SODECAF – Tutoriel Installation Nextcloud Sécurisée

Phase 1 : Installation de Base Nextcloud

Pourquoi cette phase ?

Avant d'installer Nextcloud, nous devons préparer l'environnement système : dependencies, serveur web (Apache), langage (PHP), et base de données (MariaDB).

1.1 Mise à jour du système

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

1.2 Installation des dépendances

```
apt install -y apache2 php php-cli php-gd php-mysql php-curl php-xml php-mbstring php-zip php-intl php-bcmath php-gmp php-imagick php-ldap mariadb-server
wget unzip fail2ban net-tools lsof smartmontools
```

1.3 Activation modules Apache

```
a2enmod rewrite headers env dir mime setenvif ssl http2 proxy proxy_http
```

1.4 Redémarrage Apache

```
systemctl restart apache2
```

1.5 Création base de données MariaDB

```
mysql -u root
```

Puis dans le prompt MySQL :

```
CREATE DATABASE nextcloud;
CREATE USER 'nextcloud'@'localhost' IDENTIFIED BY
'MotDePasseSecurise123!';
GRANT ALL PRIVILEGES ON nextcloud.* TO 'nextcloud'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

1.6 Téléchargement Nextcloud

```
cd /var/www
wget https://download.nextcloud.com/server/releases/nextcloud-32.0.2.zip
```

1.7 Extraction et permissions

```
unzip nextcloud-32.0.2.zip
chown -R www-data:www-data /var/www/nextcloud
chmod -R 755 /var/www/nextcloud
```

Phase 2 : Certificat SSL Auto-signé

2.1 Création répertoire certificats

```
mkdir -p /etc/ssl/private/nextcloud.team21.local
cd /etc/ssl/private/nextcloud.team21.local
```

2.2 Génération clé et certificat

```
openssl req -x509 -nodes -days 825 -newkey rsa:4096 -keyout
nextcloud.team21.local.key -out nextcloud.team21.local.crt  -subj
"/C=FR/L=Brive-la-Gaillarde/O=Team21/CN=nextcloud.team21.local"
```

Phase 3 : Configuration Apache HTTPS

3.1 Création VirtualHost HTTPS

/etc/apache2/sites-available/nextcloud.conf

Contenu détaillé :

```
<VirtualHost *:80>
   ServerName nextcloud.team21.local
ServerAdmin admin@team21.local
   DocumentRoot /var/www/nextcloud

Redirect /  https://localhost/

  <Directory /var/www/nextcloud>
      Require all granted
      AllowOverride All
      Options FollowSymLinks MultiViews
      <IfModule mod_dav.c>
         Dav off
      </IfModule>
   </Directory>

   ErrorLog ${APACHE_LOG_DIR}/nextcloud_error.log
   CustomLog ${APACHE_LOG_DIR}/nextcloud_access.log combined
</VirtualHost>

<VirtualHost *:443>
   ServerName nextcloud.team21.local
ServerAdmin admin@team21.local
   DocumentRoot /var/www/nextcloud

   SSLEngine on
   SSLCertificateFile
/etc/ssl/private/nextcloud.team21.local/nextcloud.team21.local.crt
   SSLCertificateKeyFile
/etc/ssl/private/nextcloud.team21.local/nextcloud.team21.local.key
   SSLProtocol -all +TLSv1.3 +TLSv1.2
   SSLCipherSuite ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256
```

```
GCM-SHA384
    SSLHonorCipherOrder on
    SSLCompression off

    Header always set Strict-Transport-Security ”max-age=31536000;
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”

    <Directory /var/www/nextcloud>
        Require all granted
        AllowOverride All
        Options FollowSymLinks MultiViews
        <IfModule mod_dav.c>
            Dav off
        </IfModule>
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/nextcloud_error.log
    CustomLog ${APACHE_LOG_DIR}/nextcloud_access.log combined
</VirtualHost>
```

3.2 Activation site et redémarrage

```
a2ensite nextcloud.conf
a2dissite 000-default.conf
systemctl reload apache2
```

Phase 4: Installation Nextcloud (Interface Web)

4.1 Accès installation

https://localhost

Alerte navigateur :

- ⚠️ "Connexion non sécurisée" (certificat auto-signé)

- ✓ Cliquer "Continuer" ou "Ajouter exception" (normal intranet)

4.2 Formulaire installation

Créer compte admin :

- Username : admin
- Password : Complexe (12+ chars, symboles)
- **Sécurité** : Seul accès root Nextcloud

Données répertoire :

- Laisser par défaut /var/www/nextcloud/data
- Nextcloud crée automatiquement

Base de données :

- **Type** : MariaDB
- **Host** : localhost
- **BD** : nextcloud
- **User** : nextcloud
- **Password** : MotDePasseSecurise123! (configuré phase 1)

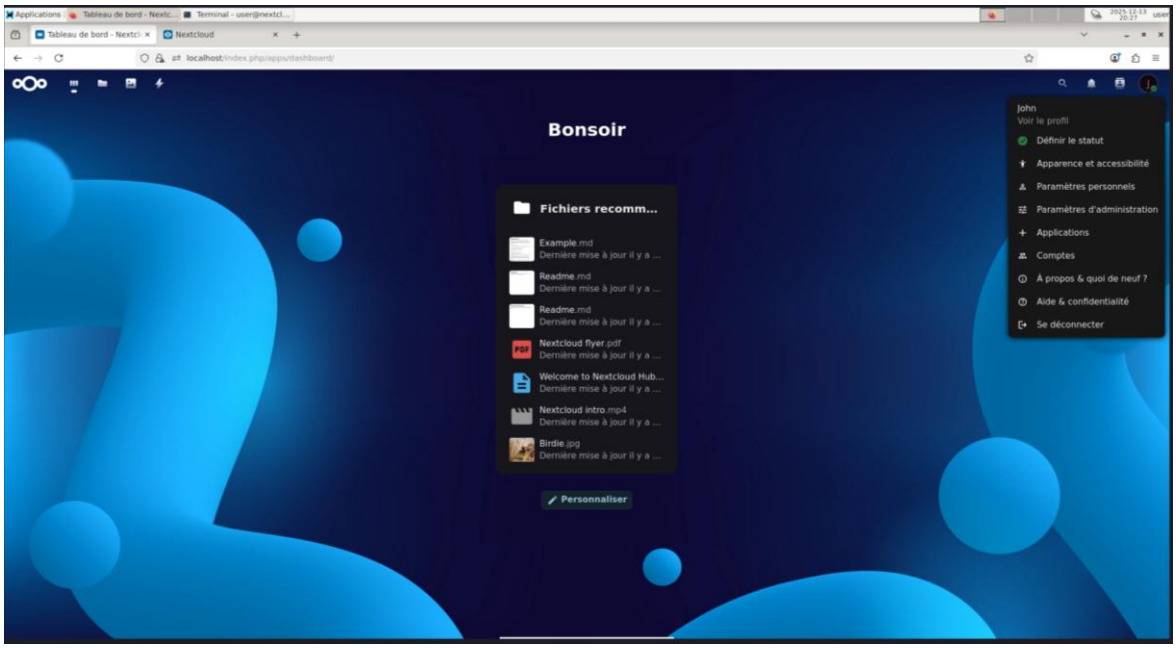
Valider :

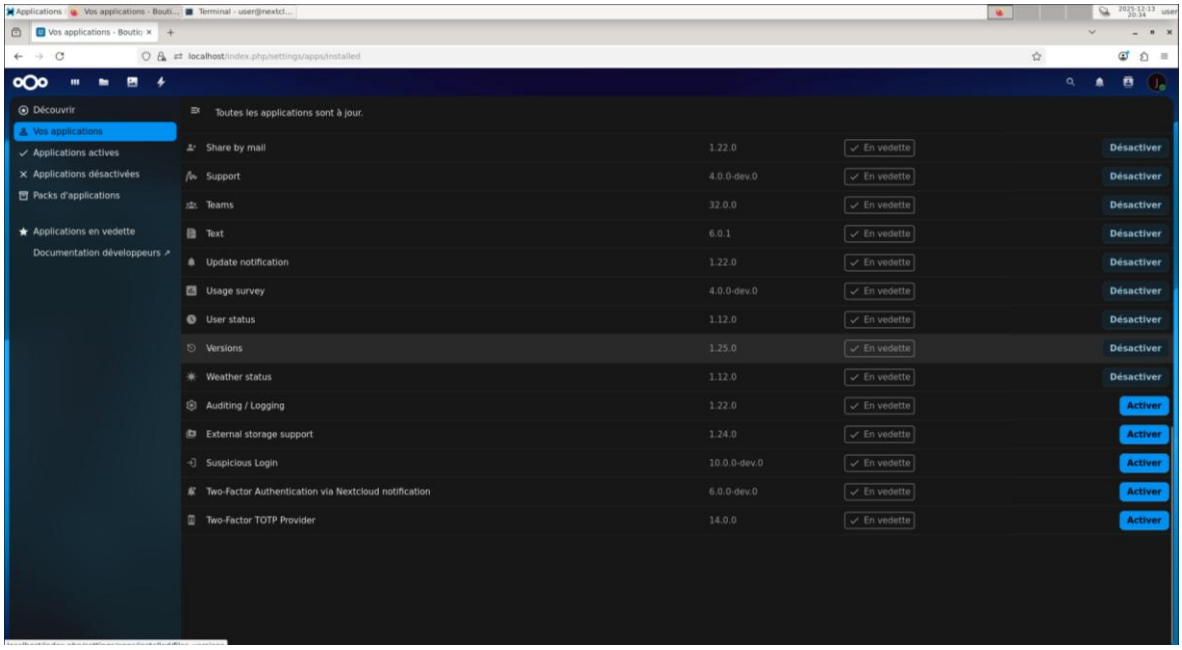
- Installation crée tables, config
- Peut prendre 1-2 minutes
- ✓ Redirection vers login

Phase 5 : Sécurisation Avancée

5.1 Activation 2FA (TOTP)

Rendez-vous dans l'onglet **Applications**, puis cherchez **Two-Factor TOTP Provider** et appuyez sur **Activer**.





5.2 Intégration LDAPS/AD

1. Installation de l'Autorité de Certification (CA) sur le DC

1. Sur le contrôleur de domaine, ouvrir le **Gestionnaire de serveur**.
2. Ajouter le rôle **Services de certificats Active Directory (AD CS)**.
3. Choisir:
    - Type de configuration : **Autorité de certification**.
    - Type d'AC: **Entreprise**.
    - Type: **Root CA**.
4. Définir:
    - Nom commun de l'AC (par ex. MonDomaine-CA).
    - Période de validité (par défaut 5 ans, suffisant dans la plupart des cas).
5. Finaliser l'installation.

2. Export du certificat SSL depuis le DC

1. Sur le DC, ouvrir la **console de gestion des certificats** (certsrv.msc / mmc + snap-in Certificats selon le cas).
2. Localiser le certificat racine de l'AC.
3. L'exporter au format:
    - **Base-64 encodé X.509 (.CER)**, compatible Linux.
4. Enregistrer le fichier (par ex. dc-ca.cer) dans un emplacement accessible (bureau, etc.).

3. Transfert et installation du certificat sur le serveur NextCloud

1. Depuis votre poste, ouvrir **WinSCP**.
2. Se connecter au serveur NextCloud (SFTP).
3. Transférer le fichier dc-ca.cer vers le serveur (par ex. dans /home/nextcloud/ ou /root/).
4. Se connecter en SSH (PuTTY) au serveur NextCloud.
5. Déplacer le certificat dans le répertoire des certificats de confiance

```
mv /chemin/ldaps-team21-ca.cer /usr/local/share/ca-certificates/ldaps-team21-ca.crt
```

6. Mettre à jour le magasin de certificats :

```
update-ca-certificates
```

6. Test de la connexion LDAPS depuis le serveur NextCloud

1. Depuis le serveur NextCloud (en SSH), tester la connexion LDAPS au DC :

```
ldapsearch -H  ldaps://team21-1.team21.local:636 -x -W –D "CN=Administrateur,CN=Users,DC=TEAM21,DC=local" -b "dc=TEAM21,dc=local"
```

2. Vérifier dans la sortie qu'on obtient un résultat du type :
    - Verify return code: 0 (ok)
3. Si ce n'est pas OK, vérifier : chemin du certificat, update-ca, FQDN, pare-feu, port 636 ouvert.

7. Création du compte de service AD pour NextCloud

1. Sur le DC, ouvrir **Utilisateurs et ordinateurs Active Directory**.
2. Créer un nouvel utilisateur (par ex. svc_nextcloud).
3. Définir un mot de passe robuste.
4. Cocher **« Le mot de passe n'expire jamais »** (ou gérer un process de rotation contrôlé).
5. Optionnel : le placer dans une OU dédiée aux comptes de service.

8. Configuration de l'intégration LDAP/AD dans NextCloud

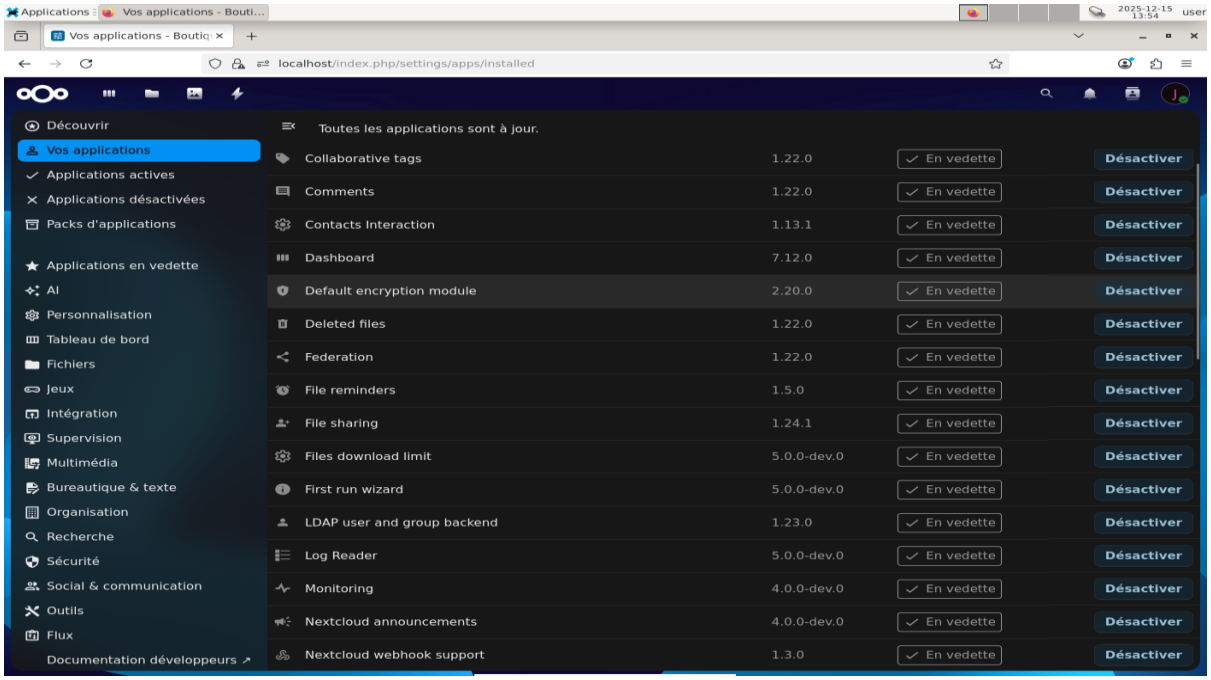
1. Se connecter à l'interface d'**administration NextCloud** avec un compte admin.
2. Aller dans les paramètres d'administration, section **LDAP / AD Integration** (ou similaire).
3. Créer une nouvelle configuration LDAP.
4. Paramètres de connexion :
    - Hôte : ldaps://team21-1.team21.local

- Port : 636
- Sécurité : **LDAPS** (chiffrement SSL/TLS).

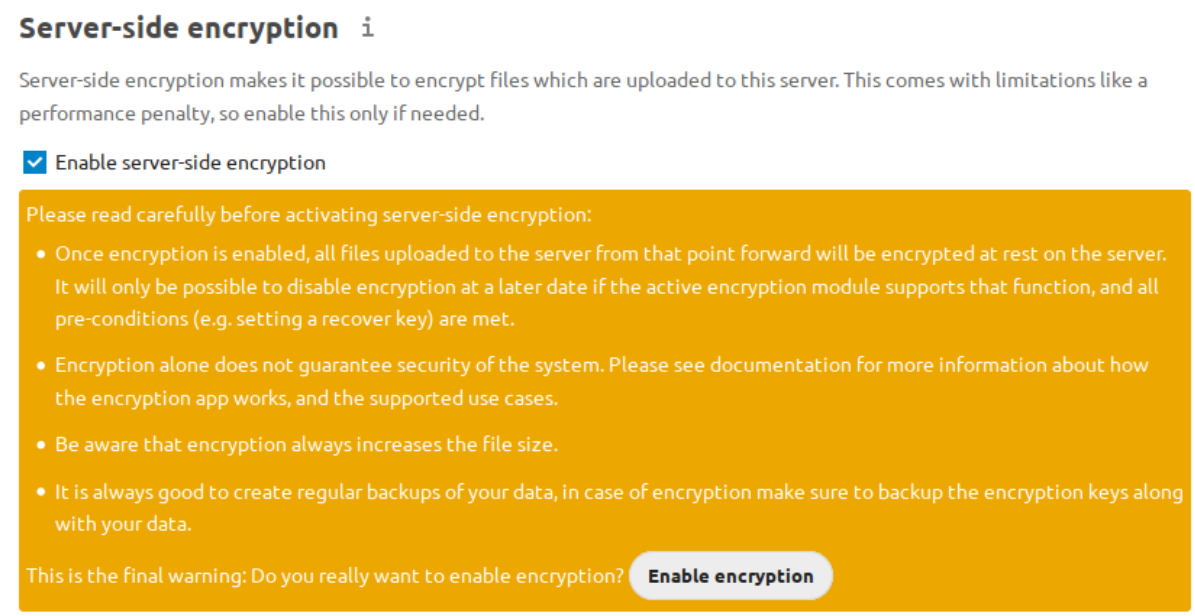
5. Saisir le DN de liaison (Bind DN) du compte de service :
    - CN=Administrateur,CN=Users,DC=TEAM21,DC=local
6. Saisir le mot de passe du compte de service.
7. Tester la connexion depuis l'interface : le test doit être **réussi**.

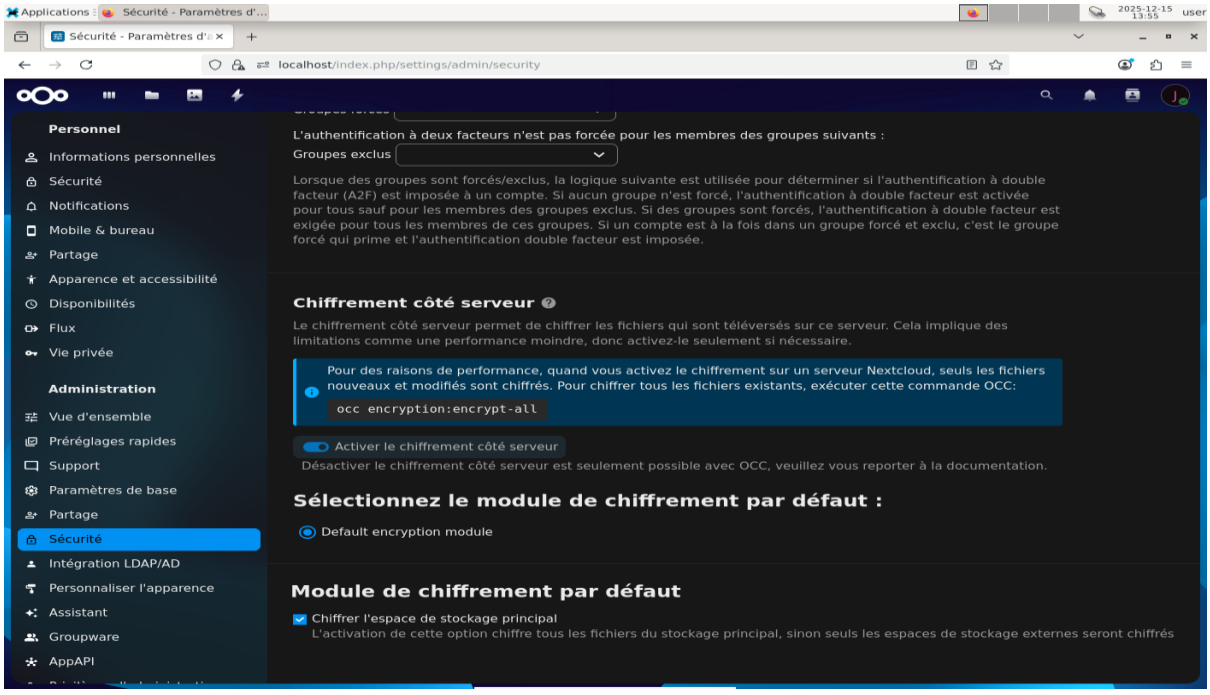
5.3 Chiffrement côté serveur

Rendez-vous dans l'onglet **Applications**, puis cherchez **Default encryption module** et appuyez sur **Activer**.



Acceptez la pop-up.





-En ligne de commande en su - :

cd /var/www/nextcloud
php occ encryption:enable
occ encryption:encrypt-all

You are about to start to encrypt all files stored in your Nextcloud.
It will depend on the encryption module you use which files get encrypted.
Depending on the number and size of your files this can take some time.
Please make sure that no users access their files during this process!

Do you really want to continue? (y/n)

Tapez **entrer** et laissez faire. Cela peut prendre du temps.

5.4 Définir quotas

Un par un :

occ user:modify --quota="50GB" nomutilisateur

5.5 SMART Monitoring

1. Configuration monitoring SMART

-Vérifier que SMART est actif avec la commande :

smartctl -i /dev/sda

-Activer si nécessaire :

smartctl -s on /dev/sda

-Créer script monitoring:

nano /usr/local/bin/check-smart.sh

À l'intérieur insérer le code suivant :

```
#!/bin/bash

DEVICES=("/dev/sda" "/dev/sdb")
for device in "${DEVICES[@]}"; do
STATUS=$(smartctl -H $device | grep "PASSED|FAILED")
if [[ $STATUS == "FAILED" ]]; then
echo "ALERTE: Disque $device défaillant !" | mail admin@sodecaf.local
fi
done
```

2. Automatisation :

-Utilisons Cron pour automatiser la tâche (toute les 6h) :

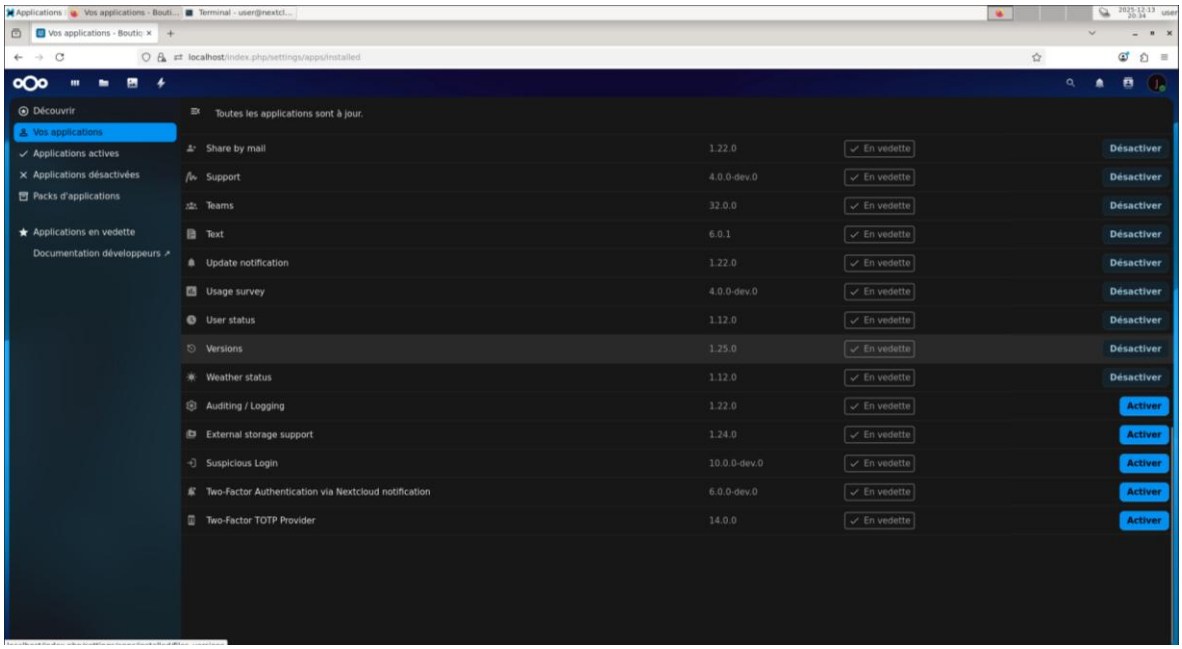
crontab -e

Ajouter :

0 */6 * * * /usr/local/bin/check-smart.sh

5.6 Audit et Logs

Rendez-vous dans l'onglet **Applications**, puis cherchez **Auditting / Logging** et appuyez sur **Activer**.



5.7 Fail2ban

1. Via Nextcloud

Par défaut, Nextcloud propose une sécurité contre les attaque brute-force. Il suffit juste d'activer dans **Applications** l'extension **Brute force.**

6.2 Avec Fail2ban

Il faut en premier lieu créer un filtre dédié au site nextcloud :

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

Avec comme contenu:

```
[Definition]
failregex = ^.*"user":"<HOST>.*401.*$
ignoreregex =
```

Il faut ensuite créer les règles de filtrage :

nano /etc/fail2ban/jail.d/nextcloud.conf

Avec comme contenu:

```
[nextcloud]
enabled = true
port = http,https
filter = nextcloud
logpath = /var/www/nextcloud/data/nextcloud.log
maxretry = 5
findtime = 1800
bantime = 3600
```

Explication :

maxretry = 5 : 5 tentatives avant blocage

findtime = 1800 : Fenêtre 30 min

bantime = 3600 : Bloquer 1 heure

Puis redémarrer Apache :

systemctl restart fail2ban

Phase 7 : Sauvegardes et Récupération

7.1 Montage NAS

1. Installez les dépendances nécessaires :

apt install -y nfs-common cifs-utils

2. Créer point de montage

mkdir -p /mnt/nas-backup

chmod 770 /mnt/nas-backup

3. Monter NAS (exemple NFS - adapter selon votre config)

mount -t nfs 10.200.200.200:/data/AT2TEAM21 /mnt/nas-backup

4. Vérifier

mount | grep nas-backup

5. Rendre permanent (fstab)

nano /etc/fstab

-Ajouter la ligne :

10.200.200.200:/data/AT2TEAM21 /mnt/nas-backup nfs rw,vers=3,_netdev,noatime,sync 0 0

6. Vérifier mount au boot

mount –a

7.2 Script sauvegarde complète quotidienne

1. Créer script

nano /usr/local/bin/backup-nextcloud-full.sh

Contenu:

```
#!/bin/bash
set -euo pipefail
NC_PATH=”/var/www/nextcloud”
```

```
BACKUP_PATH="/mnt/nas-backup/AT2TEAM21/Backup"
DATE=$(date +%Y%m%d-%H%M%S)
LOG_FILE="/var/log/nextcloud-backup.log"

DB_NAME="nextcloud"
DB_USER="nextcloud"
DB_PASS="1234"

exec >> "$LOG_FILE" 2>&1

echo "[$(date)] === DÉMARRAGE SAUVEGARDE NEXTCLOUD ==="

# Maintenance ON
echo "[$(date)] Activation mode maintenance"
su -s /bin/bash www-data -c "php $NC_PATH/occ maintenance:mode --on"

# DB
echo "[$(date)] Sauvegarde base de données"
mysqldump -u"$DB_USER" -p"$DB_PASS" "$DB_NAME" | gzip >
"$BACKUP_PATH/nextcloud-db-$DATE.sql.gz"

# Config
echo "[$(date)] Sauvegarde configuration"
tar -czf "$BACKUP_PATH/nextcloud-config-$DATE.tar.gz" ¥
   -C "$NC_PATH" config/ ¥
   --checkpoint=5000 ¥
   --checkpoint-action=echo="[$(date)] config toujours en cours..."

# Data
echo "[$(date)] Sauvegarde données utilisateurs"
tar -czf "$BACKUP_PATH/nextcloud-data-$DATE.tar.gz" ¥
   -C "$NC_PATH" data/  ¥
   --checkpoint=10000 ¥
   --checkpoint-action=echo="[$(date)] data toujours en cours..."

# Maintenance OFF
echo "[$(date)] Désactivation mode maintenance"
su -s /bin/bash www-data -c "php $NC_PATH/occ maintenance:mode --off"

# Nettoyage
echo "[$(date)] Nettoyage backups > 7 jours"
find "$BACKUP_PATH" -type f -mtime +7 -delete

echo "[$(date)] === SAUVEGARDE TERMINÉE AVEC SUCCÈS ==="
```

2. Rendre executable

chmod +x /usr/local/bin/backup-nextcloud-full.sh

-Cron : chaque jour à 22h

crontab -e

-Ajouter:

0 22 * * * /usr/local/bin/backup-nextcloud-full.sh

6.3 Script sauvegarde incrémentielle horaire

1. Créer script

nano /usr/local/bin/nextcloud-backup-incremental.sh

-Contenu:

```
#!/bin/bash
set -euo pipefail

# ================================
# VARIABLES
# ================================
NC_DATA="/var/www/nextcloud/data"
BACKUP_PATH="/mnt/nas-backup/AT2TEAM21/Backup"
DATE=$(date +%Y%m%d-%H%M%S)
LOG_FILE="/var/log/nextcloud-backup.log"
FULL_SCRIPT="/usr/local/bin/backup-nextcloud-full.sh"

# ================================
# LOGS : terminal + fichier
# ================================
exec > >(tee -a "$LOG_FILE") 2>&1

echo "[$(date)] === DÉMARRAGE SAUVEGARDE INCRÉMENTIELLE ==="

# ================================
# DÉTERMINER DERNIER BACKUP COMPLET
```

```
# ================================
LAST_FULL_FILE=$(ls -t "$BACKUP_PATH"/nextcloud-data-*.tar.gz 2>/dev/null | head -1 || true)

if [ -z "$LAST_FULL_FILE" ]; then
    echo "[$(date)] Aucun backup complet trouvé"
    echo "[$(date)] Lancement d'un backup complet forcé"
    exec "$FULL_SCRIPT"
fi

LAST_FULL_DATE=$(stat -c %y "$LAST_FULL_FILE")

echo "[$(date)] Dernier backup complet : $LAST_FULL_FILE"
echo "[$(date)] Date de référence : $LAST_FULL_DATE"

# ================================
# COMPTER LES FICHIERS MODIFIÉS
# ================================
echo "[$(date)] Comptage des fichiers modifiés depuis le dernier backup..."
TOTAL_FILES=$(find "$NC_DATA" -type f -newermt "$LAST_FULL_DATE" | wc -l)

if [ "$TOTAL_FILES" -eq 0 ]; then
    echo "[$(date)] Aucun fichier modifié depuis le dernier backup"
    echo "[$(date)] === FIN SAUVEGARDE INCRÉMENTIELLE ==="
    exit 0
fi

echo "[$(date)] $TOTAL_FILES fichiers à sauvegarder"

# ================================
# SAUVEGARDE INCRÉMENTIELLE AVEC PROGRESSION
# ================================
find "$NC_DATA" -type f -newermt "$LAST_FULL_DATE" -print0 ¥
| pv -0 -l -s "$TOTAL_FILES" ¥
| tar --null -T - -czf "$BACKUP_PATH/nextcloud-data-incr-$DATE.tar.gz"

# ================================
# FIN
# ================================
echo "[$(date)] Sauvegarde incrémentielle terminée avec succès"
echo "[$(date)] === FIN SAUVEGARDE INCRÉMENTIELLE ==="
```

2. Rendre exécutable

```
chmod +x /usr/local/bin/nextcloud-backup-incremental.sh
```

-Cron : chaque heure

crontab -e

Ajouter:

0 * * * * /usr/local/bin/backup-nextcloud-incr.sh

7.3 Script de vérification des sauvegardes

1. Créer script

nano /usr/local/bin/check-nextcloud-backups.sh

```
#!/bin/bash
set -euo pipefail

# ================================
# VARIABLES
# ================================
BACKUP_PATH="/mnt/nas-backup/AT2TEAM21/Backup"
LOG_FILE="/var/log/nextcloud-backup-check.log"
MAX_AGE_DAYS=1   # âge max autorisé

# ================================
# LOGS : terminal + fichier
# ================================
exec > >(tee -a "$LOG_FILE") 2>&1

echo "[$(date)] === DÉBUT VÉRIFICATION SAUVEGARDES NEXTCLOUD ==="

ERRORS=0

# ================================
# FONCTION ERREUR
# ================================
fail() {
  echo "[$(date)] ❌ ERREUR : $1"
  ERRORS=$((ERRORS + 1))
}

ok() {
  echo "[$(date)] ✅ OK : $1"
}
```

```
# ================================
# VÉRIFIER PRÉSENCE DES FICHIERS
# ================================
LATEST_DB=$(ls -t "$BACKUP_PATH"/nextcloud-db-*.sql.gz 2>/dev/null | head -1 || true)
LATEST_CONFIG=$(ls -t "$BACKUP_PATH"/nextcloud-config-*.tar.gz 2>/dev/null | head -1 || true)
LATEST_DATA=$(ls -t "$BACKUP_PATH"/nextcloud-data-*.tar.gz 2>/dev/null | head -1 || true)

[ -z "$LATEST_DB" ] && fail "Aucun dump DB trouvé" || ok "Dump DB trouvé"
[ -z "$LATEST_CONFIG" ] && fail "Aucune archive config trouvée" || ok "Archive config trouvée"
[ -z "$LATEST_DATA" ] && fail "Aucune archive data trouvée" || ok "Archive data trouvée"


# ================================
# ÂGE DES SAUVEGARDES
# ================================
check_age() {
  FILE="$1"
  LABEL="$2"

  AGE_DAYS=$(( ( $(date +%s) - $(stat -c %Y "$FILE") ) / 86400 ))
  if [ "$AGE_DAYS" -gt "$MAX_AGE_DAYS" ]; then
    fail "$LABEL trop ancienne (${AGE_DAYS} jours)"
  else
    ok "$LABEL récente (${AGE_DAYS} jours)"
  fi
}

[ -n "$LATEST_DB" ] && check_age "$LATEST_DB" "Dump DB"
[ -n "$LATEST_CONFIG" ] && check_age "$LATEST_CONFIG" "Archive config"
[ -n "$LATEST_DATA" ] && check_age "$LATEST_DATA" "Archive data"

# ================================
# INTÉGRITÉ DES ARCHIVES
# ================================
echo "[$(date)] Vérification intégrité des archives..."

# DB
if [ -n "$LATEST_DB" ]; then
  if gzip -t "$LATEST_DB"; then
    ok "Dump DB intègre"
  else
```

```
        fail "Dump DB corrompu"
    fi
fi

# Config
if [ -n "$LATEST_CONFIG" ]; then
    if tar -tzf "$LATEST_CONFIG" >/dev/null; then
        ok "Archive config intègre"
    else
        fail "Archive config corrompue"
    fi
fi

# Data
if [ -n "$LATEST_DATA" ]; then
    if tar -tzf "$LATEST_DATA" >/dev/null; then
        ok "Archive data intègre"
    else
        fail "Archive data corrompue"
    fi
fi

# ================================
# TAILLE NON NULLE
# ================================
check_size() {
    FILE="$1"
    LABEL="$2"
    SIZE=$(stat -c %s "$FILE")
    if [ "$SIZE" -le 1024 ]; then
        fail "$LABEL trop petite (${SIZE} octets)"
    else
        ok "$LABEL taille OK (${SIZE} octets)"
    fi
}

[ -n "$LATEST_DB" ] && check_size "$LATEST_DB" "Dump DB"
[ -n "$LATEST_CONFIG" ] && check_size "$LATEST_CONFIG" "Archive config"
[ -n "$LATEST_DATA" ] && check_size "$LATEST_DATA" "Archive data"

# ================================
# RÉSULTAT FINAL
# ================================
if [ "$ERRORS" -eq 0 ]; then
    echo "[$(date)]  TOUTES LES VÉRIFICATIONS SONT OK"
    exit 0
```

```
else
   echo "[$(date)]  $ERRORS ERREUR(S) DÉTECTÉE(S)"
   exit 1
Fi
```

2. Rendre exécutable

```
chmod +x /usr/local/bin/check-nextcloud-backups.sh
```

-Cron : chaque heure

```
crontab -e
```

-Ajouter :

```
0 23 * * * /usr/local/bin/check-nextcloud-backups.sh
```

6.4 Restauration d'urgence

Scénario : Ransomware a chiffré disque, besoin de restaurer depuis backup

Étapes:

1. **Arrêter Nextcloud**:

```
systemctl stop apache2
```

2. **Restaurer base de données**:

```
mysql -u nextcloud -p votremotdepasse nextcloud < <(gunzip -c /mnt/nas-backup/nextcloud-db-20251207-220000.sql.gz)
```

3. **Restaurer fichiers**:

```
tar -xzf /mnt/nas-backup/nextcloud-data-20251207-220000.tar.gz -C /var/data/
```

4. **Permissions correctes**:

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

5. **Redémarrer**:

```
systemctl start apache2
```

6. **Vérifier**:

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