PortailSIO

NOUVELLE FORMULE

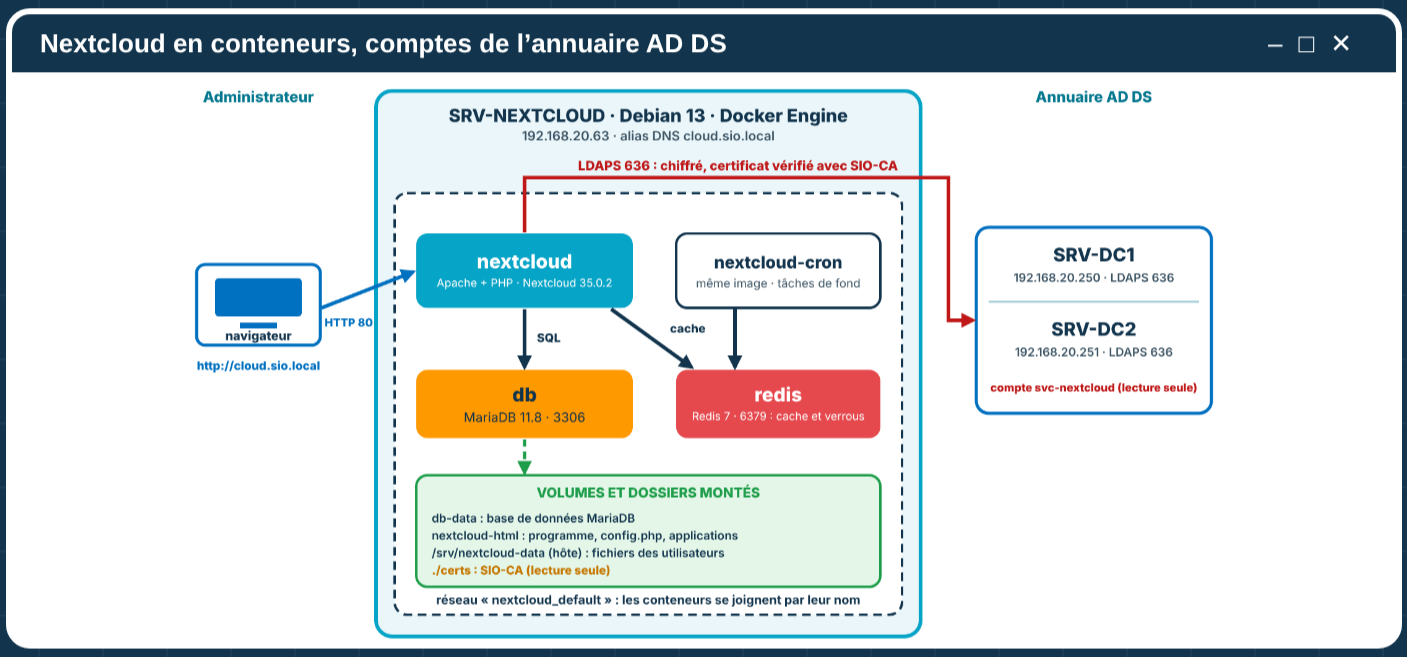
TUTORIEL

DOCKER COMPOSE · NEXTCLOUD 35

Installer avec Docker Compose
NEXTCLOUD



NIVEAU
Intermédiaire



1. Préparer le serveur Debian 13 et le DNS
2. Installer Docker Engine et Docker Compose
3. Écrire le projet Compose de Nextcloud
4. Démarrer la pile et optimiser Nextcloud
5. Préparer AD DS : compte de service et autorité de certification
6. Lier Nextcloud à l'annuaire et se connecter

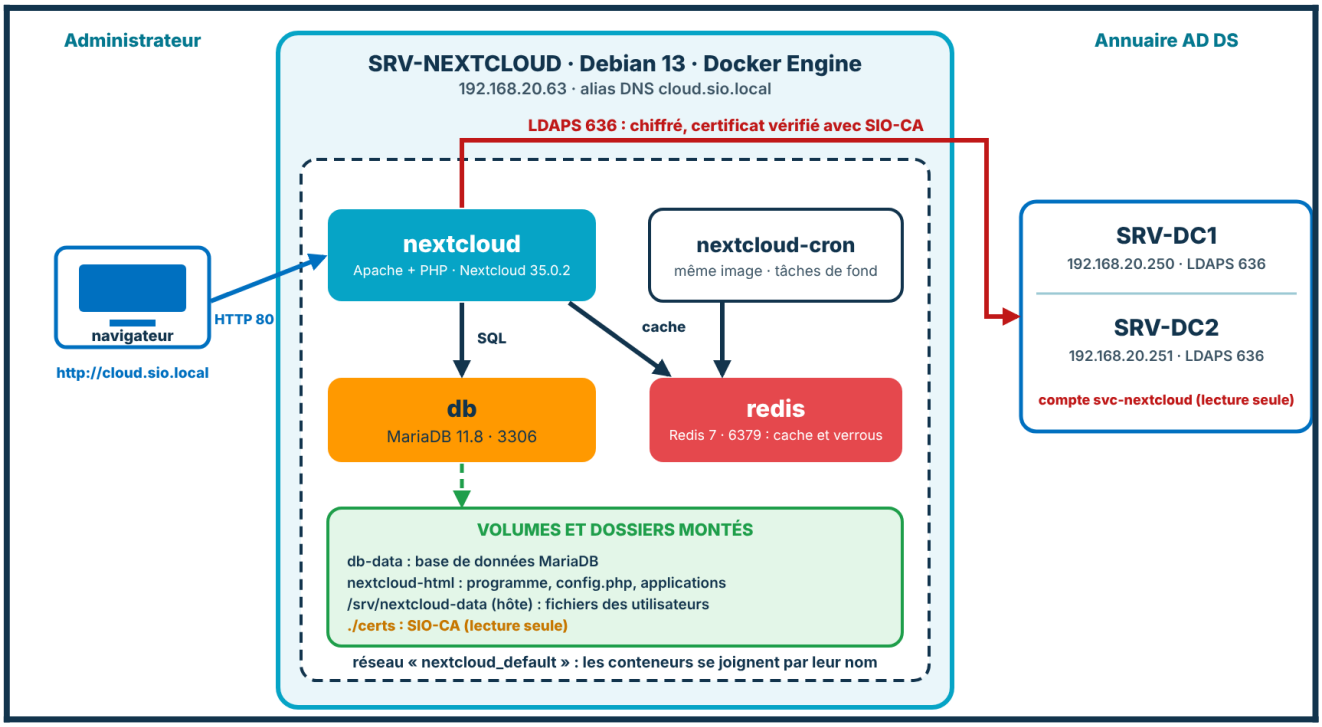
CC BY-NC
Utilisation commerciale interdite
© PortailSIO – 11/2026

Ce tutoriel en ligne
Scannez le QR code. Tous les tutoriels :
portail.sio-ndlp.fr/tutoriels.php

Ce tutoriel a été réalisé avec l'hyperviseur © Proxmox VE. Il peut être réalisé dans un environnement de virtualisation personnel à l'aide des logiciels © VirtualBox ou © VMware Player. Il **suppose que vous disposez d'une machine virtuelle Debian 13** (installation minimale avec serveur SSH) et du domaine **sio.local** construit dans les tutoriels précédents (contrôleurs **SRV-DC1** et **SRV-DC2** sous Windows Server 2025, avec l'autorité de certification **SIO-CA** et **LDAPS** activé : voir le tutoriel « Activer LDAPS sur AD DS »).

Nextcloud est une plateforme libre de **stockage et de partage de fichiers en ligne** : un « cloud » que l'on héberge soi-même, avec ses comptes, ses dossiers partagés, son agenda et ses contacts, accessible depuis un navigateur, un poste Windows ou un téléphone. La version **35.0.2** est publiée sous forme d'**image officielle** pour Docker.

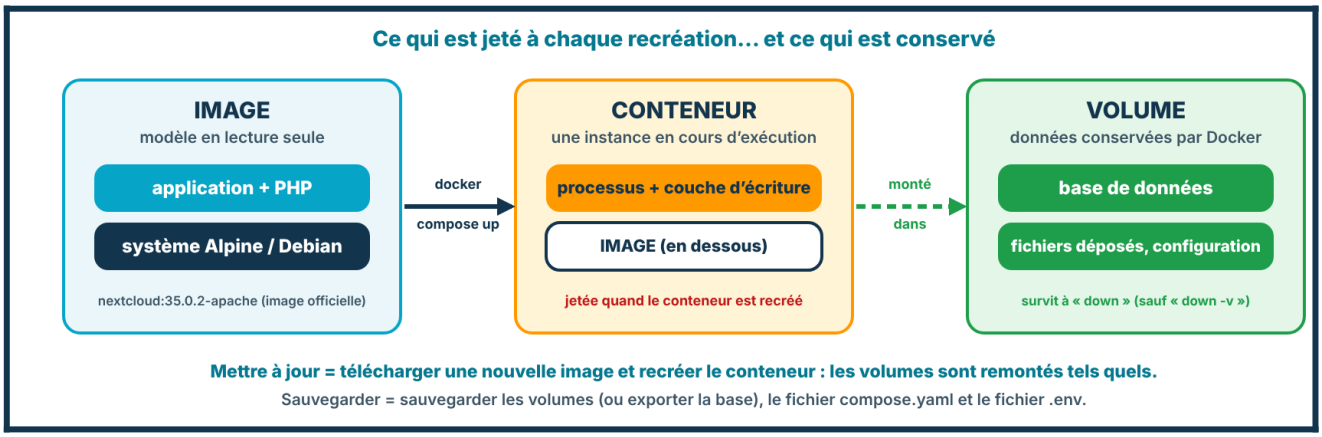
Dans ce tutoriel, nous installons Nextcloud **avec Docker Compose** : le site, la base de données, le cache et les tâches de fond s'exécutent chacun dans leur **conteneur**, et toute l'installation se décrit dans **deux fichiers** (`compose.yaml` et `.env`). L'image officielle de Nextcloud se configure **toute seule au premier démarrage** à partir de variables d'environnement : pas d'assistant web, pas de commande d'installation. Nous lions ensuite Nextcloud à l'annuaire **AD DS** pour que les utilisateurs se connectent avec leur compte Windows. Voici l'architecture obtenue :



Les quatre conteneurs

- **nextcloud** : Apache, PHP et le programme Nextcloud 35.0.2 ; il publie le port **80** du serveur
- **nextcloud-cron** : la **même image**, qui exécute les tâches de fond (nettoyages, notifications, indexation…) toutes les 5 minutes ; elle remplace la tâche `cron` du système
- **db** : MariaDB 11.8, qui stocke les comptes, les partages et l'index des fichiers ; son port 3306 reste **invisible** du réseau
- **redis** : le cache partagé, qui accélère Nextcloud et évite que deux personnes modifient le même fichier en même temps ; il n'est lui non plus joignable que par les autres conteneurs
- **AD DS** (SRV-DC1 et son réplicat SRV-DC2) authentifie les utilisateurs par une connexion **chiffrée LDAPS**, avec un compte de service en lecture seule ; le certificat de l'autorité **SIO-CA** est fourni aux conteneurs par un dossier monté en lecture seule

Une **image** est le modèle en lecture seule d'une application ; un **conteneur** en est une instance en cours d'exécution ; un **volume** conserve les données en dehors du conteneur. C'est ce qui permet de supprimer et de recréer le conteneur (pour une mise à jour, par exemple) sans rien perdre :



Dans le cadre de ce tutoriel, nous utilisons les machines suivantes (VLAN SERVEURS, réseau 192.168.20.0 /24) :

Machine	Rôle	Système	Adresse IP
SRV-NEXTCLOUD	Serveur Docker exécutant Nextcloud (nom `srv-nextcloud`, alias DNS `cloud.sio.local`)	Debian 13	192.168.20.63
SRV-DC1	Contrôleur de domaine, DNS, autorité de certification SIO-CA	Windows Server 2025	192.168.20.250
SRV-DC2	Contrôleur de domaine, DNS, annuaire de secours	Windows Server 2025	192.168.20.251
Poste admin	Navigateur web pour utiliser Nextcloud (DNS = les deux contrôleurs)	Windows 11	DHCP

Pour **SRV-NEXTCLOUD**, prévoyez au minimum **2 vCPU, 4 Go de mémoire et 30 Go de disque** (davantage si les utilisateurs doivent y déposer beaucoup de fichiers : le dossier de données grandit avec eux). Ces valeurs, comme les adresses et les mots de passe d'exemple, sont à adapter à votre environnement. Les commandes Debian sont exécutées avec un compte autorisé à utiliser `sudo` (ou directement en `root`, sans le préfixe `sudo`) ; les commandes Windows, dans **PowerShell** ouvert **en tant qu'administrateur** sur **SRV-DC1**.

01 ÉTAPE 1 SUR 6 — Préparer le serveur Debian 13 et le DNS

Docker n'a besoin que d'un système à jour, d'un nom, d'une adresse fixe et de l'heure exacte. On déclare aussi le nom **cloud.sio.local** dans le DNS du domaine. Sur **SRV-NEXTCLOUD** :

1. Mettez le système à jour : `sudo apt update && sudo apt full-upgrade -y`
2. Donnez son nom au serveur et déclarez-le dans le fichier `hosts` (nom complet et nom court) :

TERMINAL

```
sudo hostnamectl set-hostname srv-nextcloud
echo "192.168.20.63  cloud.sio.local  srv-nextcloud" | sudo tee -a /etc/hosts
```

3. Réglez le fuseau horaire : `sudo timedatectl set-timezone Europe/Paris`

4 Donnez au serveur une **adresse IP fixe** (adaptez le nom de la carte, ici `ens18`, avec `ip -br address`) : éditez `/etc/network/interfaces` puis redémarrez le réseau avec `sudo systemctl restart networking` :

```
/etc/network/interfaces
auto ens18
iface ens18 inet static
    address 192.168.20.63/24
    gateway 192.168.20.254
```

5 Indiquez les **DNS du domaine** dans `/etc/resolv.conf` : les conteneurs s'en servent pour retrouver `srv-dc1.sio.local` et `srv-dc2.sio.local` :

```
/etc/resolv.conf
search sio.local
nameserver 192.168.20.250
nameserver 192.168.20.251
```

6 Sur **SRV-DC1**, déclarez l'alias DNS de Nextcloud (l'enregistrement est répliqué sur SRV-DC2 avec la zone) :

```
POWERSHELL
Add-DnsServerResourceRecordA -Name "cloud" -ZoneName "sio.local" `
    -IPv4Address "192.168.20.63"
```

7 Vérifiez le résultat : nom, adresse IP, résolution DNS des contrôleurs et de Nextcloud, heure :

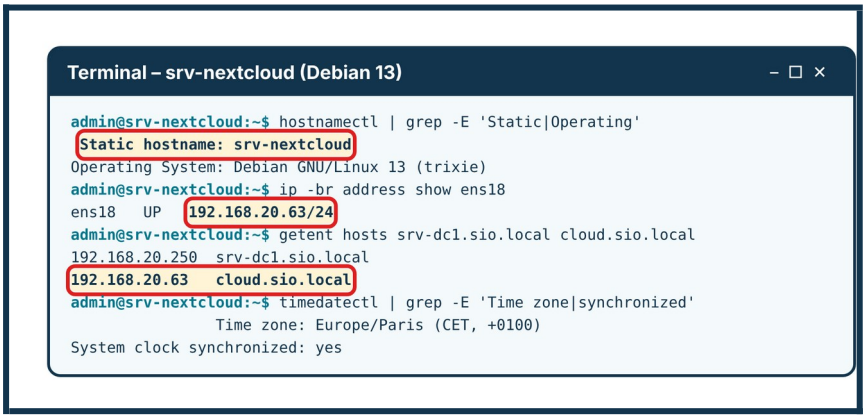


Figure 1 – Contrôle de la préparation du serveur

À SAVOIR

Si « System clock synchronized » indique « no », installez un service de temps (`sudo apt install -y chrony`). L'heure doit être juste : le certificat LDAPS de l'étape 5 a des dates de validité.

✔ **POINT DE CONTRÔLE**

- `hostnamectl` affiche le nom **srv-nextcloud** et Debian GNU/Linux **13**
- L'adresse IP **192.168.20.63** est fixe
- `getent hosts srv-dc1.sio.local cloud.sio.local` renvoie les deux adresses
- Le fuseau horaire est **Europe/Paris** et l'horloge est **synchronisée**

02 ÉTAPE 2 SUR 6 Installer Docker Engine et Docker Compose

Docker Engine est le moteur qui exécute les conteneurs ; **Docker Compose** est son extension qui décrit toute une application (plusieurs conteneurs, un réseau, des volumes) dans **un seul fichier** et la pilote avec une seule commande. Docker s'installe depuis le **dépôt officiel de Docker** : les paquets de Debian sont plus anciens et n'incluent pas Compose. Sur le serveur :

A – Ajouter le dépôt officiel de Docker

1. Retirez d'éventuels paquets concurrents (sur une installation neuve, il n'y a rien à retirer) puis installez les outils de téléchargement :

```
TERMINAL
sudo apt remove $(dpkg --get-selections docker.io docker-compose docker-doc \
    podman-docker containerd runc 2>/dev/null | cut -f1)
sudo apt update && sudo apt install -y ca-certificates curl
```

2. Enregistrez la clé de signature du dépôt :

```
TERMINAL
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg \
    -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

3. Déclarez le dépôt (format « deb822 »). Le nom de la version de Debian, `trixie`, est lu automatiquement dans le système :

```
TERMINAL
sudo tee /etc/apt/sources.list.d/docker.sources > /dev/null <<EOF
Types: deb
URIs: https://download.docker.com/linux/debian
Suites: $(. /etc/os-release && echo "$VERSION_CODENAME")
Components: stable
Signed-By: /etc/apt/keyrings/docker.asc
EOF
sudo apt update
```

B – Installer Docker Engine et le plugin Compose

1. Installez les cinq paquets de Docker :

```
TERMINAL
sudo apt install -y docker-ce docker-ce-cli containerd.io \
    docker-buildx-plugin docker-compose-plugin
```

Paquet	Rôle
docker-ce	Le moteur (service `docker`) qui crée et exécute les conteneurs
docker-ce-cli	La commande `docker`
containerd.io	Le composant de bas niveau qui exécute réellement les conteneurs
docker-buildx-plugin	L'outil de construction d'images (`docker build`)
docker-compose-plugin	**Docker Compose** : la commande `docker compose`

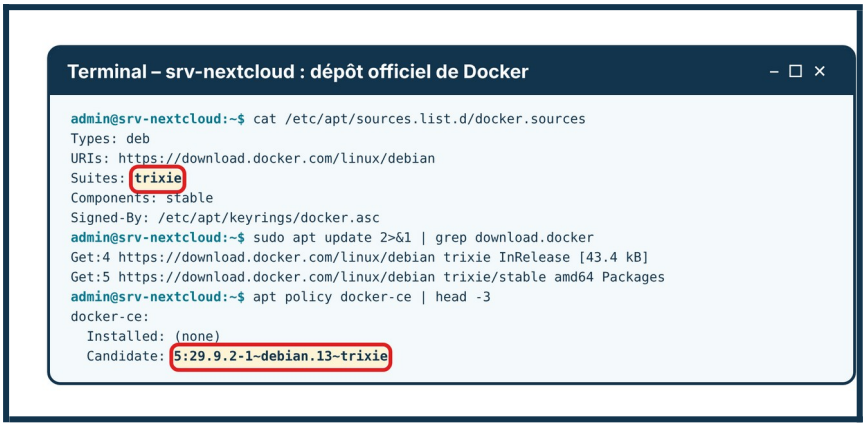


Figure 2 – Dépôt officiel de Docker et version candidate

À SAVOIR

Si la ligne « Candidate » indique une version venant de Debian (`docker.io`), le dépôt n'est pas pris en compte : relisez le fichier `docker.sources` (indentation, nom de la version).

C – Vérifier que Docker fonctionne

1 Le service démarre tout seul à l'installation. Contrôlez-le, affichez les versions, puis lancez le conteneur de test officiel :

TERMINAL

```
sudo systemctl is-active docker containerd
sudo docker --version && sudo docker compose version
sudo docker run --rm hello-world
```

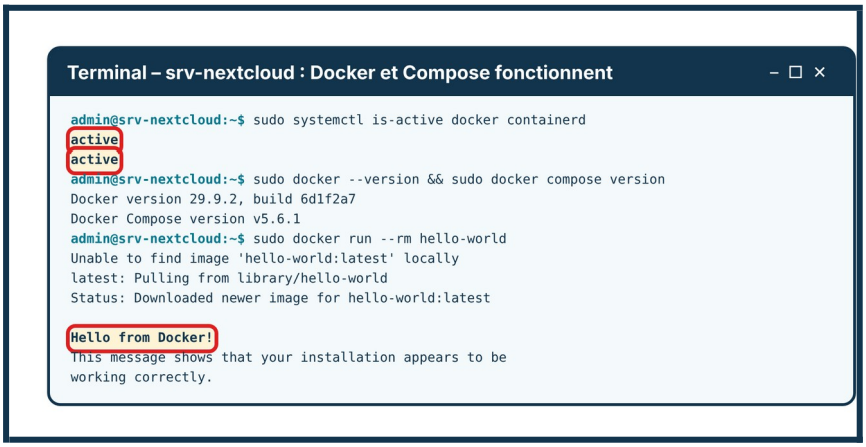


Figure 3 – Docker et Compose en état de marche

À SAVOIR

Le message « Hello from Docker! » prouve que le moteur sait télécharger une image sur Internet et exécuter un conteneur. L'option `--rm` supprime ensuite ce conteneur de test.

À SAVOIR

Les commandes de ce tutoriel utilisent `sudo docker`. Ajouter votre compte au groupe `docker` permettrait de s'en passer, mais ce groupe équivaut à des droits **root** sur le serveur : on l'évite en production. Autre point d'attention : les ports **publiés** par Docker (`ports:`) ne sont pas filtrés par un pare-feu configuré sur l'hôte (ufw, firewalld) ; limitez au besoin leur accès par une règle réseau en amont.

✔ **POINT DE CONTRÔLE**

- Le dépôt `download.docker.com` apparaît dans `apt update` et `apt policy docker-ce` propose une version **5:29** (ou supérieure)
- `systemctl is-active docker` répond **active**
- `docker compose version` affiche une version de **Docker Compose**
- `docker run --rm hello-world` affiche « **Hello from Docker!** »

ÉTAPE 3 SUR 6

03 Écrire le projet Compose de Nextcloud

Un **projet Compose** est un dossier qui contient le fichier `compose.yaml` (les conteneurs, leurs réglages, leurs volumes) et, à côté, le fichier `.env` (les valeurs qui changent d'un serveur à l'autre : versions, mots de passe). Séparer les deux permet de **partager** le premier sans divulguer les secrets du second.

A – Créer les dossiers et le fichier d'environnement

1. Créez le dossier du projet (le sous-dossier `certs` recevra le certificat de SIO-CA à l'étape 5) et le **dossier des fichiers des utilisateurs**, hors du projet. Le compte `www-data` du conteneur a l'identifiant **33**, le même que sur Debian : il doit être propriétaire du dossier de données :

TERMINAL

```
sudo mkdir -p /opt/nextcloud/certs /srv/nextcloud-data
sudo chown 33:33 /srv/nextcloud-data
sudo chmod 750 /srv/nextcloud-data
cd /opt/nextcloud
```

2. Créez le fichier `.env` avec `sudo nano /opt/nextcloud/.env` :

/opt/nextcloud/.env

```
NEXTCLOUD_TAG=35.0.2-apache
MARIADB_TAG=11.8
REDIS_TAG=7-alpine
TZ=Europe/Paris
DB_PASSWORD=Nxc_SIO_2026
DB_ROOT_PASSWORD=Root_SIO_2026
ADMIN_PASSWORD=Admin_SIO_2026!
```

3. Ce fichier contient des mots de passe : réservez-le à `root` :

TERMINAL

```
sudo chmod 600 /opt/nextcloud/.env
```

REMARQUE IMPORTANTE

`Nxc_SIO_2026`, `Root_SIO_2026` et `Admin_SIO_2026!` ne sont que des exemples : **choisissez trois mots de passe solides et différents**. Attention : MariaDB et Nextcloud ne lisent ces valeurs **qu'au premier démarrage** (création de la base, du compte administrateur) ; les modifier ensuite dans `.env` ne change pas les mots de passe déjà enregistrés (voir le dépannage).

Variable	Rôle
`NEXTCLOUD_TAG`	Version de l'image Nextcloud. `35.0.2-apache` fige la version (Apache et PHP sont inclus dans l'image) ; `35-apache` suivrait les mises à jour de la branche 35
`MARIADB_TAG`, `REDIS_TAG`	Versions de MariaDB (11.8, prise en charge par Nextcloud 35) et de Redis
`TZ`	Fuseau horaire appliqué aux conteneurs
`DB_PASSWORD`, `DB_ROOT_PASSWORD`	Mot de passe de l'utilisateur `nextcloud` de la base et mot de passe de l'administrateur MariaDB
`ADMIN_PASSWORD`	Mot de passe du compte **admin** de Nextcloud (compte local, absent de l'annuaire AD DS)

B – Écrire le fichier compose.yaml

Le fichier est écrit en **YAML** : l'**indentation** (deux espaces, jamais de tabulation) donne la structure. Un **bloc d'ancrage** (`x-nextcloud`) rassemble ce que partagent les deux conteneurs Nextcloud : l'image, l'environnement et les volumes. Créez `sudo nano /opt/nextcloud/compose.yaml` et commencez par ce bloc :

/opt/nextcloud/compose.yaml (1/3)

```
name: nextcloud

x-nextcloud: &nextcloud
  image: nextcloud:${NEXTCLOUD_TAG}
  restart: unless-stopped
  environment:
    TZ: ${TZ}
    MYSQL_HOST: db
    MYSQL_DATABASE: nextcloud
    MYSQL_USER: nextcloud
    MYSQL_PASSWORD: ${DB_PASSWORD}
    REDIS_HOST: redis
    NEXTCLOUD_ADMIN_USER: admin
    NEXTCLOUD_ADMIN_PASSWORD: ${ADMIN_PASSWORD}
    NEXTCLOUD_TRUSTED_DOMAINS: cloud.sio.local
    OVERWRITECLIURL: http://cloud.sio.local
    PHP_MEMORY_LIMIT: 512M
    PHP_UPLOAD_LIMIT: 1G
    LDAPTLS_CACERT: /certs/sio-ca.crt
  volumes:
    - nextcloud-html:/var/www/html
    - /srv/nextcloud-data:/var/www/html/data
    - ./certs:/certs:ro
```

Ajoutez à la suite la section `services:` avec la **base de données** et le **cache Redis**. MariaDB demande ici trois réglages que recommande Nextcloud (niveau d'isolation et journal binaire) :

/opt/nextcloud/compose.yaml (2/3)

```
services:
  db:
    image: mariadb:${MARIADB_TAG}
    restart: unless-stopped
    command:
      - --transaction-isolation=READ-COMMITTED
      - --log-bin=binlog
      - --binlog-format=ROW
    environment:
      MARIADB_DATABASE: nextcloud
      MARIADB_USER: nextcloud
      MARIADB_PASSWORD: ${DB_PASSWORD}
      MARIADB_ROOT_PASSWORD: ${DB_ROOT_PASSWORD}
      TZ: ${TZ}
    volumes:
      - db-data:/var/lib/mysql
    healthcheck:
      test: ["CMD", "healthcheck.sh", "--connect", "--innodb_initialized"]
      interval: 10s
      timeout: 5s
      retries: 12
      start_period: 20s

  redis:
    image: redis:${REDIS_TAG}
    restart: unless-stopped
    healthcheck:
      test: ["CMD", "redis-cli", "ping"]
      interval: 10s
      timeout: 3s
      retries: 5
```

Puis, à la suite dans le même fichier (au même niveau que `db`), les deux conteneurs Nextcloud et la déclaration des volumes :

/opt/nextcloud/compose.yaml (3/3)

```
  nextcloud:
    <<: *nextcloud
    depends_on:
      db:
        condition: service_healthy
      redis:
        condition: service_healthy
    ports:
      - "80:80"

  nextcloud-cron:
    <<: *nextcloud
    entrypoint: /cron.sh
    depends_on:
      - nextcloud

volumes:
  db-data:
  nextcloud-html:
```

Élément	Signification
`&nextcloud` et `<<: *nextcloud`	Ancre YAML : les deux conteneurs `nextcloud` et `nextcloud-cron` reprennent le même bloc (image, variables, volumes)
`MYSQL_*`, `NEXTCLOUD_ADMIN_*`	Variables lues par l'image au **premier démarrage** : elle crée les tables, le compte `admin` et le fichier `config.php`. Le serveur de la base est désigné par son **nom de service** (`db`), grâce au DNS interne de Docker
`NEXTCLOUD_TRUSTED_DOMAINS`, `OVERWRITECLIURL`	Nom sous lequel Nextcloud accepte d'être appelé et adresse utilisée dans ses liens
`REDIS_HOST`	Active Redis comme cache partagé et pour le verrouillage des fichiers ; APCu (cache local) est actif d'office
`nextcloud-html`	Volume nommé : le programme, `config/config.php` et les applications. Il survit à la recréation du conteneur
`/srv/nextcloud-data`	Dossier de l'hôte monté sur `/var/www/html/data` : les **fichiers des utilisateurs**, faciles à sauvegarder
`./certs:/certs:ro`	Dossier monté en **lecture seule** ; il contiendra `sio-ca.crt`, le certificat de l'autorité SIO-CA. La variable `LDAPTLS_CACERT` le désigne à la bibliothèque LDAP de PHP
`depends_on` + `healthcheck`	Nextcloud ne démarre qu'une fois MariaDB et Redis **prêts**
`ports: "80:80"`	Seul port publié : la base et Redis restent invisibles du réseau
`entrypoint: /cron.sh`	Le conteneur `nextcloud-cron` n'exécute pas Apache : il lance `cron.php` toutes les 5 minutes

REMARQUE IMPORTANTE

Dans un fichier YAML, une **erreur d'indentation** change le sens du fichier ou l'invalide. Évitez de copier-coller depuis un document qui remplace les espaces par des caractères spéciaux : la commande de validation du point suivant repère la plupart des erreurs.

C – Valider le projet et télécharger les images

1 Contrôlez la syntaxe, puis affichez les images qui seront utilisées (les variables du fichier `.env` doivent y être remplacées ; l'image Nextcloud apparaît deux fois, pour `nextcloud` et `nextcloud-cron`) :

TERMINAL

```
cd /opt/nextcloud
ls -A
ls -ld /srv/nextcloud-data
sudo docker compose config --quiet && echo "compose.yaml : syntaxe correcte"
sudo docker compose config --images
```

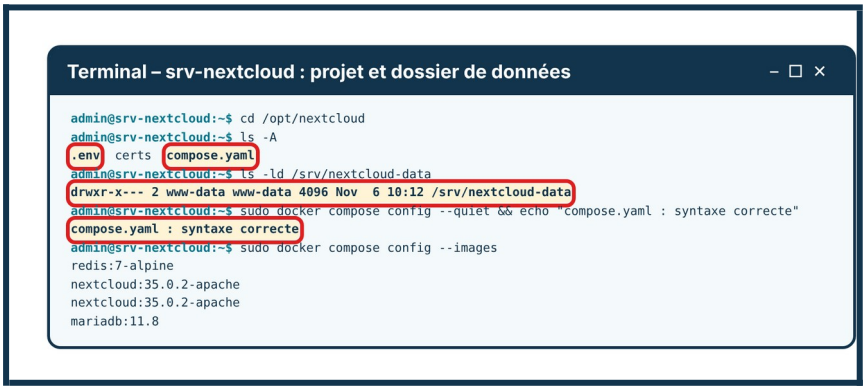


Figure 4 – Fichiers du projet et validation de la syntaxe

À SAVOIR

`docker compose config` affiche le fichier tel que Compose le comprend, variables remplacées : sans l'option `--quiet`, il montrerait aussi les mots de passe en clair. Une erreur indique la ligne en cause. Le dossier de données doit appartenir à `www-data`.

2 Téléchargez les images (environ 2 Go au total) et vérifiez-les :

TERMINAL

```
sudo docker compose pull
sudo docker images
```

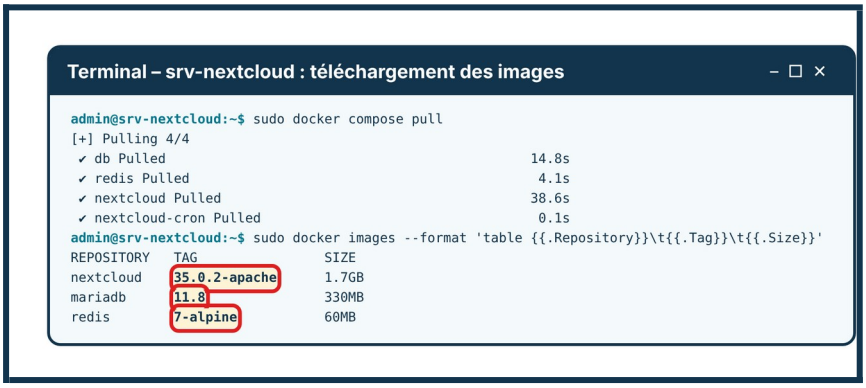


Figure 5 – Téléchargement des images

À SAVOIR

Le téléchargement est fait une seule fois : un démarrage ultérieur réutilise les images locales. Une erreur « pull access denied » ou « not found » signale une faute dans le nom ou la version de l'image (fichier `.env`).

✔ POINT DE CONTRÔLE

- `/opt/nextcloud/.env` n'est lisible que par **root** (droits 600) et contient des mots de passe personnalisés
- `/srv/nextcloud-data` appartient à **www-data** (identifiant 33)
- `docker compose config --quiet` ne renvoie **aucune erreur**
- `docker images` liste **nextcloud 35.0.2-apache**, **mariadb 11.8** et **redis 7-alpine**

04 ÉTAPE 4 SUR 6 Démarrer la pile et optimiser Nextcloud

A – Démarrer les conteneurs

1 Lancez toute la pile **en arrière-plan** (`-d`), puis affichez l'état des services :

TERMINAL

```
cd /opt/nextcloud
sudo docker compose up -d
sudo docker compose ps
```

Compose crée le réseau et les volumes, démarre **MariaDB et Redis en premier** ; quand ils sont « healthy », il démarre `nextcloud` puis `nextcloud-cron`. Le premier démarrage dure **une à deux minutes** : le conteneur copie le programme dans le volume, crée les tables de la base et le compte administrateur.

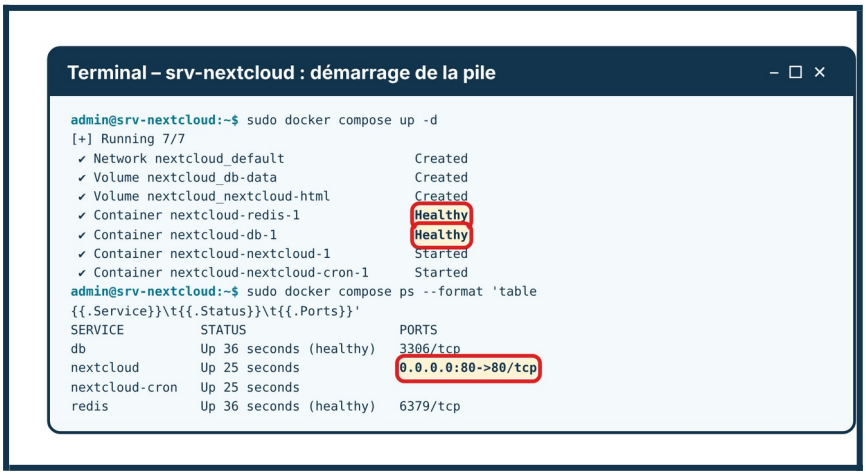


Figure 6 – Démarrage de la pile et état des services

À SAVOIR

Seul le port 80 est publié ; 3306 (MariaDB) et 6379 (Redis) ne sont accessibles qu'aux autres conteneurs. Si un conteneur indique « Restarting », lisez son journal (point suivant).

1 Suivez l'installation automatique dans le journal du conteneur : la ligne « **Nextcloud was successfully installed** » marque la fin. Quittez le mode suivi avec **Ctrl + C** :

TERMINAL

```
sudo docker compose logs -f --tail 10 nextcloud
```

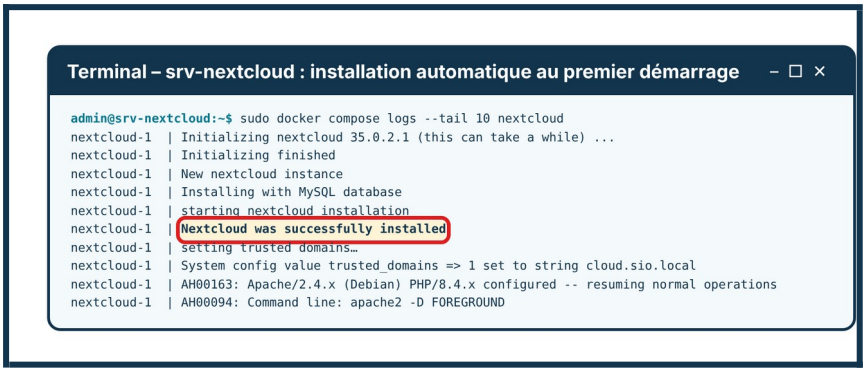


Figure 7 – Journal du conteneur nextcloud au premier démarrage

À SAVOIR

Les lignes affichées varient légèrement selon la version de l'image. Une erreur de connexion à la base ou un refus d'écriture dans le dossier de données se lit ici.

B – Première connexion

Depuis le **poste d'administration**, ouvrez un navigateur et saisissez `http://cloud.sio.local`. La page de connexion de Nextcloud s'affiche : saisissez **admin** et le mot de passe `ADMIN_PASSWORD` du fichier `.env`.

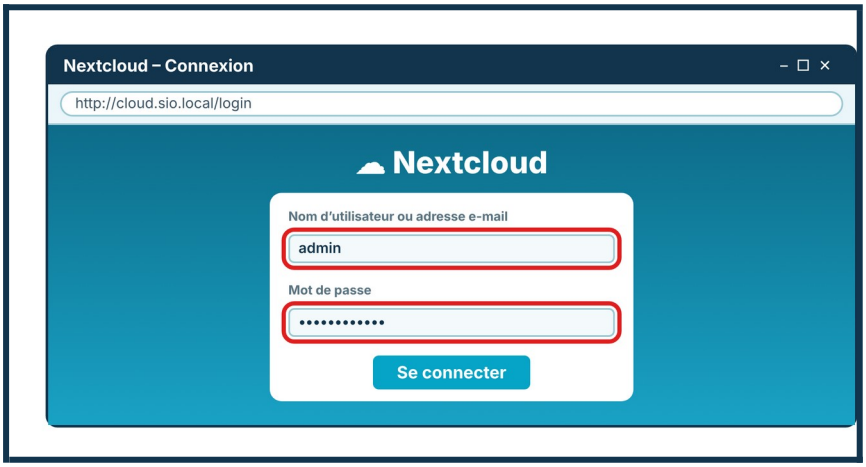


Figure 8 – Page de connexion de Nextcloud

À SAVOIR

Si le message « Accès via un domaine non approuvé » s'affiche, le nom saisi n'est pas dans `trusted_domains` : voir le point suivant. À la première connexion, Nextcloud peut proposer une fenêtre de bienvenue et d'applications recommandées : vous pouvez la fermer.

C – Disposer de la commande occ

occ est l'outil en ligne de commande de Nextcloud. Il s'exécute **dans le conteneur**, avec le compte `www-data` (sinon il crée des fichiers que le site ne peut plus lire). On crée sur le serveur un petit script qui s'en charge : toutes les commandes Nextcloud s'écriront alors `sudo occ …`, comme sur une installation native. Créez `sudo nano /usr/local/bin/occ` :

/usr/local/bin/occ

```
#!/bin/sh
exec docker compose -f /opt/nextcloud/compose.yaml exec -T --user www-data \
    nextcloud php occ "$@"
```

1. Rendez-le exécutable, vérifiez l'état de Nextcloud, déclarez le mode « cron » pour les tâches de fond (le conteneur `nextcloud-cron` les exécute) et contrôlez les domaines de confiance :

TERMINAL

```
sudo chmod +x /usr/local/bin/occ
sudo occ status
sudo occ background:cron
sudo occ config:system:get trusted_domains
```

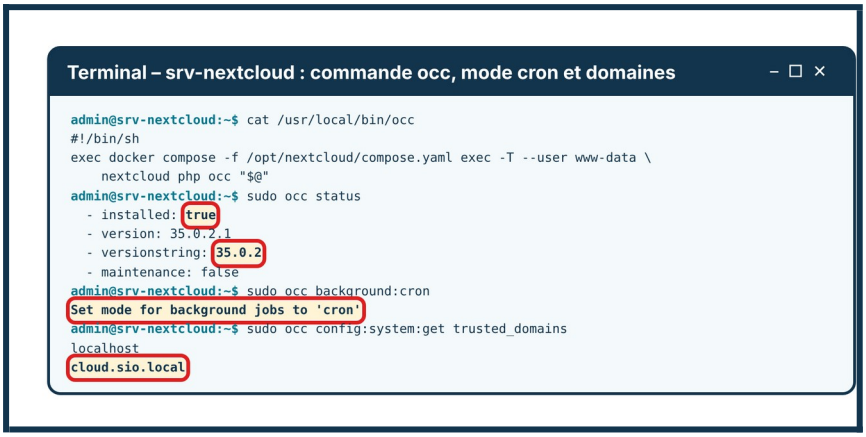


Figure 9 – Commande occ, mode cron et domaines de confiance

À SAVOIR

Le domaine `localhost` est ajouté d'office par l'image. Pour autoriser un autre nom plus tard (variables d'environnement ignorées après l'installation), utilisez `sudo occ config:system:set trusted_domains 2 --value=autre.sio.local`.

D – Appliquer les réglages de maintenance

Un petit ensemble de commandes complète l'installation : création des index manquants dans la base (pour des recherches rapides), réparations « coûteuses » qui ne sont pas lancées d'office, région téléphonique par défaut (France) et fenêtre de maintenance à 1 h du matin, heure UTC (les tâches lourdes s'exécutent la nuit) :

TERMINAL

```
sudo occ db:add-missing-indices
sudo occ maintenance:repair --include-expensive
sudo occ config:system:set default_phone_region --value=FR
sudo occ config:system:set maintenance_window_start --type=integer --value=1
```

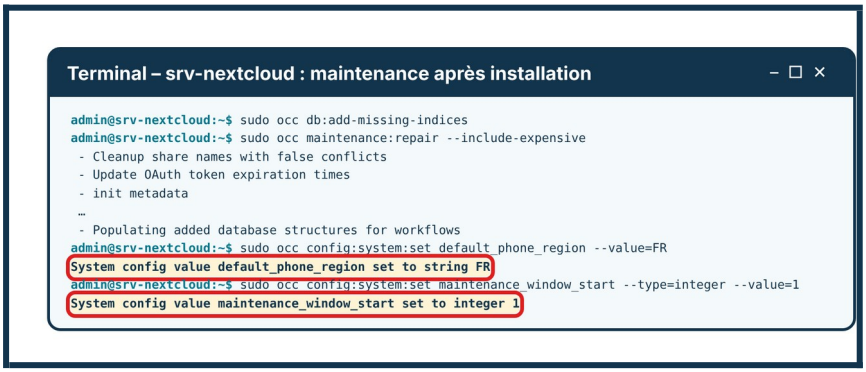


Figure 10 – Réglages de maintenance appliqués

À SAVOIR

La réparation affiche une liste d'opérations qui varie selon la version : l'important est qu'elle se termine sans message d'erreur.

E – Contrôler les caches APCu et Redis

L'image a déjà activé **APCu** (cache local) et **Redis** (cache partagé et verrouillage des fichiers : deux personnes ne peuvent pas écraser en même temps le même fichier) grâce à la variable `REDIS_HOST`. Il reste à le vérifier : Redis répond-il, et Nextcloud l'utilise-t-il ?

TERMINAL

```
sudo docker compose exec redis redis-cli ping
sudo occ config:system:get memcache.local
sudo occ config:system:get memcache.distributed
sudo occ config:system:get memcache.locking
sudo occ config:system:get redis host
```

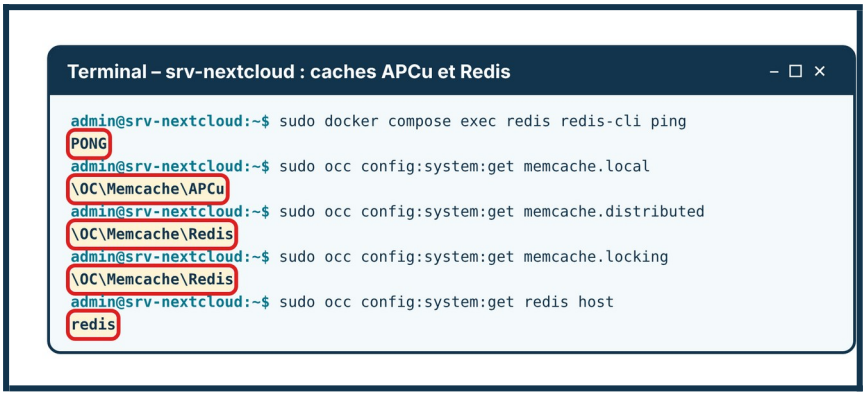


Figure 11 – Redis et les trois caches

À SAVOIR

Redis n'écoute que sur le réseau interne de Docker et n'a pas de mot de passe : c'est suffisant ici, puisque son port n'est pas publié. S'il devait un jour être joint depuis l'extérieur, protégez-le par un mot de passe et un pare-feu.

F – Lire la page « Vue d'ensemble »

Nextcloud contrôle lui-même sa configuration. Avec l'avatar en haut à droite, ouvrez « **Paramètres d'administration** » puis « **Vue d'ensemble** ». Attendez 5 minutes après le démarrage de `nextcloud-cron`, puis rechargez la page : la liste doit être presque entièrement verte.

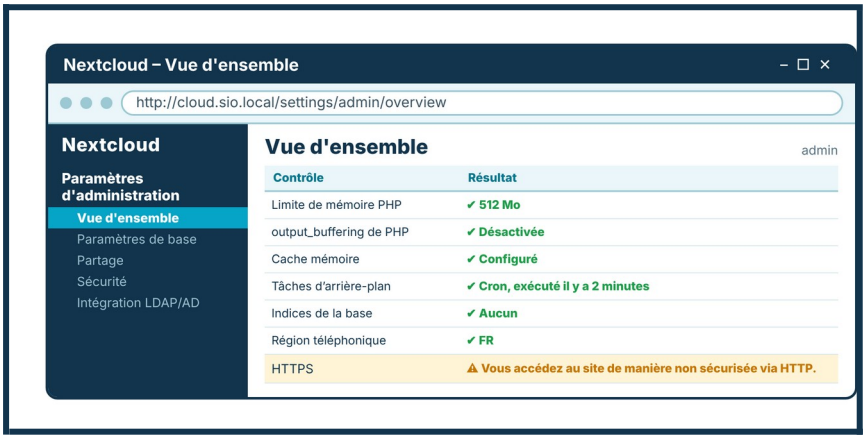


Figure 12 – Paramètres d'administration > Vue d'ensemble

À SAVOIR

Le seul avertissement attendu concerne **HTTPS** : le site est volontairement servi en HTTP dans ce tutoriel. Chaque ligne en erreur ou en avertissement indique sa cause et un lien vers la documentation ; relancez la page après chaque correction.

G – Sauvegarder et mettre à jour

Avec Docker, **quatre éléments** sont à sauvegarder : la **base de données**, le dossier des **fichiers des utilisateurs** (`/srv/nextcloud-data`), le volume `nextcloud-html` (avec `config.php` et ses secrets) et le dossier du projet (`compose.yaml` et `.env`). Les images, elles, se retéléchargent. Passez en maintenance pour que rien ne change pendant la copie :

TERMINAL

```
cd /opt/nextcloud
sudo occ maintenance:mode --on
sudo docker compose exec -T db sh -c 'mariadb-dump --single-transaction \
    -uroot -p"$MARIADB_ROOT_PASSWORD" nextcloud' | gzip \
    | sudo tee /var/backups/nextcloud-$(date +%F).sql.gz > /dev/null
sudo tar czf /var/backups/nextcloud-donnees-$(date +%F).tar.gz -C /srv nextcloud-data
sudo tar czf /var/backups/nextcloud-html-$(date +%F).tar.gz \
    -C /var/lib/docker/volumes nextcloud_nextcloud-html
sudo tar czf /var/backups/nextcloud-projet-$(date +%F).tar.gz -C /opt nextcloud
sudo occ maintenance:mode --off
```

Pour **mettre à jour** Nextcloud, sauvegardez d'abord, puis changez `NEXTCLOUD_TAG` dans `.env` et recréez les conteneurs : l'image détecte la nouvelle version et met la base à jour toute seule au démarrage. Avancez **une version majeure à la fois** (34 vers 35, puis 35 vers 36) : l'image refuse de sauter une version. Copiez les sauvegardes **hors du serveur** (partage réseau, serveur de sauvegarde).

TERMINAL

```
sudo nano /opt/nextcloud/.env      # NEXTCLOUD_TAG=<nouvelle version>-apache
cd /opt/nextcloud
sudo docker compose pull
sudo docker compose up -d
sudo occ status
```

À SAVOIR

En production, publiez Nextcloud en **HTTPS** (certificat délivré par l'autorité de certification de l'entreprise) : les identifiants et les fichiers des utilisateurs ne doivent pas circuler en clair sur le réseau. Le tutoriel « Installer Nextcloud derrière Traefik et le lier à AD DS » montre comment le faire avec un proxy inverse ; dans ce tutoriel, le site reste en HTTP pour se concentrer sur Docker et sur l'annuaire.

✔ **POINT DE CONTRÔLE**

- `docker compose ps` indique **4 services** démarrés (base et Redis « healthy »)
- La connexion en **admin** fonctionne et `sudo occ status` indique la version **35.0.2**
- `redis-cli ping` répond **PONG** et les trois caches sont enregistrés
- La « Vue d'ensemble » ne signale plus que l'avertissement **HTTPS**
- Une **sauvegarde** de la base, des fichiers, du volume et du projet existe

ÉTAPE 5 SUR 6

05 Préparer AD DS : compte de service et autorité de certification

Pour authentifier les utilisateurs, Nextcloud interroge l'annuaire avec un **compte de service** (lecture seule), puis vérifie le mot de passe de chaque utilisateur. Cette étape se déroule sur les **contrôleurs de domaine** et sur **SRV-NEXTCLOUD**.

REMARQUE IMPORTANTE

Windows Server 2025 impose la signature LDAP : les connexions LDAP simples (port 389, sans chiffrement) avec identifiant et mot de passe sont **refusées par défaut** sur un domaine neuf. Nextcloud doit donc utiliser **LDAPS** (LDAP chiffré, port **636**). Chaque contrôleur présente un certificat délivré par **SIO-CA** ; il suffit que **les conteneurs fassent confiance à cette autorité**.

À SAVOIR

Si vous avez déjà réalisé un tutoriel précédent du même domaine, les unités d'organisation, les utilisateurs **amartin** et **bdurand** existent peut-être déjà : les commandes de création répondent alors « existe déjà » (ignorez-le).

A – Créer le compte de service, les utilisateurs de test et les groupes

Sur **SRV-DC1**, dans PowerShell (**administrateur**), créez les unités d'organisation (OU), le compte **svc-nextcloud** et deux utilisateurs pour les tests. Si votre domaine possède déjà une OU pour vos utilisateurs, utilisez-la à la place de « Utilisateurs » (ce sera le **DN de base** de Nextcloud) :

POWERSHELL

```
$base = "DC=sio,DC=local"
New-ADOrganizationalUnit -Name "Comptes de service" -Path $base
New-ADOrganizationalUnit -Name "Utilisateurs" -Path $base
New-ADOrganizationalUnit -Name "Groupes" -Path $base
$mdp = Read-Host -AsSecureString "Mot de passe de svc-nextcloud"
New-ADUser -Name "svc-nextcloud" -SamAccountName "svc-nextcloud" `
    -UserPrincipalName "svc-nextcloud@sio.local" `
    -Path "OU=Comptes de service,$base" -AccountPassword $mdp `
    -Enabled $true -PasswordNeverExpires $true -CannotChangePassword $true
```

POWERSHELL

```
$mdp = Read-Host -AsSecureString "Mot de passe des utilisateurs de test"
New-ADUser -Name "Alice Martin" -GivenName Alice -Surname Martin `
    -SamAccountName amartin -UserPrincipalName amartin@sio.local `
    -EmailAddress amartin@sio.local -Path "OU=Utilisateurs,$base" `
    -AccountPassword $mdp -Enabled $true
New-ADUser -Name "Bruno Durand" -GivenName Bruno -Surname Durand `
    -SamAccountName bdurand -UserPrincipalName bdurand@sio.local `
    -EmailAddress bdurand@sio.local -Path "OU=Utilisateurs,$base" `
    -AccountPassword $mdp -Enabled $true
```

1 Nextcloud sait aussi lire les **groupes** de l'annuaire (pour partager un dossier avec tout un service, par exemple). Créez deux groupes et placez-y les utilisateurs :

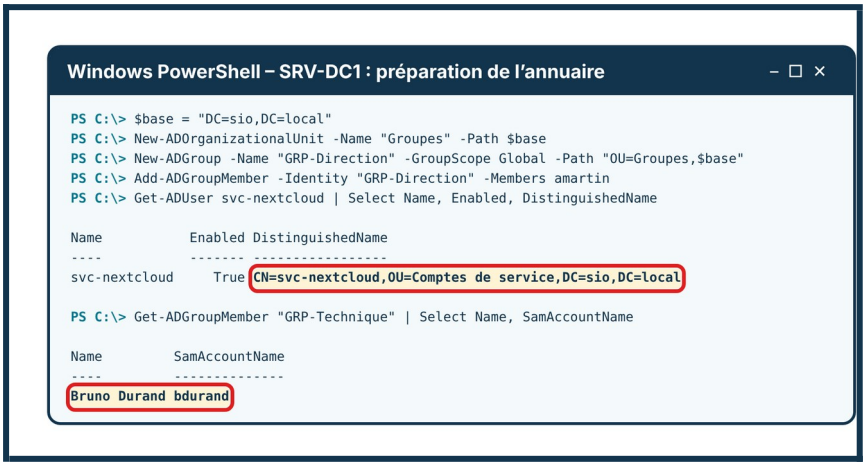
POWERSHELL

```
New-ADGroup -Name "GRP-Direction" -GroupScope Global -Path "OU=Groupes,$base"
New-ADGroup -Name "GRP-Technique" -GroupScope Global -Path "OU=Groupes,$base"
Add-ADGroupMember -Identity "GRP-Direction" -Members amartin
Add-ADGroupMember -Identity "GRP-Technique" -Members bdurand
```

2 Contrôlez la création :

POWERSHELL

```
Get-ADUser svc-nextcloud | Select Name, Enabled, DistinguishedName
Get-ADGroupMember "GRP-Technique" | Select Name, SamAccountName
```



À SAVOIR

Le compte **svc-nextcloud** est un compte de domaine ordinaire : il peut lire l'annuaire, rien d'autre. Ne l'ajoutez à aucun groupe d'administration. Son mot de passe n'expire pas, sinon Nextcloud cesserait de fonctionner le jour de l'expiration.

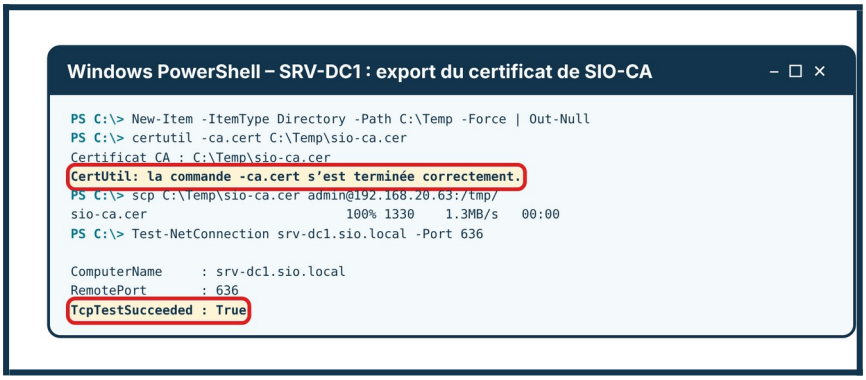
Figure 13 – PowerShell : groupes, compte de service et membres

B – Exporter le certificat de l'autorité SIO-CA

Les contrôleurs présentent un certificat signé par **SIO-CA** (tutoriel « Activer LDAPS sur AD DS »). Pour le vérifier, un client n'a besoin que du **certificat de l'autorité**, pas de ceux des contrôleurs : un seul fichier couvre SRV-DC1, SRV-DC2 et tout futur contrôleur. Sur **SRV-DC1**, exportez-le, copiez-le vers **SRV-NEXTCLOUD** et contrôlez que le port **636** répond :

POWERSHELL (SRV-DC1)

```
New-Item -ItemType Directory -Path C:\Temp -Force | Out-Null
certutil -ca.cert C:\Temp\sio-ca.cer
scp C:\Temp\sio-ca.cer admin@192.168.20.63:/tmp/
Test-NetConnection srv-dc1.sio.local -Port 636
```



À SAVOIR

Ce test prouve seulement que le port est ouvert. La vérification du certificat lui-même se fait depuis Debian (point suivant). Si vous préférez WinSCP ou un partage, seul compte le résultat : le fichier `sio-ca.cer` sur SRV-NEXTCLOUD.

Figure 14 – PowerShell : export du certificat de SIO-CA et test du port 636

C – Donner le certificat aux conteneurs et tester LDAPS

Sur **SRV-NEXTCLOUD**, convertissez le certificat au format **PEM** attendu par OpenSSL et placez-le dans le dossier `certs` du projet : il est monté en lecture seule dans les conteneurs sous `/certs/sio-ca.crt` (c'est le fichier désigné par `LDAPTLS_CACERT`). Vérifiez ensuite la chaîne **depuis le conteneur**, puis la recherche d'un utilisateur depuis l'hôte (le mot de passe de **svc-nextcloud** est demandé) :

TERMINAL

```
sudo openssl x509 -inform der -in /tmp/sio-ca.cer \
    -out /opt/nextcloud/certs/sio-ca.crt
cd /opt/nextcloud
sudo docker compose exec -T nextcloud openssl s_client \
    -connect srv-dc1.sio.local:636 -CAfile /certs/sio-ca.crt \
    -verify_return_error </dev/null 2>&1 | grep -E 'subject=|Verify'
sudo apt install -y ldap-utils
LDAPTLS_CACERT=/opt/nextcloud/certs/sio-ca.crt ldapsearch \
    -H ldaps://srv-dc1.sio.local -W \
    -D "CN=svc-nextcloud,OU=Comptes de service,DC=sio,DC=local" \
    -b "OU=Utilisateurs,DC=sio,DC=local" "(sAMAccountName=amartin)" dn mail
```

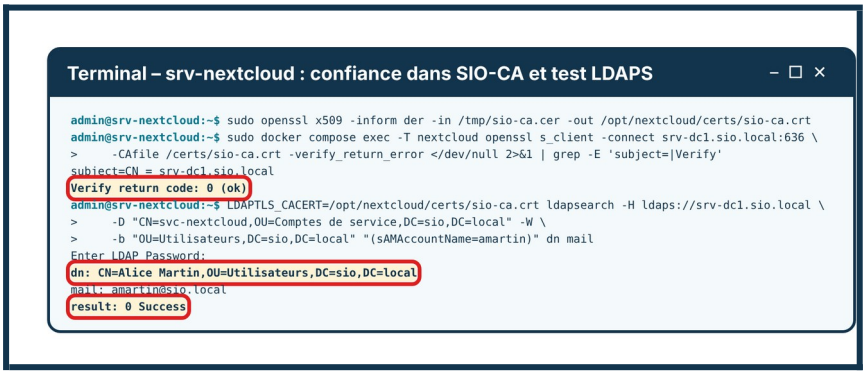


Figure 15 – Debian : confiance dans SIO-CA et recherche LDAPS

À SAVOIR

« Verify return code: 0 (ok) » prouve que le conteneur fait confiance au certificat. « result: 0 Success » et la ligne `dn:` prouvent que le compte de service, le DN de base et LDAPS fonctionnent. Répétez avec `srv-dc2.sio.local` : le même fichier suffit. Si `-inform der` échoue, le fichier est déjà au format PEM : copiez-le simplement.

À SAVOIR

Si vos contrôleurs utilisent des certificats **auto-signés**, il n'y a pas d'autorité : exportez les certificats de SRV-DC1 et de SRV-DC2, convertissez-les en PEM puis réunissez-les dans le même fichier (`cat srv-dc1.crt srv-dc2.crt | sudo tee /opt/nextcloud/certs/sio-ca.crt`). Le reste du tutoriel est identique.

REMARQUE IMPORTANTE

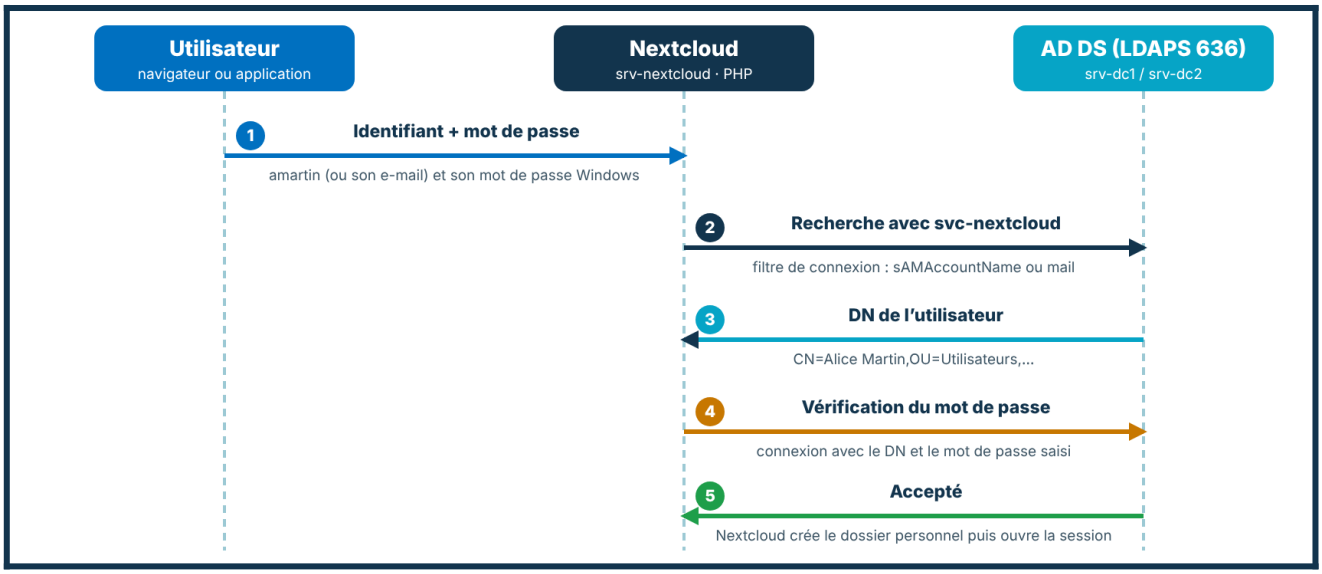
Après avoir ajouté ou remplacé le fichier, **redémarrez le conteneur nextcloud** (`sudo docker compose restart nextcloud`) : PHP conserve en mémoire sa configuration TLS et ne relit pas toujours le fichier à chaud.

✔ **POINT DE CONTRÔLE**

- Les OU **Utilisateurs**, **Groupes** et **Comptes de service**, le compte **svc-nextcloud**, les utilisateurs **amartin** et **bdurand** et les deux groupes existent
- Le fichier `/opt/nextcloud/certs/sio-ca.crt` existe et « Verify return code: **0 (ok)** » s'affiche **depuis le conteneur** `nextcloud` pour `srv-dc1` et `srv-dc2`
- `ldapsearch` en **ldaps://** renvoie l'entrée de **Alice Martin** (« result: 0 Success »)

06 ÉTAPE 6 SUR 6
Lier Nextcloud à l'annuaire et se connecter

Tout est prêt : on active l'application LDAP de Nextcloud, on déclare l'annuaire dans son assistant, on teste, puis on se connecte avec un compte Windows. Voici ce qui se passe quand un utilisateur de l'annuaire se connecte :



A – Activer l'application « Intégration LDAP/AD »

L'application est livrée avec Nextcloud mais **désactivée**. Activez-la en une commande (ou dans le menu de l'avatar > « Applications » > « Applications désactivées ») :

TERMINAL

```
sudo occ app:enable user_ldap
sudo occ app:list | grep user_ldap
```

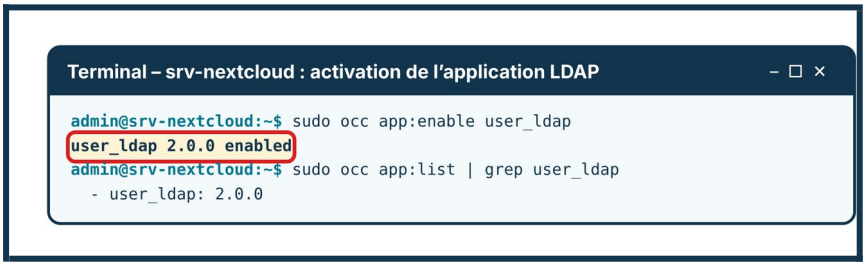


Figure 16 – Activation de l'application LDAP

À SAVOIR

Un nouveau menu « Intégration LDAP/AD » apparaît dans les paramètres d'administration. Déconnectez-vous puis reconnectez-vous en **admin** s'il n'est pas visible.

B – Déclarer l'annuaire AD DS dans l'assistant

Ouvrez « **Paramètres d'administration** » puis « **Intégration LDAP/AD** ». L'assistant se déroule en quatre onglets (**Serveur**, **Utilisateurs**, **Attributs de connexion**, **Groupes**) ; chaque champ est enregistré dès que vous le quittez, et le message en bas de page indique si la configuration est correcte (« Configuration OK »). Cliquez « **Continuer** » pour passer à l'onglet suivant.

Onglet Serveur — saisissez l'hôte en **ldaps://**, le port, le compte de service et le DN de base :

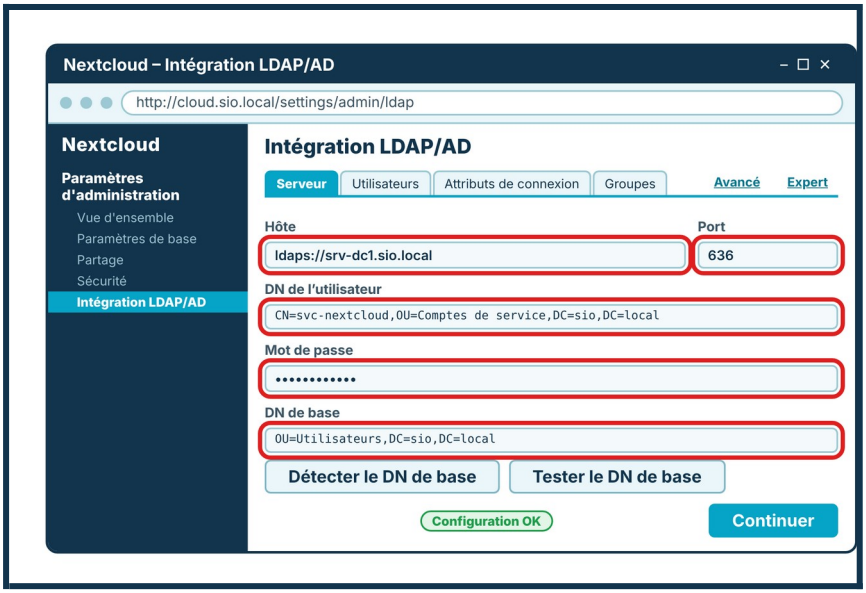


Figure 17 – Intégration LDAP/AD : onglet Serveur

À SAVOIR

Le DN de l'utilisateur est le DN **complet** du compte de service. Le bouton « Détecter le DN de base » propose la racine du domaine ; remplacez-la par l'OU des utilisateurs pour ne présenter à Nextcloud que les comptes voulus. « Tester le DN de base » doit répondre sans erreur.

Champ	Valeur à saisir	Remarque
Hôte	`ldaps://srv-dc1.sio.local`	Le préfixe **ldaps://** active le chiffrement ; utilisez le **nom complet** du contrôleur
Port	`636`	Port LDAPS (389 = LDAP non chiffré)
DN de l'utilisateur	`CN=svc-nextcloud,OU=Comptes de service,DC=sio,DC=local`	DN complet du compte de service
Mot de passe	mot de passe de **svc-nextcloud**	Stocké dans la configuration de Nextcloud
DN de base	`OU=Utilisateurs,DC=sio,DC=local`	Où chercher : seuls les comptes de cette OU sont visibles

Onglet Utilisateurs — Nextcloud propose les classes d'objets de l'annuaire (choisissez **user**). Cochez « **Modifier la requête LDAP** » et remplacez le filtre par celui-ci, qui ne retient que les **vrais comptes utilisateurs actifs** :

```
(&(objectClass=user)(objectCategory=person)(!
(userAccountControl:1.2.840.113556.1.4.803:=2)))
```

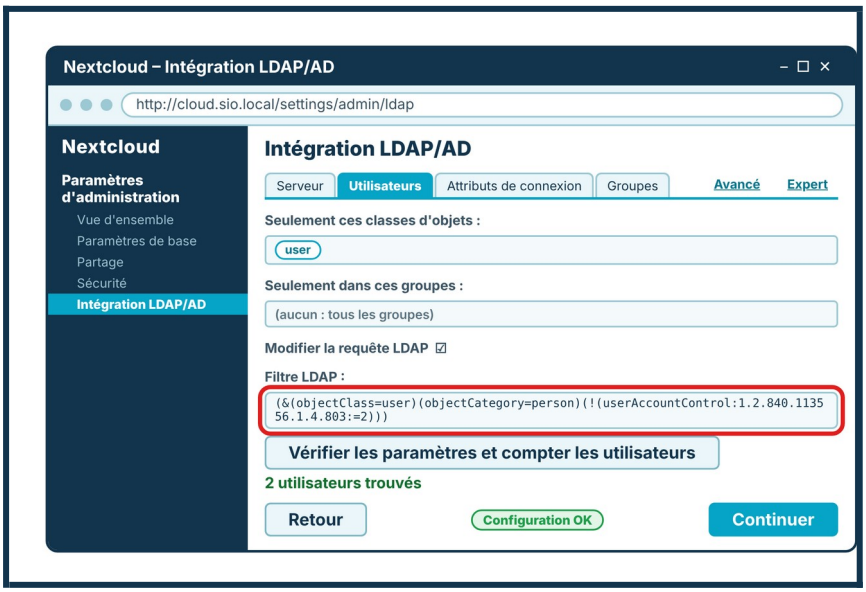


Figure 18 – Intégration LDAP/AD : onglet Utilisateurs

À SAVOIR

Le bit **2** de `userAccountControl` signifie « compte désactivé » : Nextcloud ne filtre pas lui-même les comptes désactivés d'Active Directory, ce filtre s'en charge. Cliquez « Vérifier les paramètres et compter les utilisateurs » : le nombre doit correspondre à votre annuaire (**2** ici).

Onglet Attributs de connexion — indiquez comment les utilisateurs saisissent leur identifiant : cochez « **Nom d'utilisateur LDAP** » (sAMAccountName, détecté automatiquement) et « **Adresse e-mail LDAP** » : ils pourront se connecter avec **amartin** ou avec **amartin@sio.local**. Pour tester, saisissez un identifiant dans « Nom de connexion à tester » :

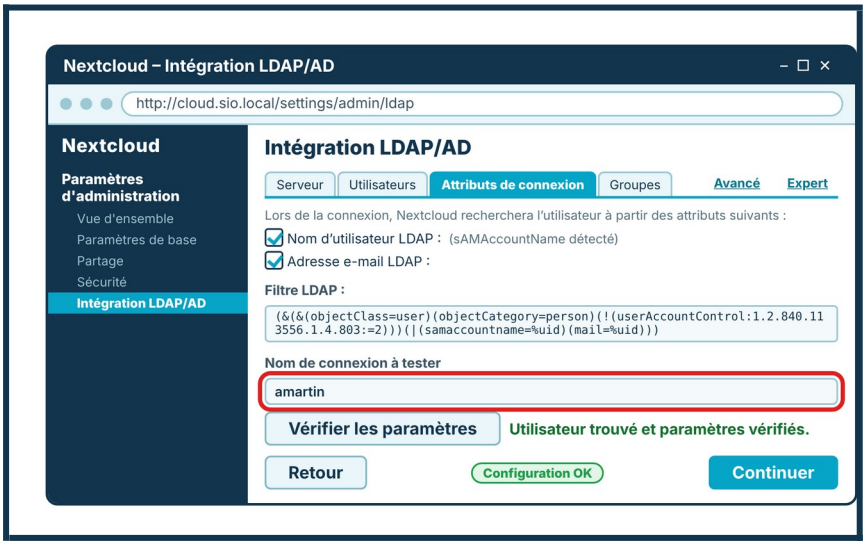


Figure 19 – Intégration LDAP/AD : onglet Attributs de connexion

À SAVOIR

Le filtre de connexion est construit à partir du filtre des utilisateurs : un compte désactivé ne peut donc pas ouvrir de session. Le message « Utilisateur trouvé et paramètres vérifiés » valide le tout.

Onglet Groupes — choisissez la classe **group**, puis sélectionnez dans « Seulement dans ces groupes » les deux groupes **GRP-Direction** et **GRP-Technique** (limiter les groupes évite d'importer tous les groupes de Windows, ce qui alourdirait Nextcloud) :

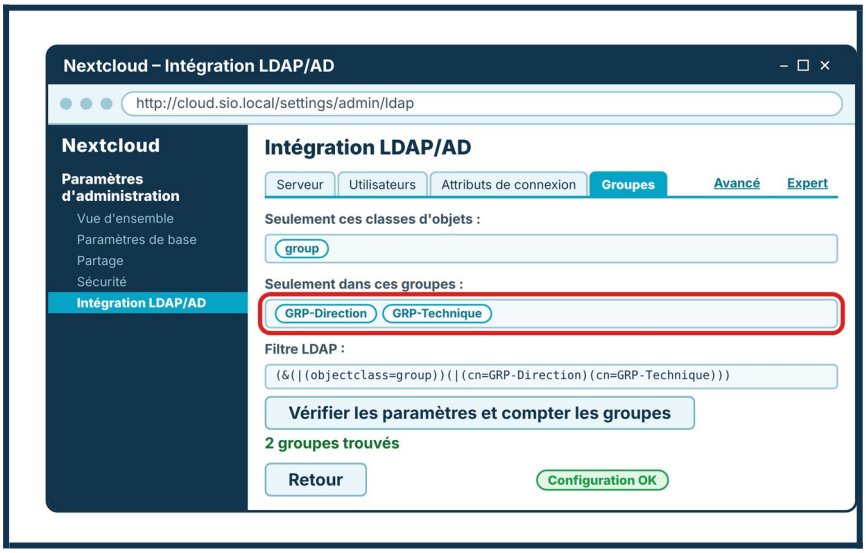


Figure 20 – Intégration LDAP/AD : onglet Groupes

À SAVOIR

Cliquez « Vérifier les paramètres et compter les groupes » : le message « 2 groupes trouvés » confirme la sélection. Les groupes de l'annuaire servent ensuite à partager un dossier ou à fixer des quotas.

C – Choisir le nom d'utilisateur interne et ajouter le second contrôleur

Avant la première connexion d'un utilisateur, ouvrez le lien « **Expert** » en haut de l'assistant. Par défaut, Nextcloud nomme chaque utilisateur d'après son identifiant unique (**objectGUID**) : un long nom illisible qui apparaîtrait dans les fichiers, les journaux et les liens de partage. Choisissez plutôt **samaccountname** :

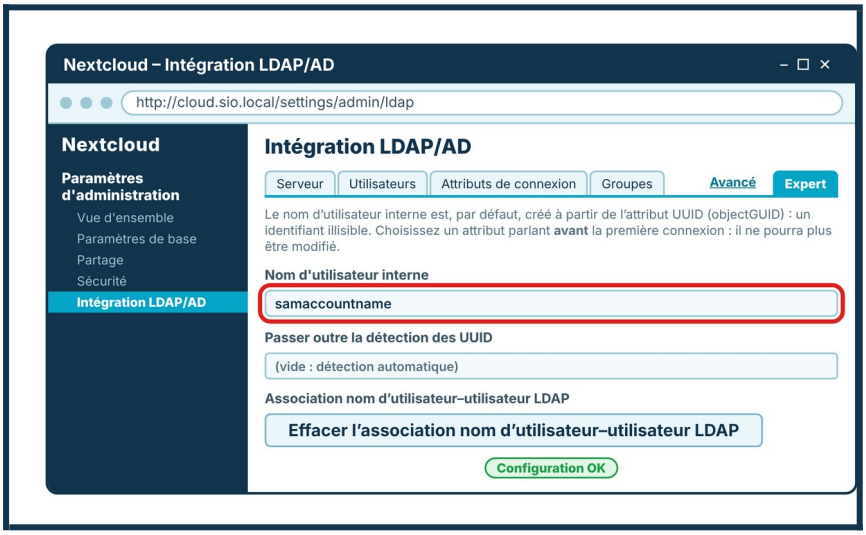


Figure 21 – Intégration LDAP/AD : onglet Expert

À SAVOIR

Ce réglage **ne doit plus changer** une fois des utilisateurs connectés : le bouton « Effacer l'association nom d'utilisateur–utilisateur LDAP » casse le lien avec les comptes et leurs données, ne l'utilisez que sur une installation vierge.

1. Pour la **tolérance de panne**, ajoutez SRV-DC2 comme **hôte de secours** : si SRV-DC1 ne répond plus, Nextcloud interroge SRV-DC2. L'assistant le propose dans l'onglet « **Avancé** » (rubrique « Paramètres de connexion », champs « Hôte de sauvegarde (réplique) » et « Port du serveur de sauvegarde ») ; la ligne de commande est équivalente et plus sûre :

TERMINAL

```
sudo occ ldap:set-config s01 ldapBackupHost ldaps://srv-dc2.sio.local
sudo occ ldap:set-config s01 ldapBackupPort 636
sudo occ ldap:set-config s01 ldapExpertUsernameAttr samaccountname
```

`s01` est le nom de la première configuration créée par l'assistant. La commande `sudo occ ldap:show-config s01` affiche toutes ses valeurs.

D – Tester l'annuaire en ligne de commande

Avant de se connecter, quatre commandes vérifient la configuration sans passer par le navigateur : le test global, la recherche d'un utilisateur, la recherche de groupes et l'état d'un compte :

```
TERMINAL
sudo occ ldap:test-config s01
sudo occ ldap:search "martin"
sudo occ ldap:search --group "GRP"
sudo occ ldap:check-user amartin
```

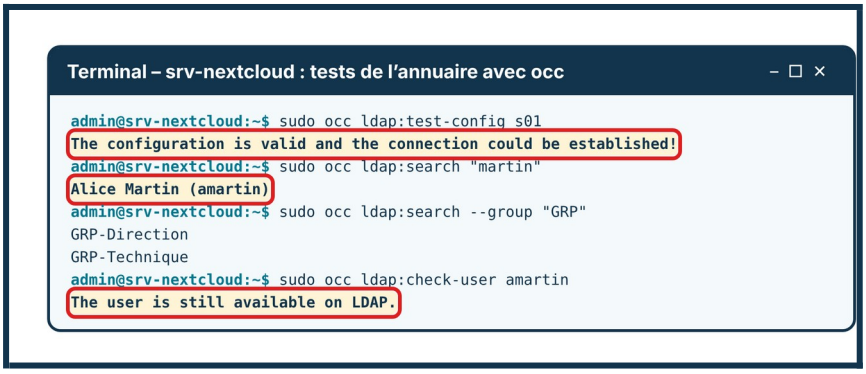


Figure 22 – Tests de l'annuaire avec occ

À SAVOIR

Si le test échoue, le message donne la cause : serveur injoignable (certificat, nom, port) ou identifiants refusés. Reportez-vous au tableau de dépannage en fin de tutoriel.

E – Se connecter avec un compte Windows

1. Connecté en **admin**, ouvrez le menu de l'avatar puis « **Comptes** » : les utilisateurs de l'annuaire y figurent, avec leurs groupes et la mention « jamais » en dernière connexion :



Figure 23 – Comptes : utilisateurs et groupes de l'annuaire

À SAVOIR

Ces comptes sont créés par l'annuaire : leur mot de passe se change dans Active Directory, pas dans Nextcloud. Le quota « Illimité » se règle par compte ou par groupe.

2 Déconnectez-vous de Nextcloud, puis ouvrez la page de connexion dans une **fenêtre de navigation privée** : saisissez **amartin** (ou son adresse e-mail) et son mot de passe de domaine. Le dossier personnel est créé à la première connexion :

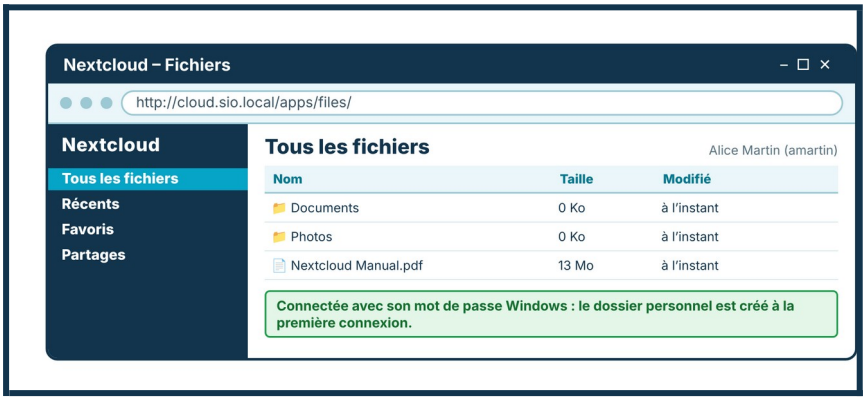


Figure 24 – Fichiers de l'utilisateur Alice Martin

À SAVOIR

Le mot de passe n'est jamais copié dans Nextcloud : à chaque connexion, Nextcloud le fait vérifier par l'annuaire. Désactiver le compte dans Active Directory suffit donc à lui retirer l'accès (ses fichiers restent sur le serveur, dans `/srv/nextcloud-data`).

3 **Pour aller plus loin** : `sudo occ ldap:show-remnants` liste les utilisateurs supprimés de l'annuaire dont les données sont encore présentes sur le serveur, et `sudo occ ldap:test-user-settings amartin` détaille ce que Nextcloud voit pour un utilisateur

✔ **POINT DE CONTRÔLE**

- La **Vue d'ensemble** est verte et l'application **Intégration LDAP/AD** est active
- `sudo occ ldap:test-config s01` répond que la configuration est **valide**
- **amartin** ouvre une session Nextcloud avec son **mot de passe Windows**, sous le nom `amartin`
- Si SRV-DC1 est arrêté, la connexion fonctionne encore grâce à **SRV-DC2**

Notre Nextcloud est **opérationnel** : il tourne dans quatre conteneurs sur Debian 13, avec ses fichiers sur le serveur, ses caches, ses tâches de fond planifiées, des sauvegardes possibles, et un lien LDAPS vers AD DS avec un second contrôleur en secours. Il reste à le publier en HTTPS et à organiser les dossiers partagés par groupe.

Dépannage : les symptômes les plus fréquents

Symptôme	Cause probable	Solution
« permission denied ... Docker daemon socket »	Commande `docker` lancée sans `sudo`	Utiliser `sudo docker …` (ou se connecter en root)
« Bind for 0.0.0.0:80 failed: port is already allocated »	Un autre service (Apache, nginx...) occupe déjà le port 80	`sudo ss -tlnp | grep ':80 '` puis arrêter ce service, ou publier un autre port (`"8080:80"`)
Un conteneur est en « Restarting »	Erreur de configuration ou base inaccessible	`sudo docker compose logs --tail 30 <service>` ; vérifier `.env` et l'indentation de `compose.yaml`
Page vide ou erreur 502 juste après `up -d`	Première initialisation encore en cours (1 à 2 minutes)	Suivre `sudo docker compose logs -f nextcloud` jusqu'à « Nextcloud was successfully installed »
« Cannot write into data directory »	Le dossier `/srv/nextcloud-data` n'appartient pas à l'identifiant 33	`sudo chown -R 33:33 /srv/nextcloud-data` puis `sudo docker compose restart nextcloud`
« Accès via un domaine non approuvé »	Le nom saisi n'est pas dans `trusted_domains`	`sudo occ config:system:set trusted_domains 2 --value=<nom>`
Mot de passe ou variable modifiés dans `.env` sans effet	L'image ne lit ces variables qu'au premier démarrage	Changer le mot de passe dans l'application ou la base (`sudo occ user:resetpassword admin`) ; `docker compose down -v` efface tout
`occ` : « no such service » ou « service is not running »	Le conteneur `nextcloud` est arrêté	`sudo docker compose ps` ; `sudo docker compose up -d`
Vue d'ensemble : tâches d'arrière-plan jamais exécutées	Conteneur `nextcloud-cron` arrêté ou mode cron non déclaré	`sudo docker compose ps` ; `sudo occ background:cron`
Test LDAP : « Can't contact LDAP server »	`sio-ca.crt` absent ou invalide, conteneur non redémarré, nom différent du certificat, port 636 bloqué	Refaire l'étape 5-C ; `docker compose restart nextcloud` ; utiliser le **nom complet** du contrôleur, pas son adresse IP
`ldapsearch` : « Strong(er) authentication required »	Connexion LDAP non chiffrée refusée par Windows Server 2025	Utiliser `ldaps://` et le port **636**, jamais `ldap://` avec mot de passe
« Invalid credentials » (code 49)	DN du compte ou mot de passe incorrect, compte désactivé	Saisir le **DN complet** du compte ; tester `ldapsearch -W` ; vérifier `Get-ADUser svc-nextcloud`
« 0 utilisateur trouvé »	DN de base incorrect ou filtre trop restrictif	Comparer avec `Get-ADUser -SearchBase` ; contrôler la casse et les virgules du DN de base
Les utilisateurs ont un nom illisible (suite de chiffres et de lettres)	Nom d'utilisateur interne laissé sur l'UUID avant la première connexion	Installation vierge : régler **samaccountname** (étape 6-C) puis effacer l'association ; sinon, conserver l'existant

Pense-bête : commandes et emplacements

Commande / emplacement	Utilité
`sudo docker compose ps`	État des services du projet et ports publiés
`sudo docker compose logs -f <service>`	Journal d'un conteneur en direct (Ctrl + C pour quitter)
`sudo docker compose up -d`	Crée ou met à jour les conteneurs d'après `compose.yaml`
`sudo docker compose restart nextcloud`	Redémarre Nextcloud sans le recréer (après un changement de certificat, par exemple)
`sudo docker compose down`	Supprime conteneurs et réseau (les volumes et `/srv/nextcloud-data` sont conservés ; `-v` efface les volumes !)
`sudo occ …`	Commande Nextcloud exécutée dans le conteneur, avec le compte `www-data`
`sudo occ maintenance:mode --on` / `--off`	Passer en maintenance avant une sauvegarde ou une mise à jour
`/opt/nextcloud/compose.yaml` **et** `.env`	Configuration du projet : à sauvegarder, `.env` reste confidentiel
`/srv/nextcloud-data/`	Fichiers des utilisateurs (et `nextcloud.log`) : à sauvegarder avec la base
`/opt/nextcloud/certs/sio-ca.crt`	Certificat de l'autorité SIO-CA, monté dans les conteneurs
Paramètres d'administration > Vue d'ensemble	Contrôles de santé de Nextcloud, avec la cause et le correctif de chaque alerte
`sudo occ ldap:show-config s01`	Afficher la configuration de l'annuaire (sans le mot de passe)