbot-apache
```

### 10.2 Génération du certificat

```
sudo certbot --apache -d nextcloud.domaine.fr \
-m votre@adresse-email.fr --no-redirect
```

■ ■ Remplacer *nextcloud.domaine.fr* par le nom de domaine réel pointant vers le serveur. Un enregistrement DNS de type A doit préalablement exister.

### 10.3 Renouvellement automatique

Certbot configure automatiquement un timer systemd pour renouveler le certificat avant son expiration (90 jours). Il est recommandé de tester cette procédure avec l'option `--dry-run`.

```
# Test de renouvellement (sans effectuer de changement réel)
sudo certbot renew --dry-run

# Visualiser le timer de renouvellement
sudo systemctl list-timers certbot
```

### 10.4 Ajout du domaine dans Nextcloud

```
# Lister les domaines de confiance actuels
sudo -u www-data php /var/www/nextcloud/occ config:system:get trusted_domains

# Ajouter le domaine Let's Encrypt
sudo -u www-data php /var/www/nextcloud/occ config:system:set \
trusted_domains 1 --value=nextcloud.domaine.fr
```

## 11. Sauvegarde de Nextcloud

La stratégie de sauvegarde couvre deux éléments distincts : les **fichiers** (répertoire `/var/www/nextcloud`) et la **base de données** MariaDB. Le mode maintenance est activé pendant la sauvegarde pour garantir la cohérence des données.

### 11.1 Préparation des répertoires

```
sudo mkdir -p /backups/nextcloud/files
sudo mkdir -p /backups/nextcloud/database
sudo chmod -R o-rwx /backups/
```

### 11.2 Activation du mode maintenance

```
sudo -u www-data php /var/www/nextcloud/occ maintenance:mode --on
sudo -u www-data php /var/www/nextcloud/occ maintenance:mode # vérification
```

### 11.3 Sauvegarde des fichiers

```
sudo rsync -Aavx /var/www/nextcloud/ \
/backups/nextcloud/files/nextcloud-backup_`date +%Y%m%d_%H%M%S`/ \
--exclude={'data/*','*/files trashbin/files/*'}
```

■ Supprimer l'option `--exclude` pour inclure le répertoire `data` et la corbeille dans la sauvegarde (sauvegarde complète).

### 11.4 Sauvegarde de la base de données

```
sudo mysqldump --single-transaction -h localhost -u root nextcloud | \
sudo tee /backups/nextcloud/database/nextcloud-sqlbkp_`date +%Y%m%d_%H%M%S`.bak
```

### 11.5 Désactivation du mode maintenance

```
sudo -u www-data php /var/www/nextcloud/occ maintenance:mode --off
```

## 12. Mise à jour de Nextcloud

Nextcloud intègre un programme de mise à jour interne (updater.php) qui automatise l'ensemble du processus : vérification des fichiers, téléchargement, vérification d'intégrité, extraction, activation du mode maintenance, remplacement des fichiers et finalisation.

```
sudo -u www-data php /var/www/nextcloud/updater/updater.php
# Confirmer le démarrage de la mise à jour avec : y
```

```
root@modele:/home/test# sudo -u www-data php /var/www/nextcloud/occ maintenance:mode --off
Maintenance mode disabled
root@modele:/home/test# sudo -u www-data php /var/www/nextcloud/updater/updater.php
Nextcloud Updater - version: v29.0.0-23-g6a5f9ec

Step 5 is currently in process. Please call this command later or remove the following file to start from scratch: /var/www/nextcloud/data/updater-ocynj38kscdt/.step
root@modele:/home/test# sudo rm /var/www/nextcloud/data/updater-ocynj38kscdt/.step
rm: Impossible de supprimer "/var/www/nextcloud/data/updater-ocynj38kscdt/.step": aucun fichier ou dossier de ce nom
root@modele:/home/test# sudo rm /var/www/nextcloud/data/updater-ocynj38kscdt/.step
root@modele:/home/test# sudo -u www-data php /var/www/nextcloud/updater/updater.php
Nextcloud Updater - version: v29.0.0-23-g6a5f9ec

Current version is 30.0.9.

Update to Nextcloud 31.0.3 available. (channel: "stable")
Following file will be downloaded automatically: https://download.nextcloud.com/server/releases/nextcloud-31.0.3.zip
Open changelog ↗

Steps that will be executed:
[ ] Check for expected files
[ ] Check for write permissions
[ ] Create backup
[ ] Downloading
[ ] Verify integrity
[ ] Extracting
[ ] Enable maintenance mode
[ ] Replace entry points
[ ] Delete old files
[ ] Move new files in place
[ ] Done

Start update? [y/N] 
```

Figure 16 — Processus de mise à jour : toutes les étapes validées, 'Update of code successful'

■ ■ En cas d'erreur de mise à jour sur un step précédent, supprimer le fichier .step dans /var/www/nextcloud/data/updater-ocmj38kscdt/ avant de relancer.

## Conclusion

Ce TP a permis de déployer et configurer une instance Nextcloud 30 complète sur Ubuntu 22.04, en suivant les bonnes pratiques recommandées par la documentation officielle :

✓ <b>Installation</b>	Pile LAMP + PHP 8.3 avec toutes les extensions requises
✓ <b>Configuration</b>	VirtualHost Apache, trusted_domains, HTTPS forcé
✓ <b>Base de données</b>	MariaDB sécurisée avec isolation READ-COMMITTED
✓ <b>Cache</b>	Redis en mode socket Unix (memcache + file locking)
✓ <b>Performance</b>	OPcache PHP, memory_limit 512 Mo, upload 16 Go
✓ <b>Sécurité</b>	Certificat Let's Encrypt via Certbot, HSTS
✓ <b>Exploitation</b>	Cron système, politique de rétention corbeille, sauvegarde

L'instance est prête pour un usage en production ou en environnement de laboratoire. Pour aller plus loin, il est recommandé de configurer l'intégration LDAP/Active Directory, le stockage externe (S3, SMB) et la surveillance des performances via l'application Monitoring.