Documentation Technique

Déploiement SSO Keycloak

Fédération Active Directory — SAML 2.0 — OIDC

Épreuve E6 — Réalisation professionnelle n°1

RJIBA Yanis

BTS SIO — Option SISR — Session 2026

NYM-IT — Avril 2026

Table des matières

Introduction

Ce document constitue la documentation technique de la première réalisation professionnelle réalisée dans le cadre de l'épreuve E6 du BTS Services Informatiques aux Organisations, option Solutions d'Infrastructure, Systèmes et Réseaux (SISR). Il décrit l'ensemble des travaux menés pour déployer une solution d'authentification centralisée au sein de l'infrastructure NYM-IT.

L'infrastructure NYM-IT hébergé plusieurs services internes — Nextcloud pour la collaboration, GLPI pour la gestion de parc, ainsi que divers outils d'administration et de supervision. Avant ce projet, chaque service gérait indépendamment l'authentification de ses utilisateurs via des connexions LDAP individuelles vers l'Active Directory. Cette approche décentralisée générait de la redondance, des risques d'erreur de configuration, et ne permettait pas le partage de sessions entre applications.

Le projet a consisté à déployer Keycloak, une plateforme open source d'Identity and Access Management (IAM) développée par Red Hat, comme point central d'authentification. Keycloak fédéré l'annuaire Active Directory existant et offre aux applications un service SSO (Single Sign-On) via les protocoles standards SAML 2.0 et OpenID Connect (OIDC). Un portail d'accès unifie a également été développé pour offrir aux utilisateurs un point d'entrée unique vers l'ensemble des services.

Cette documentation détaillé l'ensemble du processus : de l'architecture choisie au déploiement technique, en passant par la configuration de chaque intégration, la résolution des incidents rencontres (19 au total), et la mise en place de la supervision. Elle est destinée à servir de référence pour la maintenance et l'évolution de la solution.

1. Contexte et objectifs

1.1 Présentation de l'environnement

L'infrastructure NYM-IT est un laboratoire interne virtualise, construit autour du domaine Active Directory nym-it.local. Le serveur principal, SRV-NYMIT, fonctionne sous Debian 13 et hébergé l'ensemble des services conteneurises via Docker. Le contrôleur de domaine Windows Server (192.168.1.1) géré l'annuaire Active Directory qui centralisé les comptes utilisateurs de l'organisation.

Plusieurs services applicatifs sont hébergés sur cette infrastructure, chacun répondant a un besoin spécifique :

1. Nextcloud 33 — plateforme collaborative pour le partage de fichiers et l'édition de documents, déployée sur une VM dédiée (192.168.1.205)
2. GLPI 11 — solution de gestion de parc informatique et de ticketing, déployée en conteneur Docker (192.168.20.2, port 10101)
3. Portainer — interface web de gestion des conteneurs Docker, facilitant l'administration et la supervision
4. UrBackup — serveur de sauvegarde centralisé pour l'ensemble des machines de l'infrastructure
5. Vaultwarden — gestionnaire de mots de passe auto-hébergé, accessible depuis l'extérieur (hors périmètre SSO)

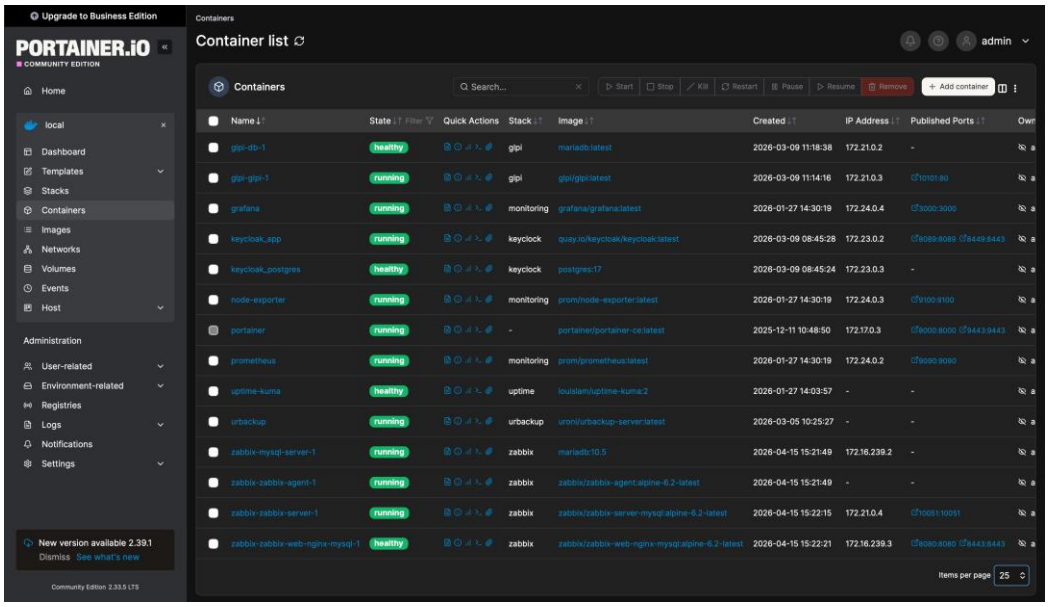

Fig. *Portainer — Vue des conteneurs Docker en cours d'exécution sur SRV-NYMIT*

1.2 Problématique

Avant ce projet, chaque application maintenait sa propre connexion LDAP vers l'Active Directory. Concrètement, Nextcloud, GLPI et les autres services étaient chacun configures indépendamment pour interroger l'annuaire AD et vérifier les credentials des utilisateurs. Cette approche décentralisée posait plusieurs problèmes concrets :

Du point de vue de l'expérience utilisateur, un collaborateur souhaitant accéder à Nextcloud puis a GLPI devait saisir ses identifiants deux fois, sans aucun mécanisme de session partagée. Au-delà de l'inconfort, cette situation augmentait le risque que les utilisateurs choisissent des mots de passe simples ou les notent pour éviter de les ressaisir.

Du point de vue de l'administration, la multiplication des connexions LDAP multipliait les points de configuration et donc les risques d'erreur. Chaque modification dans l'AD, changement de mot de passe du compte de service, modification d'une OU, ajout d'un attribut, devait être répercutée dans la configuration de chaque application individuellement.

Enfin, l'ajout d'utilisateurs externes (prestataires, partenaires) était particulièrement problématique : soit il fallait les intégrer au domaine AD (ce qui n'est pas souhaitable pour des comptes temporaires), soit il fallait créer des comptes locaux dans chaque application (ce qui est ingérable à l'échelle).

1.3 Objectifs

Face à ces constats, le projet vise à déployer une solution de Single Sign-On centrée sur Keycloak. Le choix de Keycloak repose sur plusieurs critères : c'est une solution open source mature, maintenue par Red Hat, qui supporte nativement les protocoles SAML 2.0 et OIDC. Contrairement a une solution SaaS comme Azure AD, Keycloak permet un hébergement on-premise, ce qui est essentiel dans un environnement ou les données doivent rester internes.

Les objectifs concrets du projet sont les suivants :

6. Déployer Keycloak comme Identity Provider (IdP) centralisé, en conteneur Docker avec PostgreSQL
7. Federer l'annuaire Active Directory existant via le protocole LDAP, sans migration ni duplication de comptes
8. Configurer l'authentification SSO SAML 2.0 pour Nextcloud et GLPI, en remplacement des connexions LDAP directes
9. Développer un portail d'accès unifie utilisant le protocole OIDC, offrant un point d'entrée unique aux utilisateurs

10. Permettre l'ajout futur d'utilisateurs externes directement dans Keycloak, sans impact sur l'Active Directory

1.4 Périmètre technique

Élément	Détail
Solution SSO	Keycloak 26.5.2 (conteneur Docker, image quay.io)
Protocoles	SAML 2.0 (Nextcloud, GLPI) et OIDC (Portail)
Annuaire source	Active Directory NYM-IT.local (Windows Server)
Applications intégrées	Nextcloud 33, GLPI 11, Portail d'accès OIDC
Reverse proxy	Nginx sur Debian 13 (terminaison SSL)
Conteneurisation	Docker Engine + Docker Compose
Certificats	Auto-signes (openssl, un par service)
Serveur hôte	SRV-NYMIT sur Openmediavault/Debian (Serveur physique)
BDD Keycloak	PostgreSQL 17 (conteneur Docker)
BDD GLPI	MariaDB 12 (conteneur Docker)

2. Architecture de la solution

2.1 Schema d'architecture générale

L'architecture déployée place Keycloak au centre de tous les flux d'authentification. Contrairement à l'ancienne approche ou chaque application interrogeait directement l'Active Directory, c'est désormais Keycloak qui assure l'interface unique avec l'AD via une connexion LDAP. Les applications, quant à elles, communiquent exclusivement avec Keycloak via les protocoles SAML 2.0 ou OIDC.

Un choix architectural important a été fait concernant l'accès à Keycloak : plutôt que de le placer derrière le reverse proxy Nginx comme les autres services, Keycloak est accessible directement sur le port 8449 en HTTPS. Ce choix a été motivé par les difficultés rencontrées avec les en-têtes de proxy (X-Forwarded-*) qui perturbaient les redirections SAML. Keycloak géré lui-même la terminaison SSL via les certificats montes en volume Docker.

Les autres services (GLPI, Nextcloud, le portail) sont quant à eux accédés via Nginx qui assure la terminaison SSL et le reverse proxy. Chaque service dispose de son propre certificat auto-signe et de son propre Virtual host Nginx.

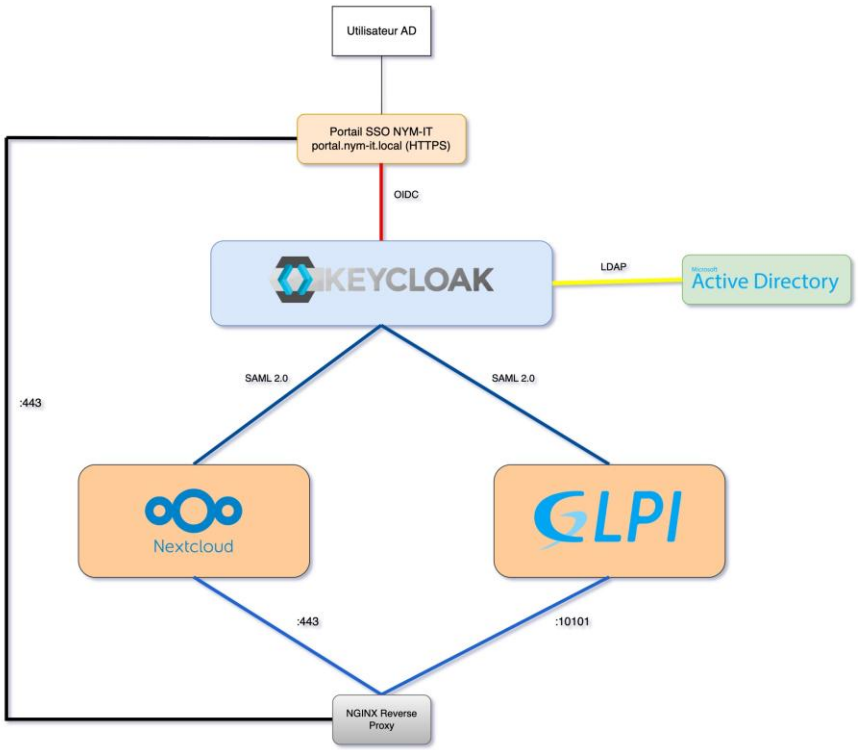

Fig. *Schema d'architecture SSO Keycloak NYM-IT*

> **Architecture logique**
> AD (NYM-IT.local) -- Fédération LDAP --> Keycloak (IdP, port 8449) -- SAML/OIDC --> Applications (Nextcloud, GLPI, Portail)

2.2 Flux d'authentification SSO

Le flux d'authentification typique via le portail SSO se déroule en six étapes. Ce mécanisme permet à l'utilisateur de ne s'authentifier qu'une seule fois pour accéder à l'ensemble des applications :

11. L'utilisateur accédé au portail SSO (portal.nym-it.local) via son navigateur
12. Le portail JavaScript détecté l'absence de token d'accès et redirige automatiquement vers Keycloak via le protocole OIDC (Authorization Code Flow)
13. Keycloak présenté sa page de connexion. L'utilisateur saisit ses identifiants AD (sAMAccountName et mot de passe). Keycloak vérifié les credentials contre l'Active Directory via la fédération LDAP
14. Après authentification réussie, Keycloak renvoie l'utilisateur vers le portail avec un code d'autorisation, échange ensuite contre un access token JWT
15. Le portail affiche les applications disponibles. Lorsque l'utilisateur clique sur Nextcloud ou GLPI, une requête IdP-initiated SAML est envoyée à Keycloak
16. Keycloak généré une assertion SAML signée contenant les attributs de l'utilisateur (username, email, groupes) et la transmet à l'application cible, qui créé automatiquement la session

2.3 Plan d'adressage et flux réseau

L'infrastructure utilise deux sous-réseaux : le réseau principal 192.168.1.0/24 pour les machines physiques et VMs, et le réseau Docker 192.168.20.0/24 pour les services conteneurises. Cette segmentation isole les communications inter-conteneurs du trafic réseau général.

Machine / Service	Adresse IP	Port(s)	FQDN
Contrôleur de domaine	192.168.1.1	389, 636	dc.nym-it.local
SRV-NYMIT (hôte Docker)	192.168.20.2	443 (Nginx)	srv-nymit.nym-it.local
Keycloak (Docker)	192.168.20.2	8449 (HTTPS direct)	keycloak.nym-it.local
GLPI (Docker)	192.168.20.2	10101 via Nginx 443	glpi.nym-it.local
Nextcloud (VM)	192.168.1.205	443	nextcloud.nym-it.local
Portail (Nginx)	SRV-NYMIT	443 (Nginx)	portal.nym-it.local

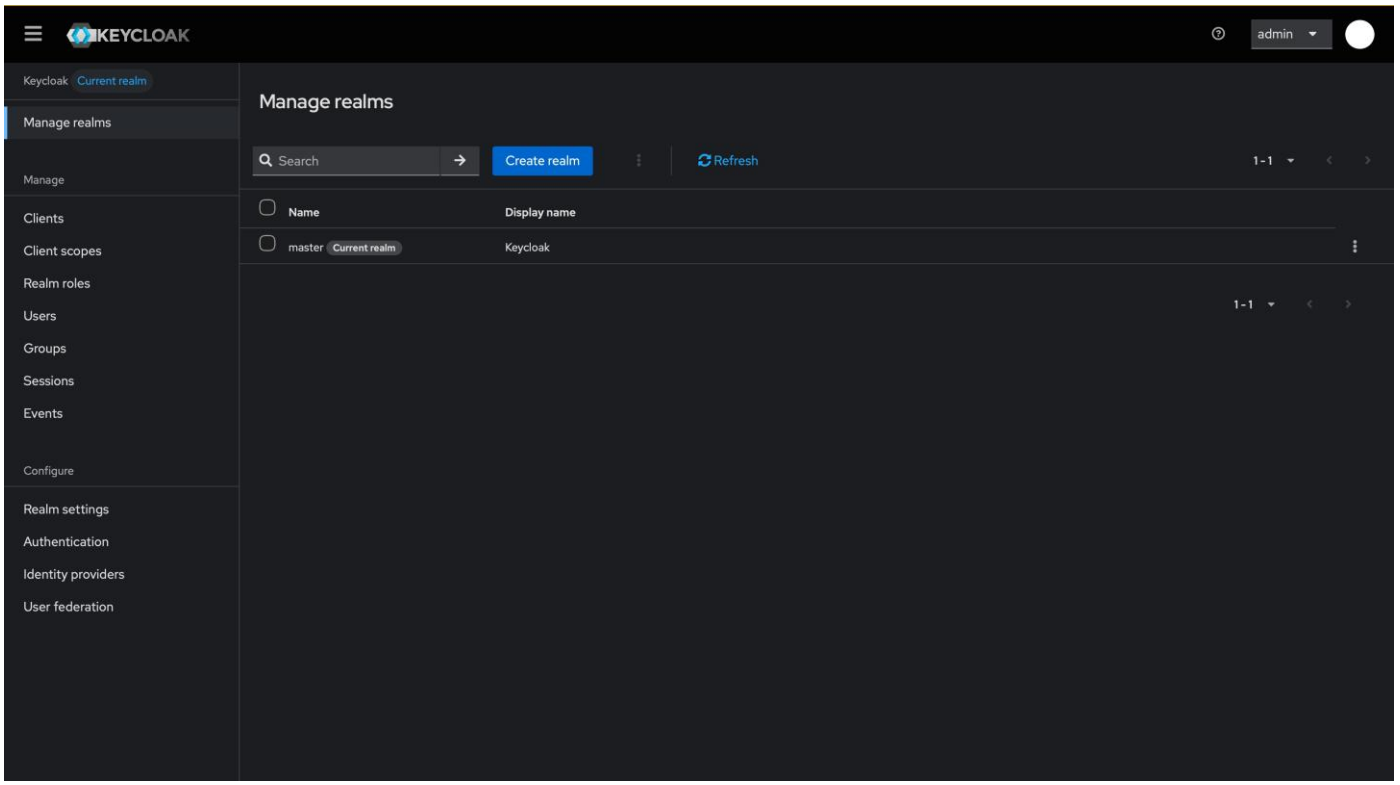

Fig. *Keycloak — Realm master (seul realm configure)*

3. Déploiement de Keycloak

3.1 Choix de déploiement Docker

Le déploiement de Keycloak a été réalisé via Docker Compose, ce qui offre plusieurs avantages pour un environnement lab : facilite de mise à jour (changement de tag d'image), isolation du service, reproductibilité de la configuration, et gestion simplifiée des dépendances. PostgreSQL 17 est déployé dans le même stack Docker et sert de base de donnees persistante pour Keycloak.

L'image officielle quay.io/keycloak/keycloak:latest est utilisée. Keycloak est démarré en mode production (commande start) avec la terminaison SSL gérée directement via des certificats montes en volume. Le port interne 8443 est mappe sur le port 8449 de l'hôte, ce qui évite les conflits avec d'autres services. PostgreSQL dispose d'un healthcheck (pg_isready) pour s'assurer que la base est prête avant le démarrage de Keycloak grâce à la directive depends_on avec condition service_healthy.

```yaml
services:
  keycloak:
    container_name: keycloak_app
    image: quay.io/keycloak/keycloak:latest
    restart: always
    environment:
      KEYCLOAK_ADMIN: ${KEYCLOAK_USER}
      KEYCLOAK_ADMIN_PASSWORD: ${KEYCLOAK_PASSWORD}
      KC_HOSTNAME: ${KEYCLOAK_URL}
      KC_DB: postgres
      KC_DB_USERNAME: ${POSTGRES_USER}
      KC_DB_PASSWORD: ${POSTGRES_PASSWORD}
      KC_DB_URL_HOST: keycloak_postgres
      KC_DB_URL_DATABASE: keycloak
      KC_HTTPS_CERTIFICATE_FILE: /etc/x509/https/tls.crt
      KC_HTTPS_CERTIFICATE_KEY_FILE: /etc/x509/https/tls.key
    volumes:
      - ./certs:/etc/x509/https
    depends_on:
      keycloak_postgres:
        condition: service_healthy
    ports:
      - "8089:8089"
      - "8449:8443"
    networks:
      - keycloak-network
    command:
      - start

  keycloak_postgres:
    container_name: keycloak_postgres
    image: postgres:17
    restart: always
    environment:
      POSTGRES_DB: keycloak
      POSTGRES_USER: ${POSTGRES_USER}
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
```

Fig. *docker-compose.yml — Service Keycloak (image, variables, ports)*

```yaml
  keycloak_postgres:
    container_name: keycloak_postgres
    image: postgres:17
    restart: always
    environment:
      POSTGRES_DB: keycloak
      POSTGRES_USER: ${POSTGRES_USER}
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
    volumes:
      - /srv/dev-disk-by-uuid-ebf98242-7571-4827-8af9-34386627471c/docker/keyclok/postgresql:/var/lib/postgresql/data
    networks:
      - keycloak-network
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U ${POSTGRES_USER} -d keycloak"]
      interval: 10s
      timeout: 5s
      retries: 5

networks:
  keycloak-network:
    name: keycloak-network
    driver: bridge
```

Fig. *docker-compose.yml — PostgreSQL 17, healthcheck pg_isready, networks*

3.2 Variables d'environnement critiques

La configuration de Keycloak repose principalement sur des variables d'environnement définies dans le docker-compose.yml. Certaines sont critiques et leur mauvaise configuration est la source de la majorité des incidents rencontres. Les credentials sensibles (mot de passe admin, credentials PostgreSQL) sont stockes dans un fichier .env non visible.

Variable	Valeur	Rôle
KC_HOSTNAME	${KEYCLOAK_URL}	FQDN public de Keycloak. CRITIQUE : doit correspondre exactement au nom utilise par les navigateurs clients.
KC_HOSTNAME_STRICT	false	Désactive la vérification stricte du hostname. Nécessaire en lab pour permettre l'accès via localhost ou IP.
KC_HTTPS_CERTIFICATE_FILE	/etc/x509/https/tls.crt	Chemin du certificat SSL dans le conteneur, monte via volume Docker.
KC_DB	postgres	Type de base de donnees.
KEYCLOAK_ADMIN	${KEYCLOAK_USER}	Compte administrateur initial (depuis fichier .env).

KC_HOSTNAME — Point critique du déploiement
La valeur de KC_HOSTNAME DOIT correspondre exactement au FQDN utilise par les navigateurs pour accéder à Keycloak. Si le navigateur accédé à keycloak.nym-it.local:8449 mais que KC_HOSTNAME est configure sur une autre valeur, Keycloak généré des URLs SAML/OIDC incorrectes, provoquant des bouclés de redirection infinies (ERR_TOO_MANY_REDIRECTS).

3.3 Configuration Nginx — Reverse Proxy

Nginx est installé directement sur l'hôte Debian et sert de reverse proxy HTTPS pour GLPI et le portail. Chaque service dispose de son propre Virtual host avec ses certificats SSL dédiés. Une redirection automatique HTTP vers HTTPS est configurée pour forcer les connexions sécurisées.

3.3.1 Virtual host glpi.nym-it.local

Nginx écoute sur le port 443 avec SSL et proxifie les requêtes vers le conteneur GLPI sur le port 10101 en HTTP interne. Les en-têtes X-Forwarded-Proto et X-Forwarded-For sont transmis pour que GLPI connaisse le protocole et l'IP d'origine du client.

```
server {
    listen 80;
    server_name glpi.nym-it.local;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name glpi.nym-it.local;

    ssl_certificate /etc/nginx/certs/glpi.crt;
    ssl_certificate_key /etc/nginx/certs/glpi.key;

    location / {
        proxy_pass http://127.0.0.1:10101;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto https;
    }
}
```

Fig. *Configuration Nginx — glpi.nym-it.local (reverse proxy HTTPS, port 10101)*

3.3.2 Virtual host portal.nym-it.local

Le portail est une application statique (HTML/CSS/JS) servie directement par Nginx en HTTPS depuis /var/www/html/portal. Il n'y a pas de proxy vers un conteneur Docker.

```
server {
    listen 80;
    server_name portal.nym-it.local;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name portal.nym-it.local;

    ssl_certificate /etc/nginx/certs/portal.crt;
    ssl_certificate_key /etc/nginx/certs/portal.key;

    root /var/www/html/portal;
    index index.html;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

Fig. *Configuration Nginx — portal.nym-it.local (HTTPS, fichiers statiques)*

3.4 Incidents lies au déploiement

Le déploiement de Keycloak a généré quatre incidents, principalement lies à la configuration du hostname et des ports. Ces incidents sont documentés en détail ci-dessous.

KC-01 — Port d'accès incorrect

Symptôme	Impossible d'accéder à l'interface Keycloak — page inaccessible dans le navigateur.
Cause racine	Le port interne 8443 de Keycloak est mappe sur le port 8449 de l'hôte (8449:8443 dans docker-compose). L'URL correcte est https://keycloak.nym-it.local:8449 et non :8443.
Résolution	Accès via https://keycloak.nym-it.local:8449. Vérification du mapping de ports avec docker compose ps.
Prévention	Toujours vérifier le mapping de ports dans le docker-compose.yml avant de tenter l'accès.

KC-02 — KC_HOSTNAME strict

Symptôme	Réponse vide lors du curl sur localhost:8449 ou via l'adresse IP directe.
Cause racine	KC_HOSTNAME est défini sur keycloak.nym-it.local — Keycloak rejette les requêtes arrivant via localhost ou IP car elles ne correspondent pas au hostname configure.
Résolution	Ajout de KC_HOSTNAME_STRICT: false dans le docker-compose.yml. Cette option autorise les accès depuis d'autres hostnames que celui configure.
Prévention	En environnement lab, toujours configurer KC_HOSTNAME_STRICT=false pour faciliter le diagnostic.

KC-03 — URL metadata SAML incorrecte

Symptôme	Page 'We are sorry... Page not found' lors de l'accès au descripteur SAML.
Cause racine	Le nom du realm utilise dans l'URL contenait une faute de frappe. L'URL du metadata SAML est sensible à la casse et au nom exact du realm.
Résolution	Correction de l'URL : https://keycloak.nym-it.local:8449/realms/master/protocol/saml/descriptor. Le realm utilise est 'master' (par défaut).

KC-04 — Warnings JGroups au démarrage

Symptôme	Warnings 'JOIN attempts timed out' dans les logs de démarrage Keycloak, provoquant un démarrage lent.
Cause racine	Une ancienne entrée dans la table JGROUPSPING de PostgreSQL pointait vers un conteneur Docker qui n'existe plus. JGroups est le protocole de clustering de Keycloak.
Résolution	Purge de la table : DELETE FROM JGROUPSPING; — En déploiement single-node, Keycloak fonctionne en mode singleton et n'a pas besoin de cluster.

4. Fédération LDAP avec Active Directory

4.1 Principe et choix de configuration

La fédération LDAP est le mécanisme central qui relie Keycloak a l'Active Directory existant. Plutôt que de dupliquer les comptes utilisateurs dans Keycloak, la fédération permet de les synchroniser depuis l'AD : Keycloak importe les informations des comptes (nom, prénom, email, groupes) et vérifié les mots de passe directement auprès de l'AD lors de chaque connexion.

Le mode READ_ONLY a été choisi volontairement : Keycloak peut lire les informations de l'AD mais ne peut pas les modifier. L'Active Directory reste ainsi la source de vérité unique pour les comptes utilisateurs. Toute modification (changement de mot de passe, désactivation de compte, changement de groupe) se fait exclusivement dans l'AD et est automatiquement reflétée dans Keycloak lors de la prochaine synchronisation.

Un compte de service dédié, Sync_keycloack, a été créé dans l'OU Connecteurs de l'AD avec des droits de lecture seule sur les OUs concernées. Ce compte est utilisé par Keycloak pour se connecter à l'annuaire (Bind DN). Il est essentiel de ne jamais utiliser un compte administrateur de domaine pour cette connexion, par principe de moindre privilégié.

4.2 Configuration du provider LDAP

Le tableau ci-dessous récapitule l'ensemble des paramètres configures dans le provider LDAP de Keycloak. Le choix de sAMAccountName comme attribut username (plutôt que UserPrincipalName) a été fait pour obtenir des noms de connexion courts (ex : yanis.rjiba) plutôt que des noms qualifies avec le domaine (ex: yanis.rjiba@NYM-IT.local).

Paramètre	Valeur
Vendor	Active Directory
Connection URL	ldap://192.168.1.1
Bind DN	CN=Sync_keycloack,OU=Connecteurs,OU=SERVICES,DC=NYM-IT,DC=local
Users DN	OU=SERVICES,DC=NYM-IT,DC=local
Username LDAP attribute	sAMAccountName
UUID LDAP attribute	objectGUID
User Object Classes	person, organizationalPerson, user
Edit Mode	READ_ONLY

Paramètre	Valeur
Search Scope	Subtree (recherche récursive dans les sous-OUs)
Import Users / Sync Registrations	ON / ON

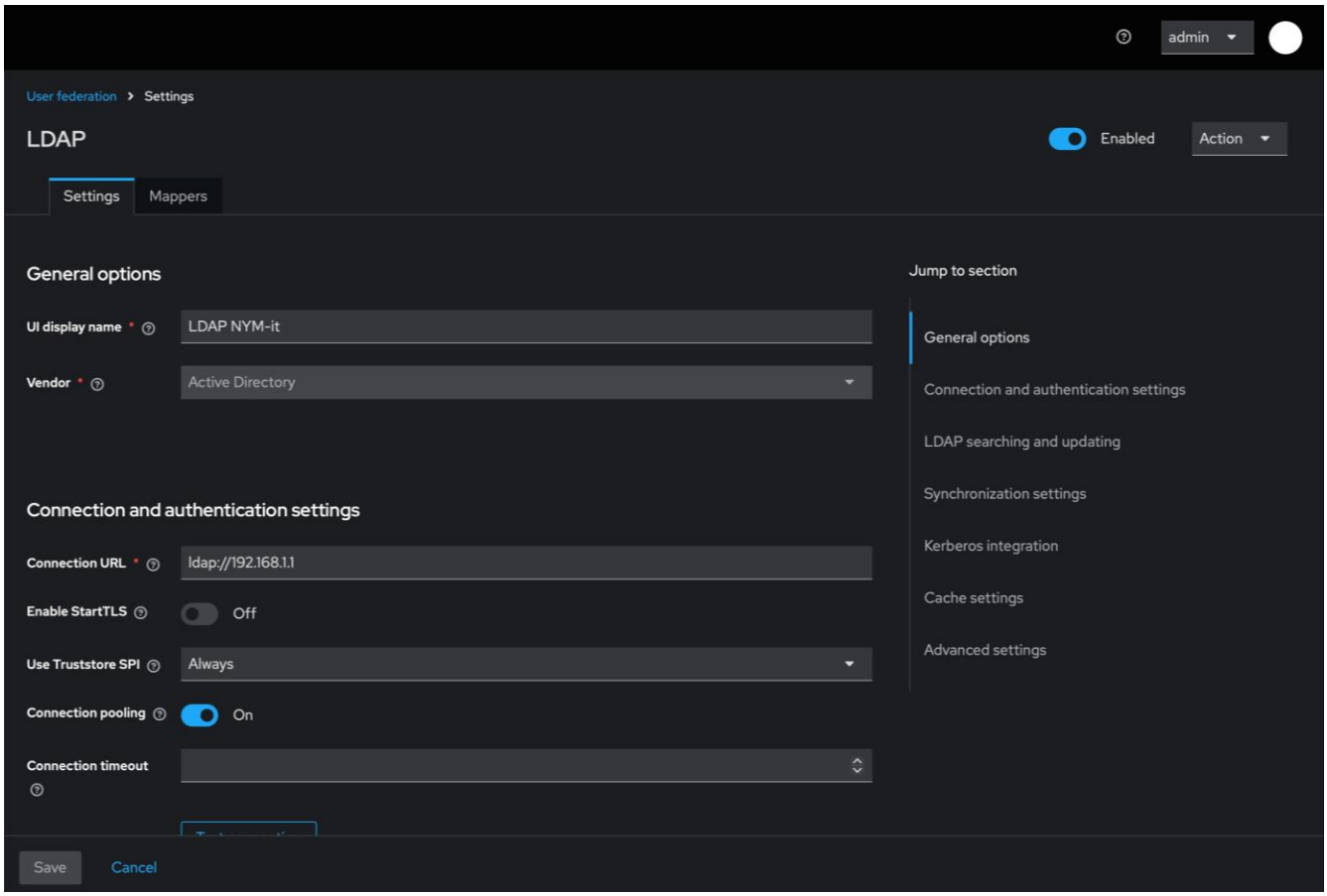

Fig. *Configuration LDAP — Connexion et général options*

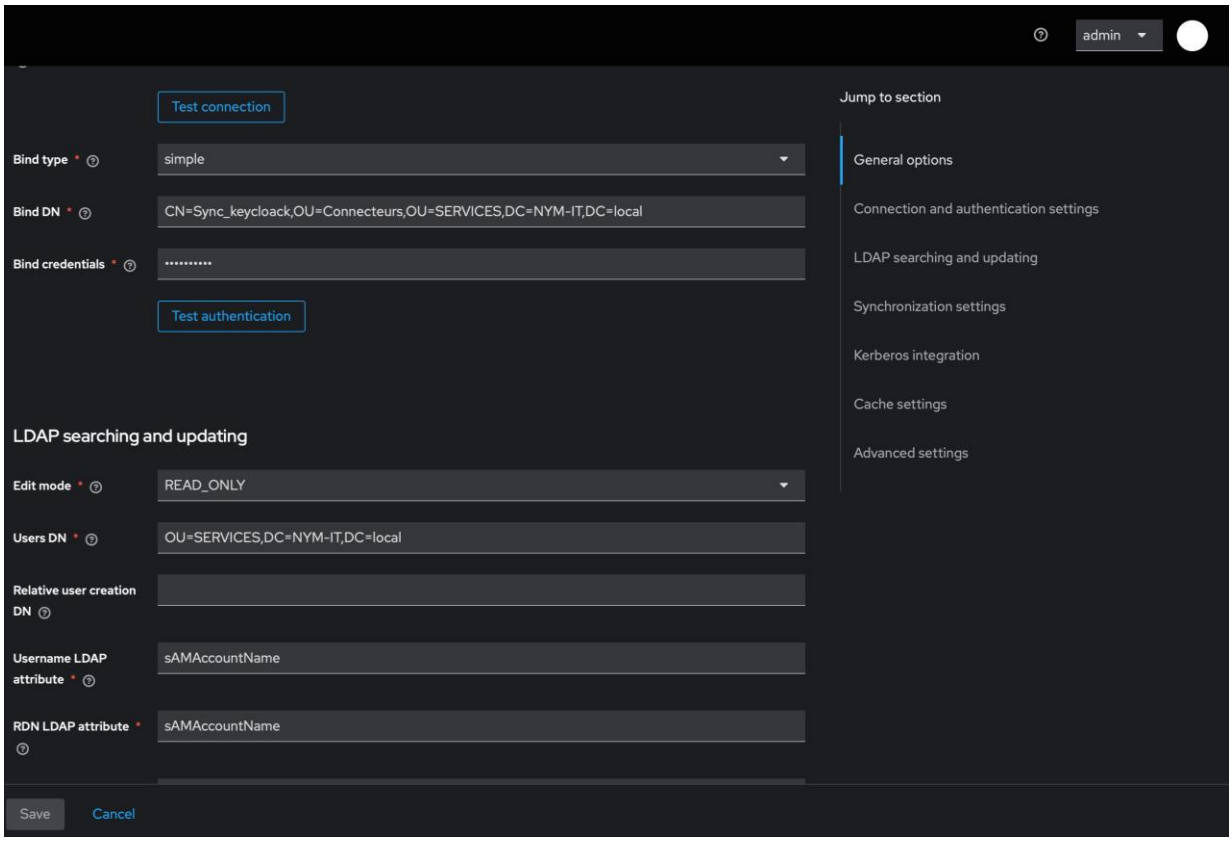

Fig. *Configuration LDAP — Bind DN et mode d'édition READ_ONLY*

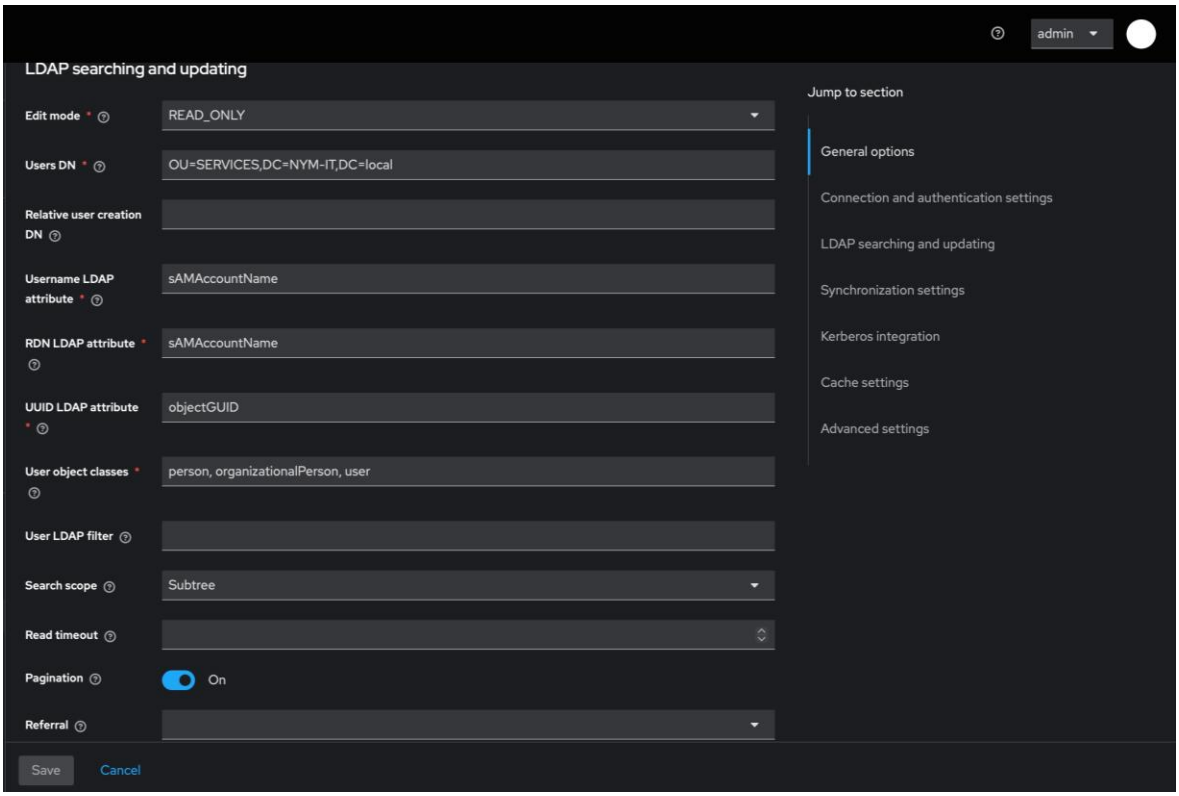

Fig. *Configuration LDAP — Users DN, sAMAccountName, Subtree search*

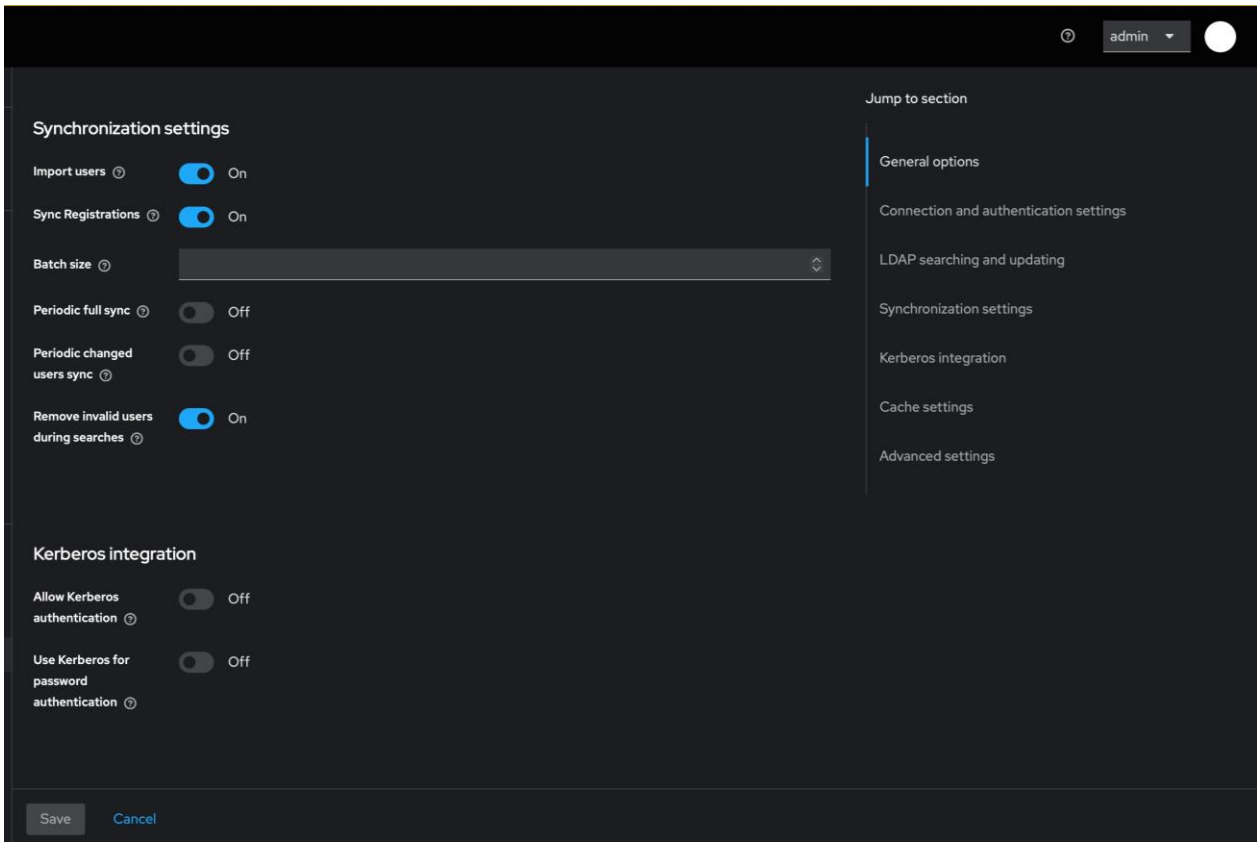

Fig. *Configuration LDAP — Synchronisation settings et Kerberos (désactivé)*

4.3 Mappers LDAP

Les mappers LDAP définissent la correspondance entre les attributs de l'Active Directory et les attributs internes de Keycloak. Huit mappers sont configures au total. Les principaux assurent la correspondance des champs username (sAMAccountName), email (mail), prénom (givenName) et nom (sn). Des mappers supplémentaires gèrent les dates de création/modification, le nom complet (full name) et les contrôles de compte MSAD (désactivation, expiration).

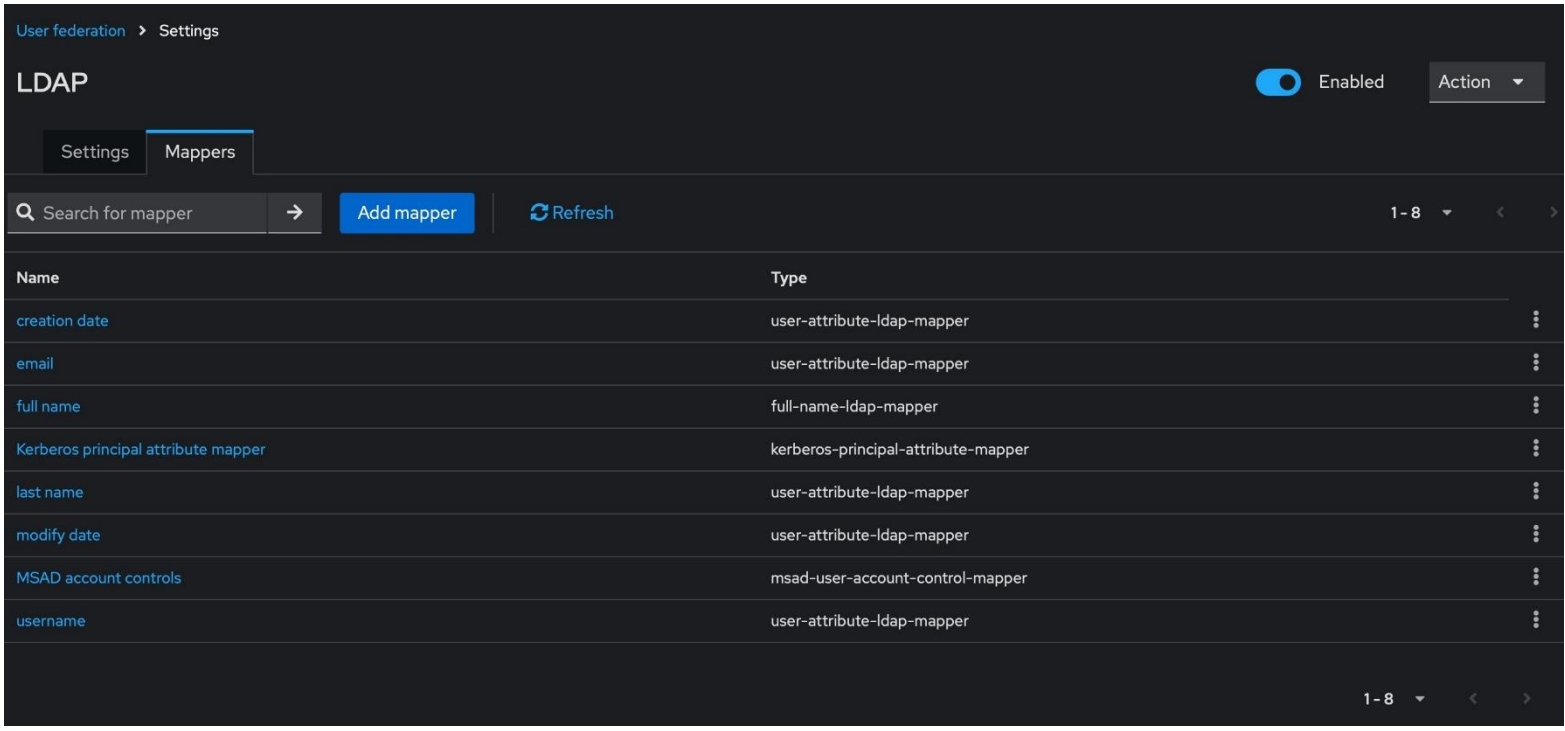

Fig. *Liste des 8 LDAP Mappers configures*

4.4 Résultat de la synchronisation

La synchronisation initiale a importé avec succès 14 utilisateurs depuis l'Active Directory. Cette opération est réalisée manuellement via le bouton 'Sync all users' dans les paramètres du provider LDAP. La synchronisation périodique automatique n'est pas activée car l'environnement est un lab avec peu de changements de comptes.

4.5 Incidents liés à la fédération LDAP

Deux incidents majeurs ont été rencontres lors de la configuration de la fédération, tous deux lies au format de l'identifiant utilisateur.

LDAP-01 — Format du username incorrect

Symptôme	Les utilisateurs se connectent avec user@NYM-IT.local au lieu de simplement 'user'. Cela pose des problèmes de correspondance dans Nextcloud et GLPI.
Cause racine	L'attribut Username LDAP était initialement configuré sur UserPrincipalName (UPN), qui inclut le suffixe de domaine (@NYM-IT.local). Les applications attendent un username court.

Résolution	Changement de Username LDAP attribute de UserPrincipalName vers sAMAccountName. Puis opération 'Remove imported' suivie de 'Sync all users' pour réimporter tous les utilisateurs avec le bon format d'identifiant.
Prévention	Toujours utiliser sAMAccountName dans un environnement Active Directory pour obtenir des usernames courts.

LDAP-02 — Persistance des anciens usernames

Symptôme	Après le changement d'attribut username, les utilisateurs déjà importes conservent leur ancien format (UPN) dans la base Keycloak.
Cause racine	Keycloak ne met pas à jour automatiquement les usernames des utilisateurs existants lors d'un changement de configuration du provider LDAP. Les anciens comptes restent avec le format UPN.
Résolution	Procédure complète : User fédération > Action > Remove imported (supprime tous les comptes fédérés de la base Keycloak), puis Sync all users pour réimporter avec le nouveau format sAMAccountName. Cette opération remet à zéro tous les comptes fédérés.

5. Intégration SAML — Nextcloud

5.1 Choix du protocole et du plugin

Nextcloud supporte nativement le protocole SAML 2.0 via le plugin 'SSO & SAML authentication' (user_saml), disponible directement dans le marketplace Nextcloud et maintenu par Nextcloud. Ce plugin supporte les flux SP-initiated et IdP-initiated. Le choix de SAML 2.0 pour Nextcloud (plutôt qu'OIDC) est motivé par la maturité de l'intégration et la richesse des attributs transmissibles dans les assertions SAML (username, email, groupes).

5.2 Configuration du client SAML dans Keycloak

Un client SAML est créé dans Keycloak pour représenter Nextcloud en tant que Service Provider (SP). Le Client ID correspond à l'Entity ID du SP Nextcloud. Le champ IDP-Initiated SSO URL name est configure sur 'nextcloud', ce qui permet de générer des liens d'accès direct depuis le portail SSO via l'URL /realms/master/protocol/saml/clients/nextcloud.

Paramètre	Valeur
Client ID	https://nextcloud.nym-it.local
Root URL / Valid redirect URIs	https://nextcloud.nym-it.local/*
ACS URL	https://nextcloud.nym-it.local/apps/user_saml/saml/acs
Name ID format	username
Force POST binding	ON
IDP-Initiated SSO URL name	nextcloud
Client signature required	OFF

5.3 Configuration côté Nextcloud

Dans l'interface d'administration Nextcloud (Paramètres > SSO & SAML authentication), l'identifiant de l'entité IdP pointe vers le realm master de Keycloak avec le port 8449. L'option 'Autoriser l'utilisation de plusieurs systèmes d'authentification' est activée, ce qui permet de conserver un accès administrateur local (login direct par mot de passe) en cas de panne du SSO Keycloak une précaution essentielle.

Le mappage des attributs traduit les attributs SAML reçus depuis Keycloak vers les champs Nextcloud : l'attribut username est utilisé comme identifiant unique (UID), sAMAccountName

comme nom d'affichage, email pour l'adresse électronique, et groups pour l'appartenance aux groupes avec le préfixe saml_.

Paramètre Nextcloud	Valeur
IdP Entity ID	https://keycloak.nym-it.local:8449/realms/master
URL SSO	https://keycloak.nym-it.local:8449/realms/master/protocol/saml
SP Entity ID	https://nextcloud.nym-it.local
Attribut UID	username
Attribut nom d'utilisateur	sAMAccountName
Attribut email	email
Attribut groupes	groups (préfixe : saml_)
Format NameID	Non spécifié

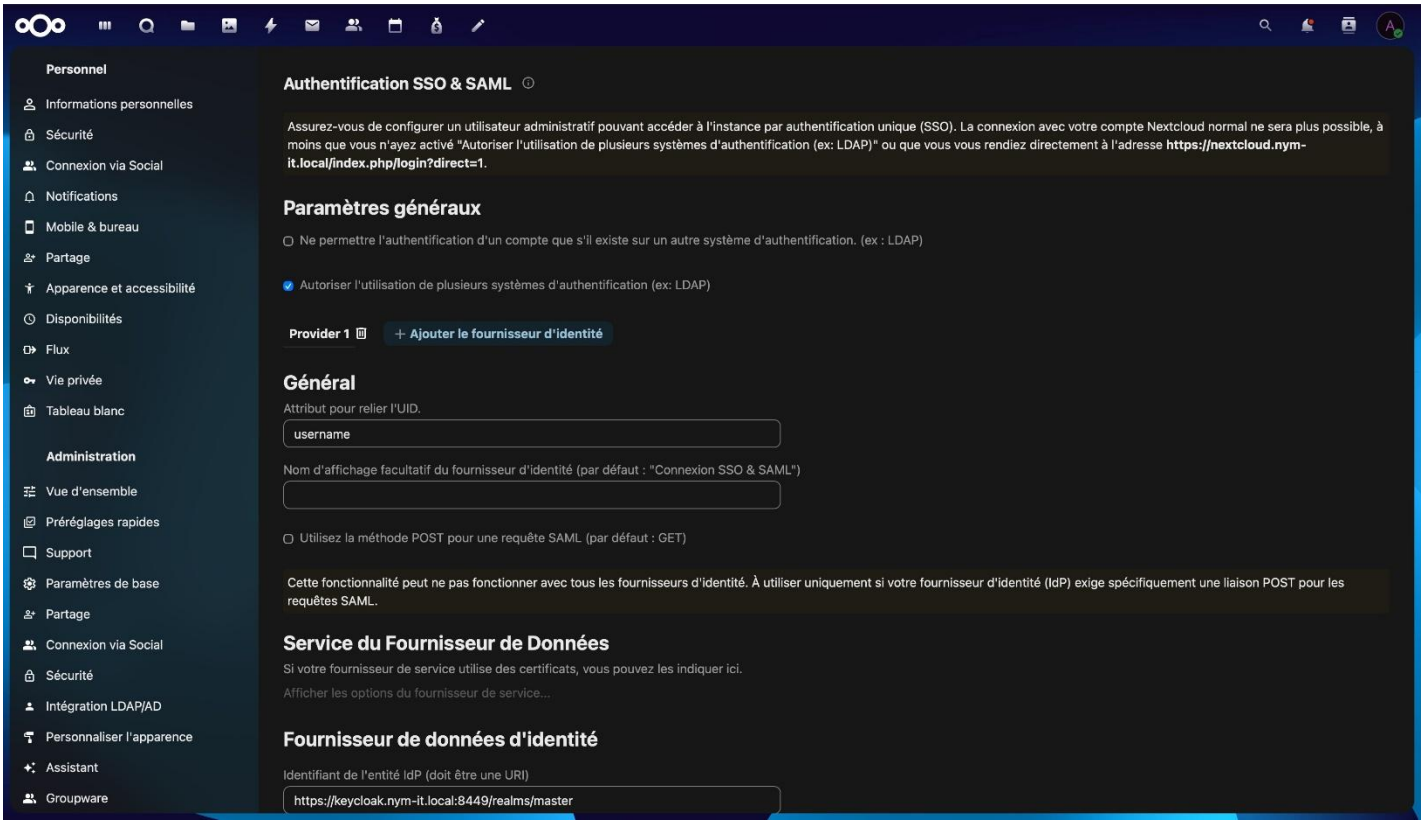

Fig. *Nextcloud — Configuration SSO & SAML, paramètres généraux*

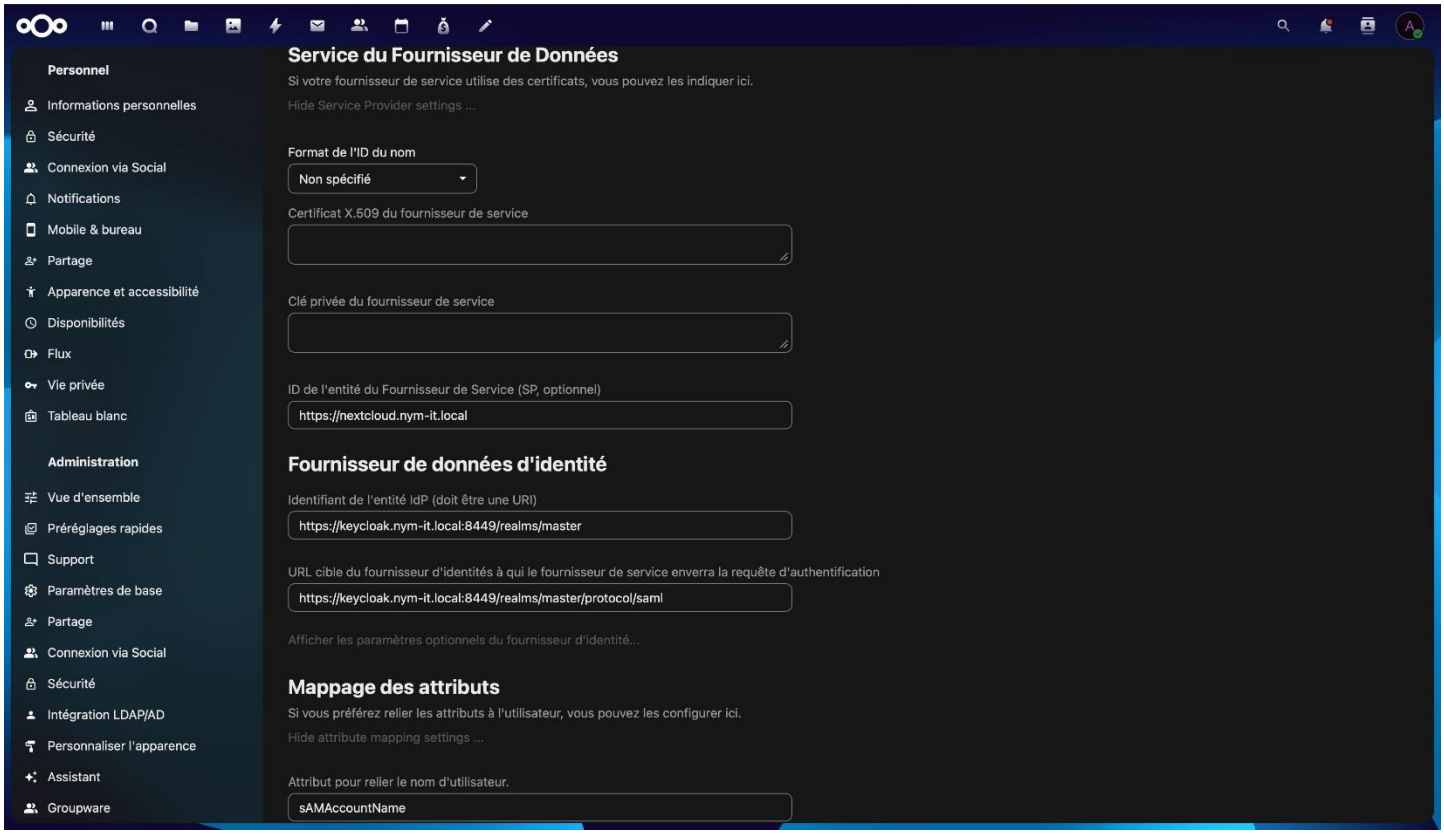

Fig. *Nextcloud — Entity IDs (SP et IdP) et URL SSO*

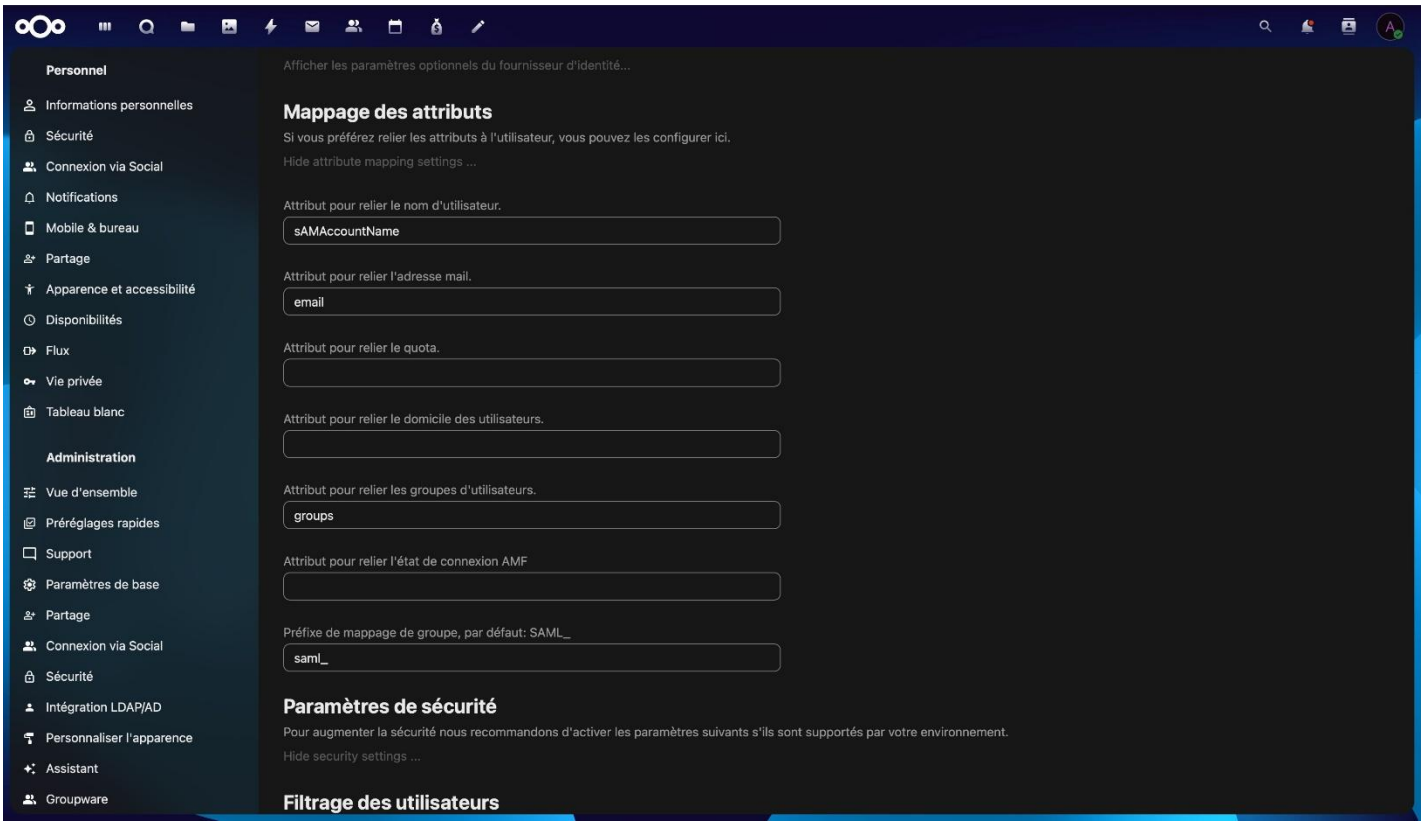

Fig. *Nextcloud — Mappage des attributs SAML vers les champs Nextcloud*

5.4 Test de connexion SSO

La connexion SSO a été testée avec succès. Après authentification sur Keycloak avec des identifiants Active Directory, l'utilisateur est automatiquement connecté à Nextcloud et accédé a ses dossiers d'équipe. La création de compte est automatique (JIT User Création) lors de la première connexion.

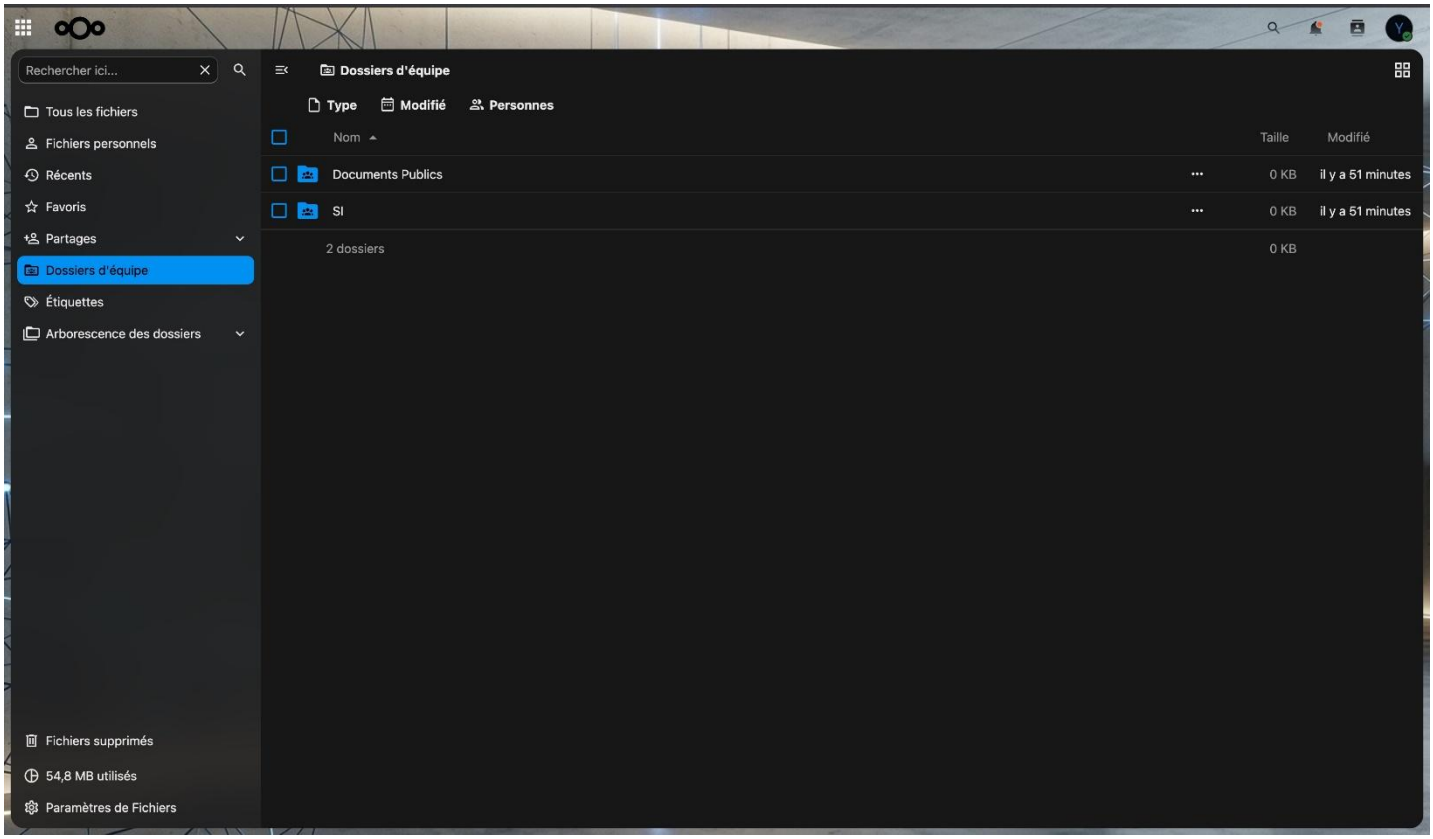

Fig. *Connexion SSO Nextcloud réussie — accès aux dossiers d'équipe*

5.5 Gestion des certificats auto-signes

Nextcloud doit faire confiance au certificat auto-signe de Keycloak pour que les échanges SAML fonctionnent (vérification de la signature des assertions). La même procédure a été nécessaire pour Collabora CODE. Le certificat est copie dans le trust store du conteneur puis les certificats CA sont mis à jour :

```
docker cp nym-it.local.crt nextcloud:/usr/local/share/ca-certifiçates/
docker exec nextcloud update-ça-certifiçates
docker restart nextcloud
```

5.6 Incidents lies a Nextcloud

NC-01 — Entity ID mismatch

Symptôme	Erreur 'invalid_request' lors de la redirection vers Keycloak.
Cause racine	Le Client ID dans Keycloak ne correspondait pas exactement au SP Entity ID dans Nextcloud. SAML exige une correspondance stricte.
Résolution	Alignement exact des deux valeurs sur https://nextcloud.nym-it.local dans les deux interfaces.
Prévention	Toujours copier-coller l'Entity ID d'une interface a l'autre pour éviter les fautes de frappe.

NC-02 — Certificat X.509 corrompu

Symptôme	Erreur 'Invalid byte 1 of 1-byte UTF-8 sequence' lors de la validation de la signature SAML.
Cause racine	Lors du copier-coller du certificat de signature depuis le metadata XML de Keycloak, des caractères parasites ont été inclus (clé de session et formats NameID colles au certificat Base64).
Résolution	Extraction manuelle du certificat uniquement depuis la balise <ds:X509Certifiçate>. Vérification systématique avec : openssl x509 -in cert.pem -text -noout.
Prévention	Toujours exporter le certificat depuis Keycloak > Realm settings > Keys plutôt que depuis le metadata XML brut.

NC-03 — NameID Format incompatible

Symptôme	Erreur 'Unsupported NameIDFormat' renvoyée par Keycloak lors de la tentative de connexion.
Cause racine	Le format NameID dans Nextcloud était configure sur 'Nom d'utilisateur' mais Keycloak était configure sur un format différent (email). L'incohérence empêchait l'échange SAML.
Résolution	Configuration du format NameID dans Nextcloud sur 'Non spécifié' (laisse Keycloak décider) et dans Keycloak sur 'username'.

NC-04 — Compte non approvisionne

Symptôme	Message 'compte non approvisionne' dans Nextcloud après authentification réussie sur Keycloak.
Cause racine	Le champ 'Attribut pour relier l'UID' était vide dans la configuration SAML. De plus, l'ancien LDAP direct Nextcloud-AD était encore actif, créant des conflits d'identité entre les comptes LDAP et les comptes SAML.
Résolution	Remplissage du champ UID avec 'username', désactivation complète de l'intégration LDAP directe de Nextcloud, et resynchronisation Keycloak avec sAMAccountName.
Prévention	Toujours désactiver la liaison LDAP directe quand le SSO est actif pour éviter les conflits.

6. Intégration SAML — GLPI

6.1 Spécificités et complexité

L'intégration SAML avec GLPI s'est révélée significativement plus complexe que celle de Nextcloud. GLPI 11 ne dispose pas d'un support SAML natif aussi mature. Le plugin utilise est samlSSO (version 1.2.5), développé par Chris Gralike sous licence GPLv3+. Ce plugin n'est pas disponible dans le marketplace GLPI standard et doit être téléchargé manuellement depuis GitHub puis installe dans le conteneur Docker.

L'installation dans un conteneur Docker a ajouté une couche de complexité : le chemin d'installation des plugins dans l'image glpi/glpi:latest n'est pas le chemin standard (/var/www/glpi/plugins) mais /var/glpi/marketplace/. Plusieurs particularités non documentées de GLPI ont été découvertes au cours de l'intégration. Chacune bloquait complètement le fonctionnement du SSO tant qu'elle n'était pas corrigée.

> **Points critiques non documentés**
> Les spécificités suivantes ne sont documentées nulle part dans la documentation officielle de GLPI ni dans celle du plugin samlSSO. Elles ont été identifiées par essai-erreur et analyse des logs Keycloak. Leur non-respect empêche complètement le fonctionnement du SSO.

6.2 Client SAML dans Keycloak

La configuration du client SAML pour GLPI présenté plusieurs différences critiques par rapport à Nextcloud. Le Client ID doit impérativement se terminer par un slash (/), le Name ID format est configure sur 'email' avec Force name ID format active, et le Force POST binding est désactivé car le plugin samlSSO utilise le Redirect Binding (GET) par défaut avec compression déflaté.

Paramètre	Valeur
Client ID	https://glpi.nym-it.local/ (slash final OBLIGATOIRE)
ACS URL	https://glpi.nym-it.local/marketplace/samlsso/front/acs/1
Name ID format	email (Force name ID format : ON)
Force POST binding	OFF
IDP-Initiated SSO URL name / Relay State	glpi / 1
Sign documents / Sign assertions	ON / OFF

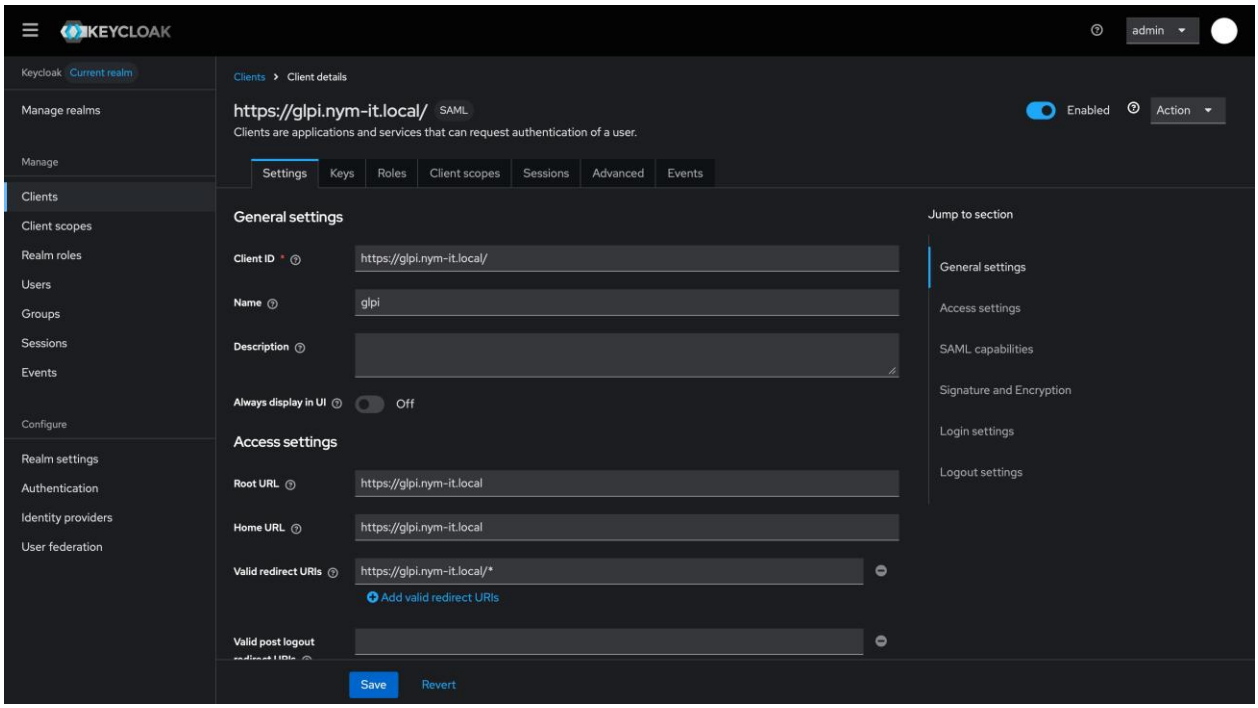

Fig. *Client SAML GLPI (HTTPS) — Client ID avec slash final*

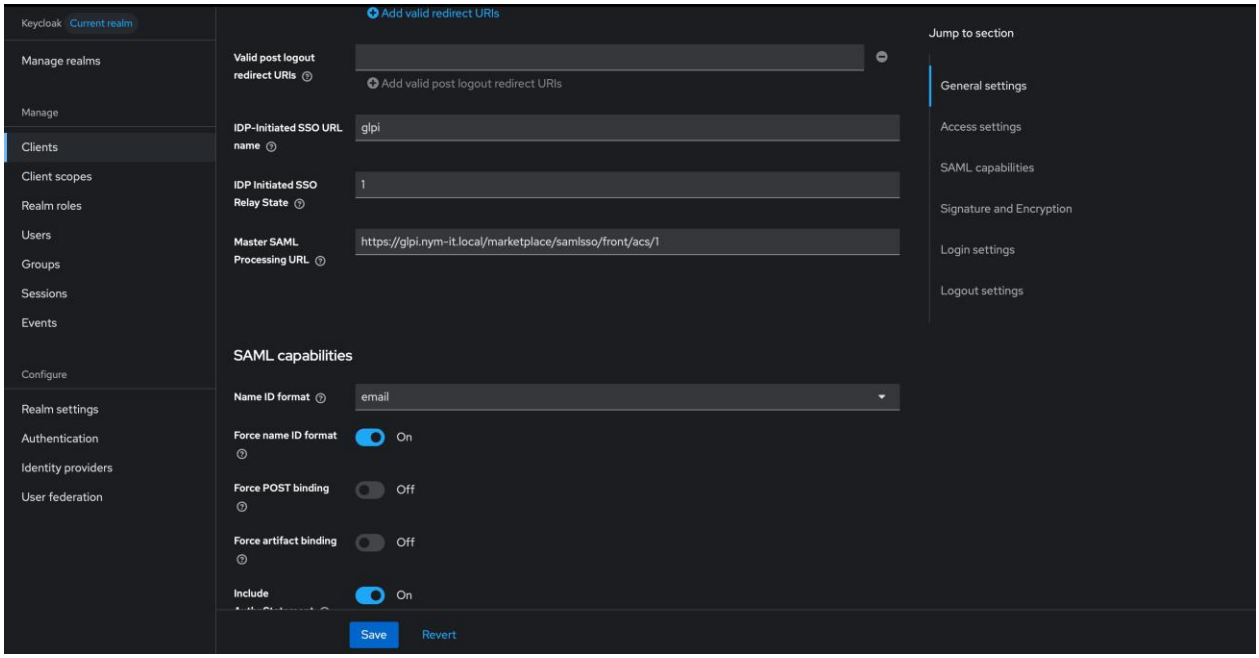

Fig. *Client SAML GLPI — IDP-Initiated SSO, ACS URL, SAML capabilities*

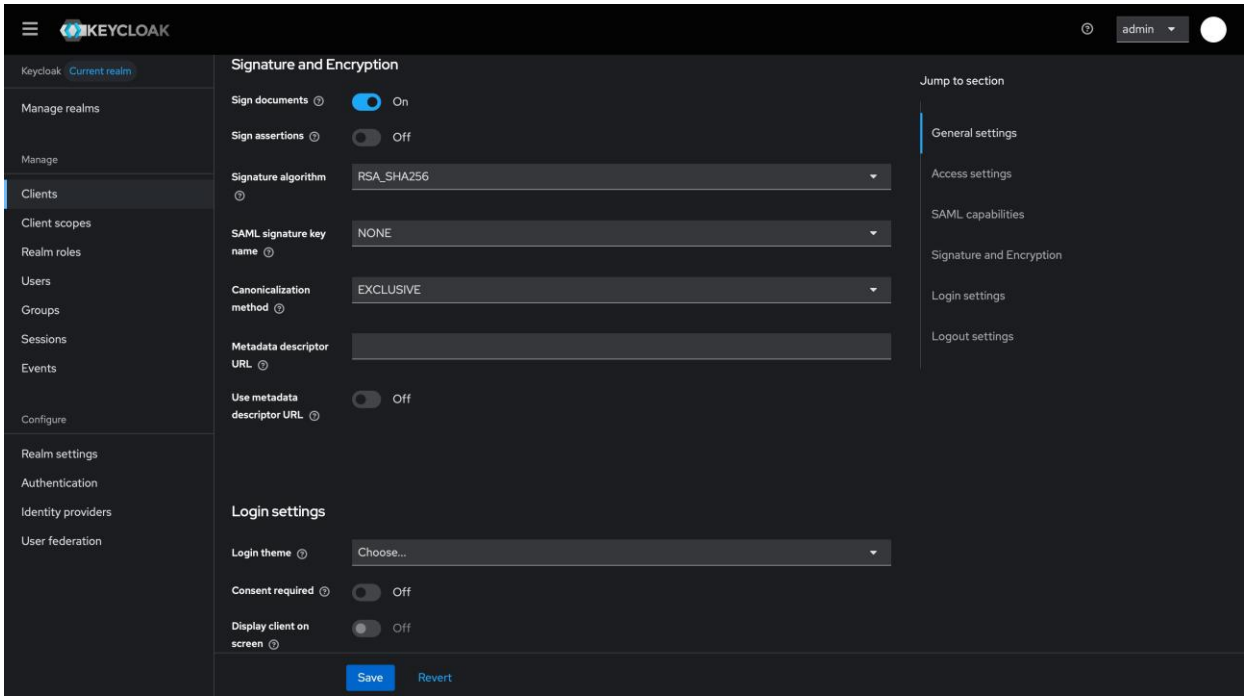

Fig. *Client SAML GLPI — Signature, Encryptions, Login settings*

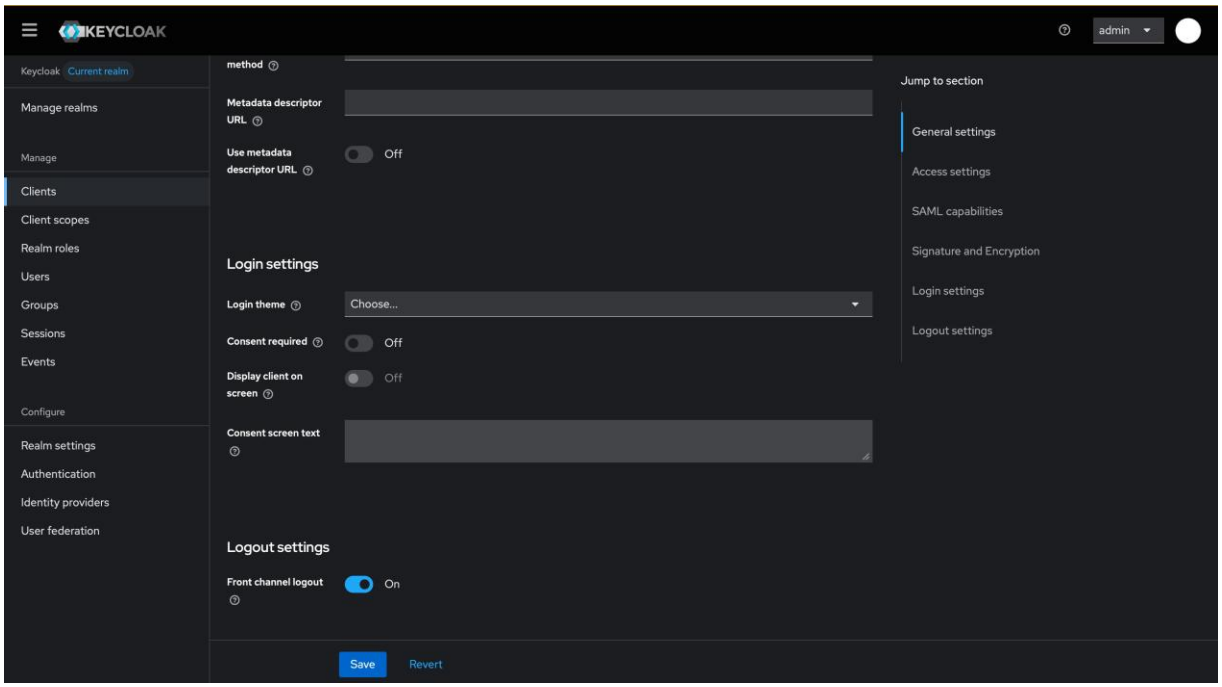

Fig. *Client SAML GLPI — Logout settings*

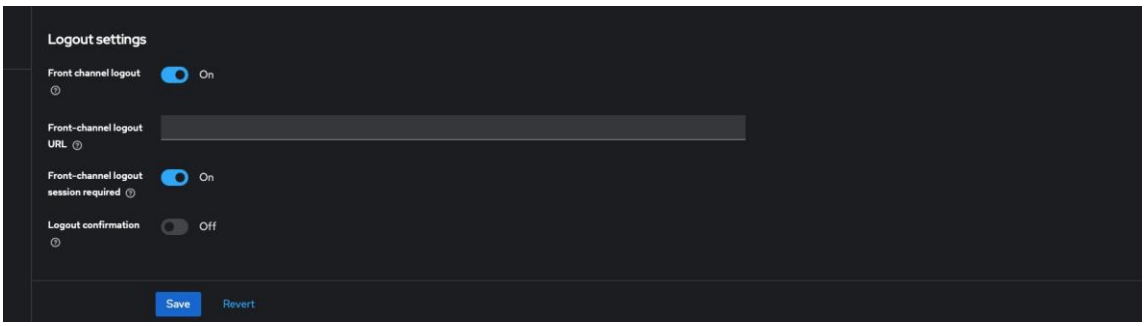

Fig. *Client SAML GLPI — Logout settings (détail)*

6.3 Client Scopes et Mappers

Un point important : les scopes SAML par défaut de Keycloak (role_list, saml_organization) doivent etre supprimés du client GLPI. Ces scopes envoient des attributs supplémentaires dans l'assertion SAML qui entrent en conflit avec les mappers du scope dédié, provoquant l'erreur 'Found an Attribute élément with duplicated Name'. Seul le scope dédié (-dedicated) avec les mappers username, email et groups doit être conserve.

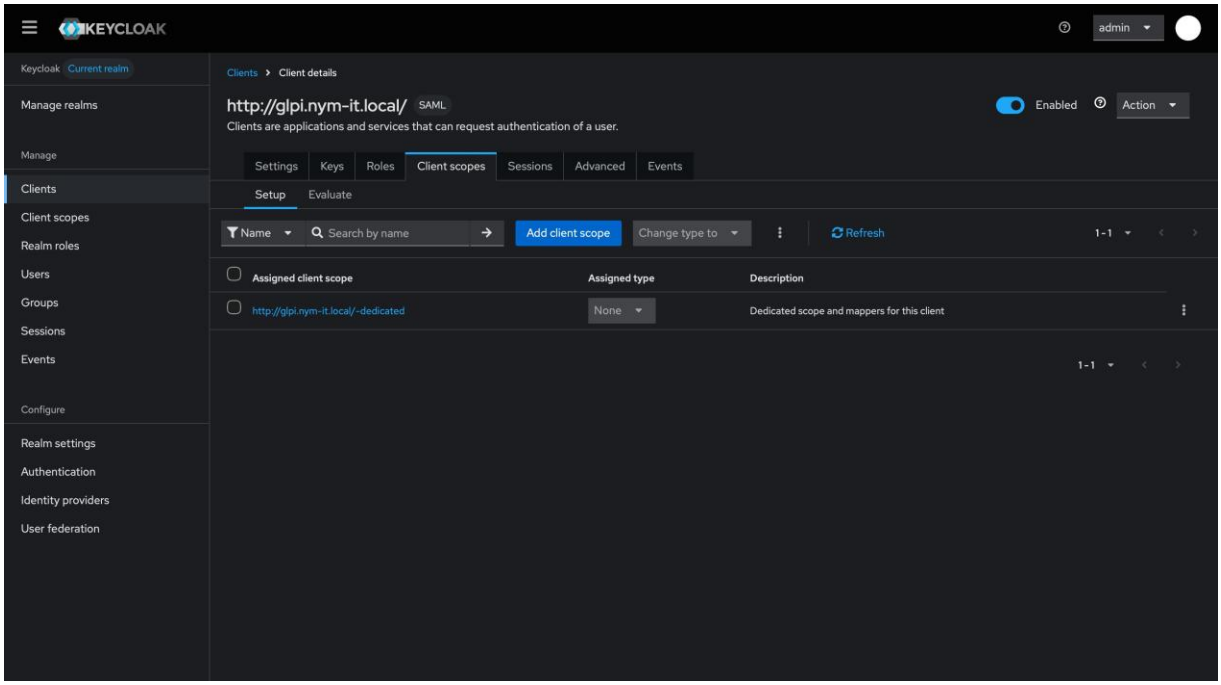

Fig. *Client Scopes GLPI — seul le scope dédié est conserve*

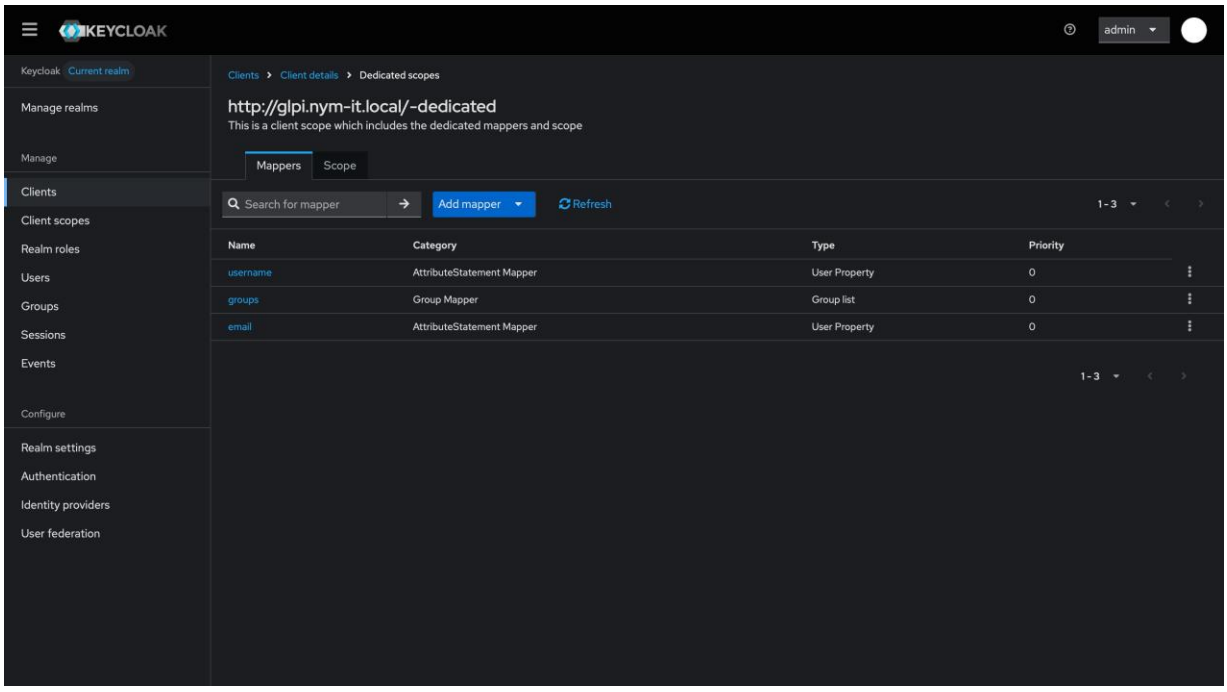

Fig. *Mappers GLPI — username, groups, email dans le scope dédié*

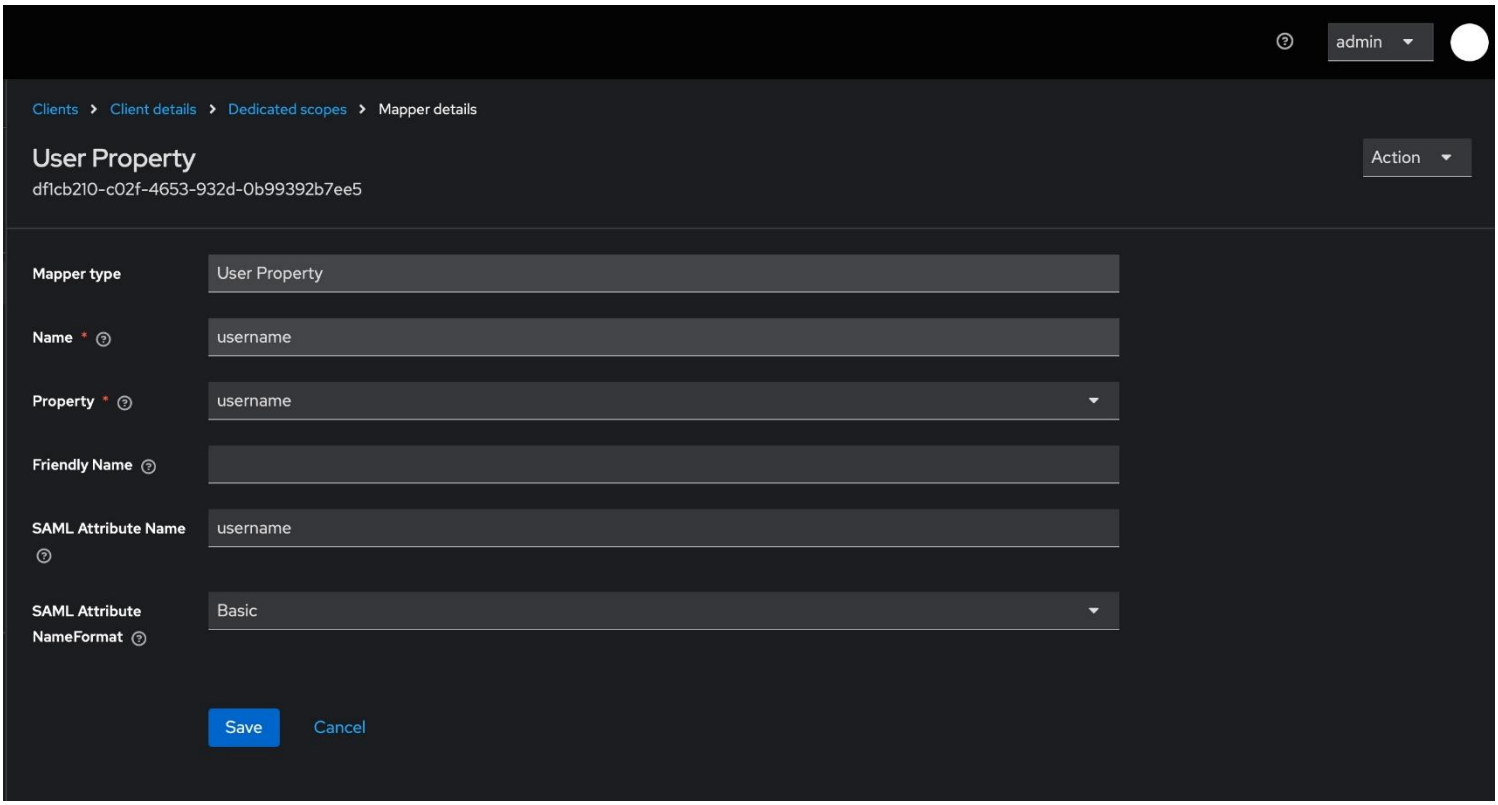

Fig. *Mapper detail — username (User Property, SAML Attribute Name Basic)*

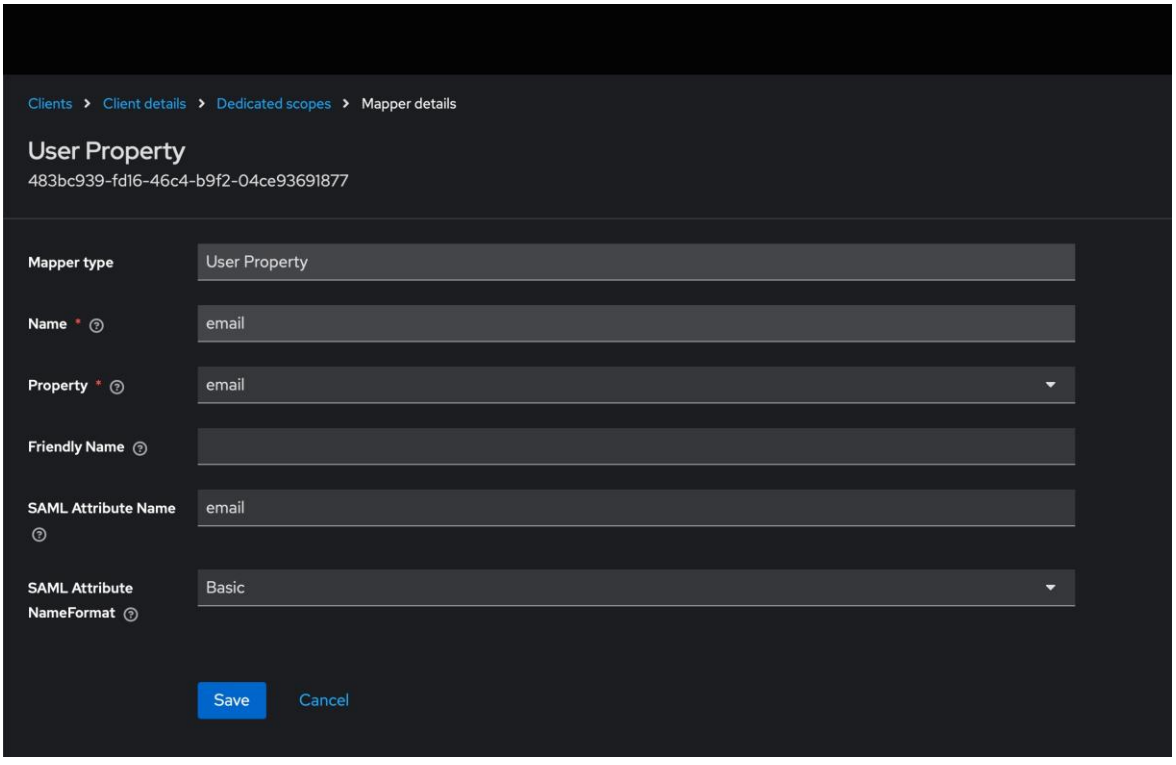

Fig. *Mapper détail — email (User Property)*

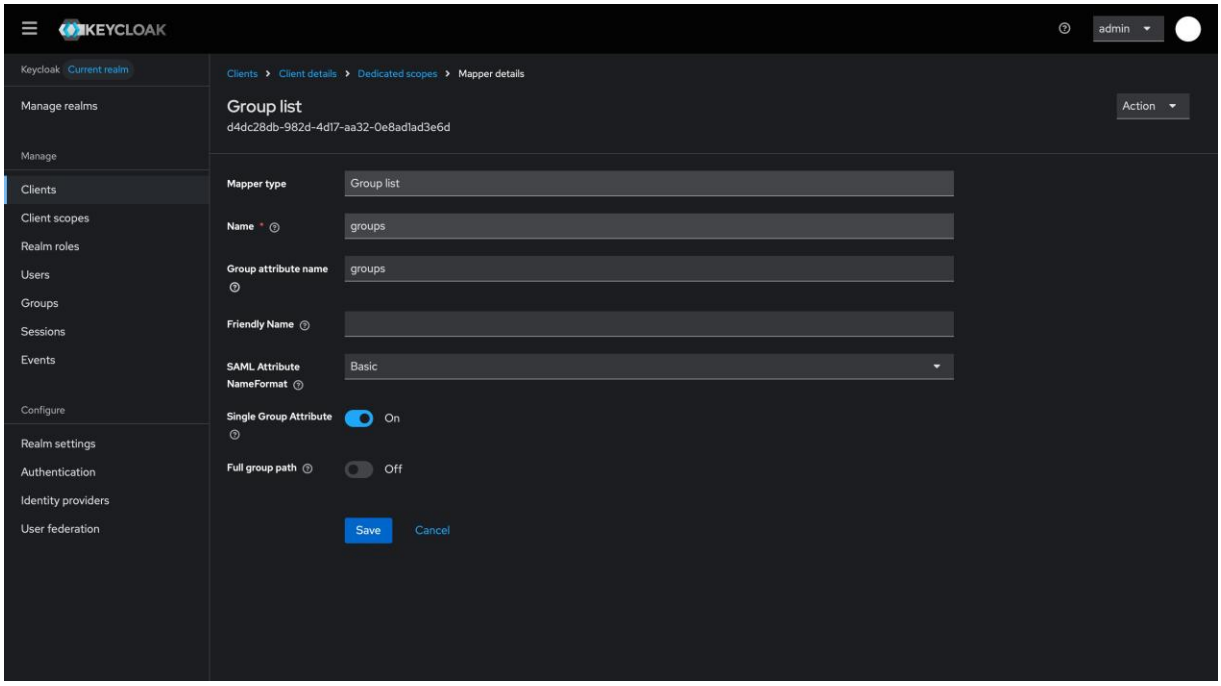

Fig. *Mapper detail — groups (Group list, Single Group Attribute ON)*

6.4 Configuration côté GLPI

La configuration du plugin samlSSO se fait dans GLPI via Configuration > Authentification > SAML. Le paramètre COMPRESS REQUESTS est critique : le plugin utilise la compression deflate pour les requêtes SAML en Redirect Binding (GET). Si ce paramètre est désactivé, Keycloak reçoit une requête non compressée qu'il ne peut pas décoder.

Paramètre GLPI	Valeur
Entity ID IdP	https://keycloak.nym-it.local:8449/realms/master
SSO URL	https://keycloak.nym-it.local:8449/realms/master/protocol/saml
COMPRESS REQUESTS	ON (provoque Zip Exception si OFF)
COMPRESS RESPONSES	OFF
JIT USER CREATION	ON

6.5 Incidents lies a GLPI

L'intégration GLPI a généré six incidents majeurs, chacun ayant nécessité une investigation approfondie, souvent directement en base de donnees MariaDB ou dans les logs Keycloak.

GLPI-01 — Chemin du plugin incorrect

Symptôme	Plugin introuvable dans /var/www/glpi/plugins — dossier vide.
Cause racine	L'image Docker glpi/glpi:latest ne place pas les donnees dans /var/www/glpi/ comme la documentation standard le suggère. Le bon chemin est /var/glpi/marketplace/.
Résolution	Identification du bon chemin via find / -name 'glpi' -type d, puis installation dans /var/glpi/marketplace/samlsso.

GLPI-02 — Attributs SAML en doublon

Symptôme	Erreur 'Found an Attribute element with duplicated Name' dans Keycloak.
Cause racine	Les scopes SAML globaux (role_list, saml_organization) envoyaient des attributs en doublon avec les mappers du scope dédié du client.
Résolution	Suppression des scopes role_list et saml_organization dans le Setup du client. Seul le scope dédié est conserve.
Prévention	Vérifier les scopes par défaut attaché à chaque nouveau client SAML.

GLPI-03 — Loginstate corrompu

Symptôme	Erreur 'GLPI did not expect an assertion from this IdP — race condition'.
Cause racine	Des entrées de loginstate corrompues ou obsolètes dans la table MariaDB glpi_plugin_samlsso_loginstates empêchent GLPI de corréler la requête SAML avec la réponse.
Résolution	Purge : DELETE FROM glpi_plugin_samlsso_loginstates ; — Mise en place d'un cron toutes les minutes pour automatiser le nettoyage permanent.
Prévention	Cron actif en permanence.

GLPI-06 — Compression SAML

Symptôme	Erreur 'ZipException: invalid code lengths set' dans Keycloak.
Cause racine	Le paramètre COMPRESS REQUESTS était désactivé en base MariaDB alors que le plugin utilise la compression deflate pour les requêtes GET.
Résolution	Activation de COMPRESS REQUESTS en base : UPDATE glpi_configs SET value='1' WHERE name='saml_compress_requests'.
Prévention	Vérifier ce paramètre après toute mise à jour de GLPI.

GLPI-07 — Client ID (trailing slash)

Symptôme	Erreur 'client_not_found — Cannot_match_source_hash' dans Keycloak.
Cause racine	GLPI ajoute automatiquement un slash final à son Entity ID (Issue), mais le Client ID Keycloak n'en avait pas.
Résolution	Correction : https://glpi.nym-it.local/ (avec le slash final) dans Keycloak.
Prévention	Spécificité non documentée — toujours vérifier le trailing slash pour GLPI.

GLPI-08 — Healthcheck MariaDB

Symptôme	Healthcheck échoué: 'mysqladmin: not found'. GLPI ne démarré pas.
Cause racine	MariaDB 12 a supprimé la commande mysqladmin. Le healthcheck Docker utilisait cette commande.
Résolution	Remplacement par : test: ['CMD', 'healthcheck.sh', '--connect', '--innodb_initialized'] avec start_period : 30s.
Prévention	Adapter les healthchecks à chaque version d'image Docker.

7. Portail SSO NYM-IT

7.1 Architecture et choix techniques

Le portail SSO est une application web légère en HTML, CSS et JavaScript, conçue pour offrir aux utilisateurs un point d'entrée unique vers l'ensemble des applications NYM-IT. Le choix d'une application JavaScript client-side (plutôt qu'une application serveur) simplifie le déploiement : le portail est un ensemble de fichiers statiques servis par Nginx.

Le portail utilise le protocole OIDC (Authorization Code Flow) pour authentifier l'utilisateur auprès de Keycloak, puis propose des liens IdP-initiated SAML vers les applications. Le token d'accès est stocke en session Storage, ce qui garantit sa suppression automatique à la fermeture de l'onglet.

7.2 Client OIDC dans Keycloak

Le client OIDC est configure en mode 'public' (sans secret client) car il s'agit d'une application JavaScript qui s'exécuté dans le navigateur — il est techniquement impossible de stocker un secret de manière sécurisée côté client.

Paramètre	Valeur
Client ID	portal-nym-it
Client type	OpenID Connect (public)
Root URL / Valid redirect URIs	http://portal.nym-it.local/*
Web origins	http://portal.nym-it.local

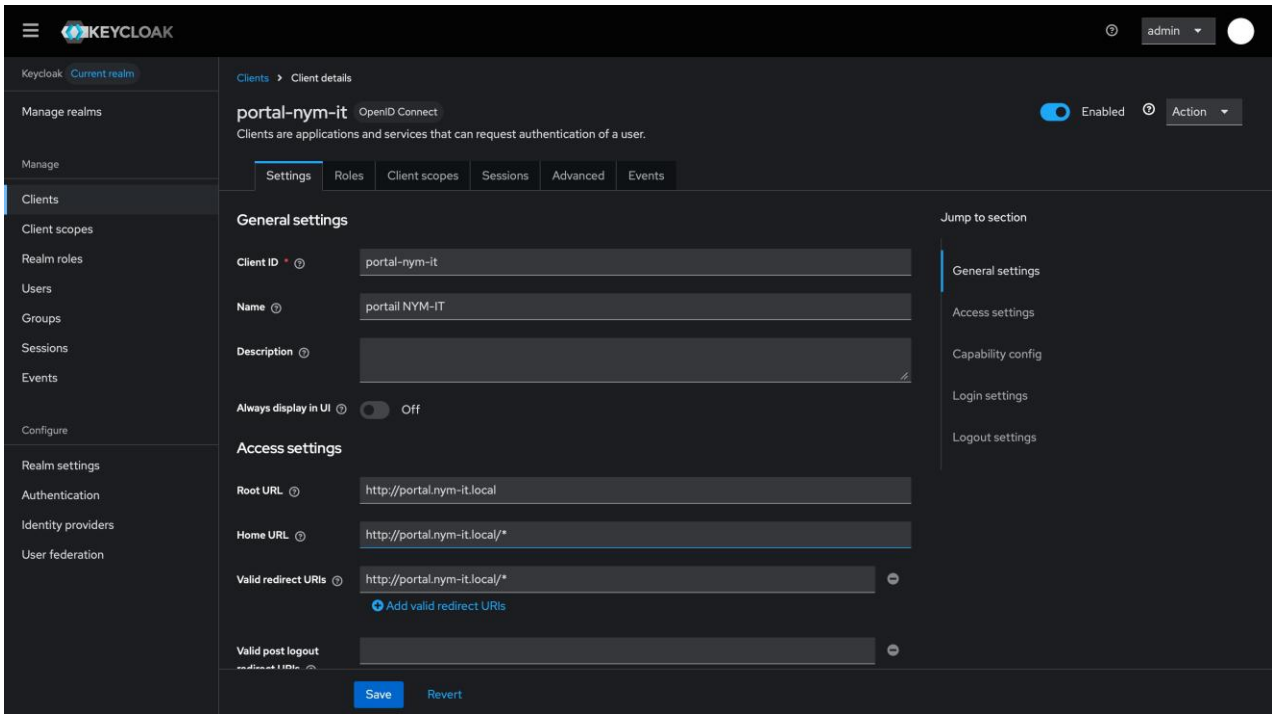

Fig. *Configuration du client OIDC portal-nym-it dans Keycloak*

7.3 Liens IdP-initiated SSO

Les liens vers les applications utilisent le mécanisme IdP-initiated SSO de Keycloak. Lorsque l'utilisateur clique sur un lien, Keycloak généré directement une assertion SAML et la transmet à l'application cible, sans que celle-ci n'ait a initier le flux. Ce mécanisme fonctionne grâce au champ 'IDP-Initiated SSO URL name' configure dans chaque client SAML (valeurs 'nextcloud' et 'glpi').

Application	URL IdP-initiated
Nextcloud	https://keycloak.nym-it.local:8449/realms/master/protocol/saml/clients/nextcloud
GLPI	https://keycloak.nym-it.local:8449/realms/master/protocol/saml/clients/glpi

7.4 Incidents lies au portail

PORTAL-01 — URLs IdP-initiated encodées

Symptôme	Erreur 'client_not_found' lors du clic sur les liens Nextcloud/GLPI.
Cause racine	Les URLs utilisaient le Client ID complet encode en URL (%3A%2F%2F...) au lieu du SSO URL name simplifie.
Résolution	Configuration du champ IDP-Initiated SSO URL name ('nextcloud', 'glpi') et mise a jour des URLs du portail.

PORTAL-02 — Redirect URI invalide

Symptôme	Erreur 'invalid_redirect_uri' lors de l'authentification OIDC.
Cause racine	Le Valid redirect URI dans Keycloak ne contenait pas le wildçard /*. Après authentification, l'URL de retour incluait un paramètre ?code=xxx qui ne matchait pas.
Résolution	Ajout du wildçard /* dans les Valid redirect URIs.
Prévention	Toujours utiliser un wildçard /* pour autoriser les query strings dans les URIs de retour.

8. Supervision et monitoring

La supervision de l'infrastructure SSO est critique : une panne de Keycloak impacte l'ensemble des applications car c'est le point central d'authentification. Cette section détaillé les différents niveaux de supervision mis en place.

8.1 Monitoring via Portainer

Portainer offre une interface graphique pour surveiller en temps réel les conteneurs Docker. Pour les conteneurs Keycloak et PostgreSQL, on peut visualiser les statistiques de ressources (CPU, mémoire, réseau, I/O), les processus actifs et les logs applicatifs. Ces informations sont accessibles via Containers > keycloak_app > Stats et Logs.

8.1.1 Statistiques de ressources

Le conteneur Keycloak consomme environ 600 MB de RAM en fonctionnement normal. Le processus principal est une JVM Java exécutant Keycloak sur Quarkus. La surveillance de la mémoire est importante car un pic peut indiquer un problème (trop de sessions actives, requêtes en bouclé).

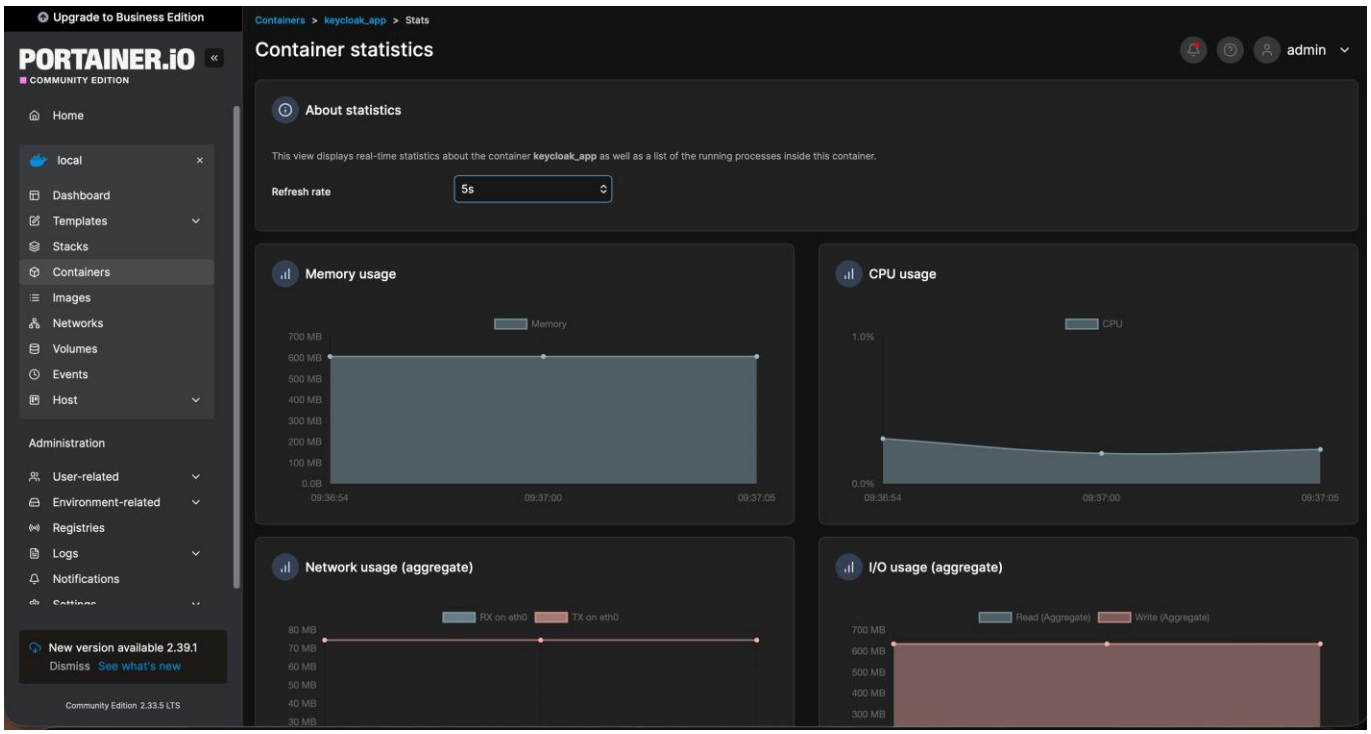

Fig. *Portainer — Statistiques keycloak_app (Memory ~600MB, CPU)*

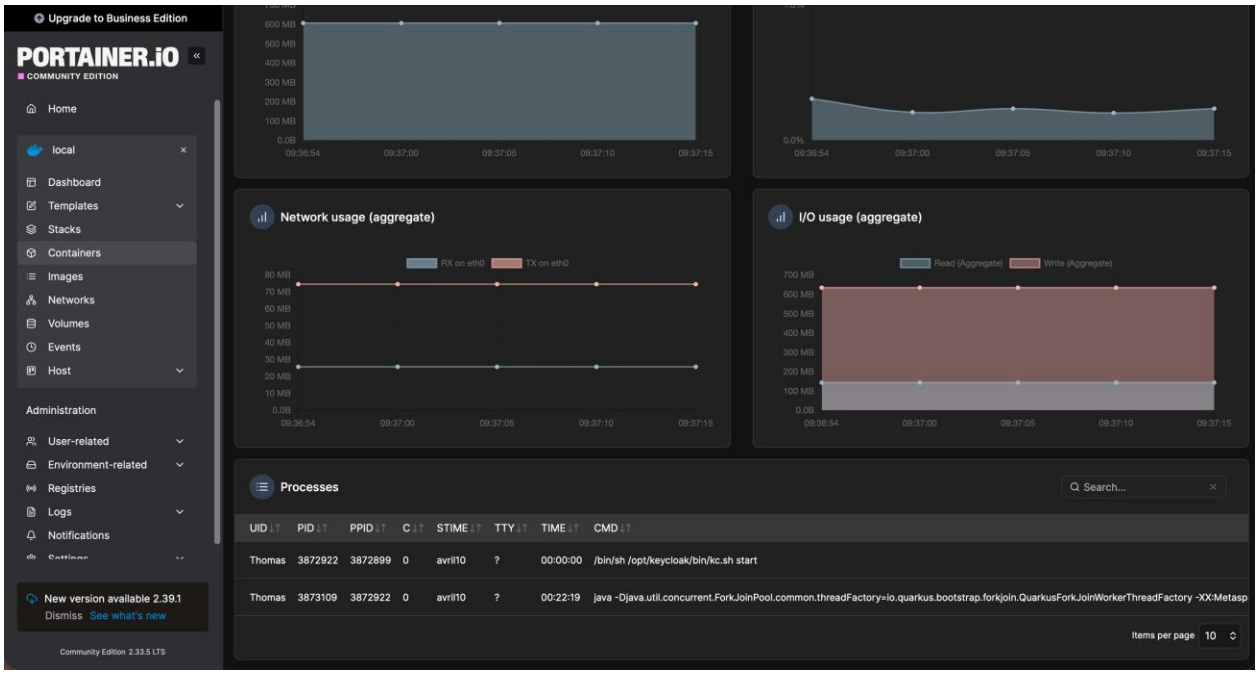

Fig. *Portainer — Network, I/O et processus Java actifs*

8.1.2 Logs des conteneurs

Les logs Keycloak permettent d'identifier les erreurs d'authentification, les problèmes de fédération LDAP et les erreurs internes. Les logs PostgreSQL montrent les checkpoints, les phases de recovery et le statut de la base.

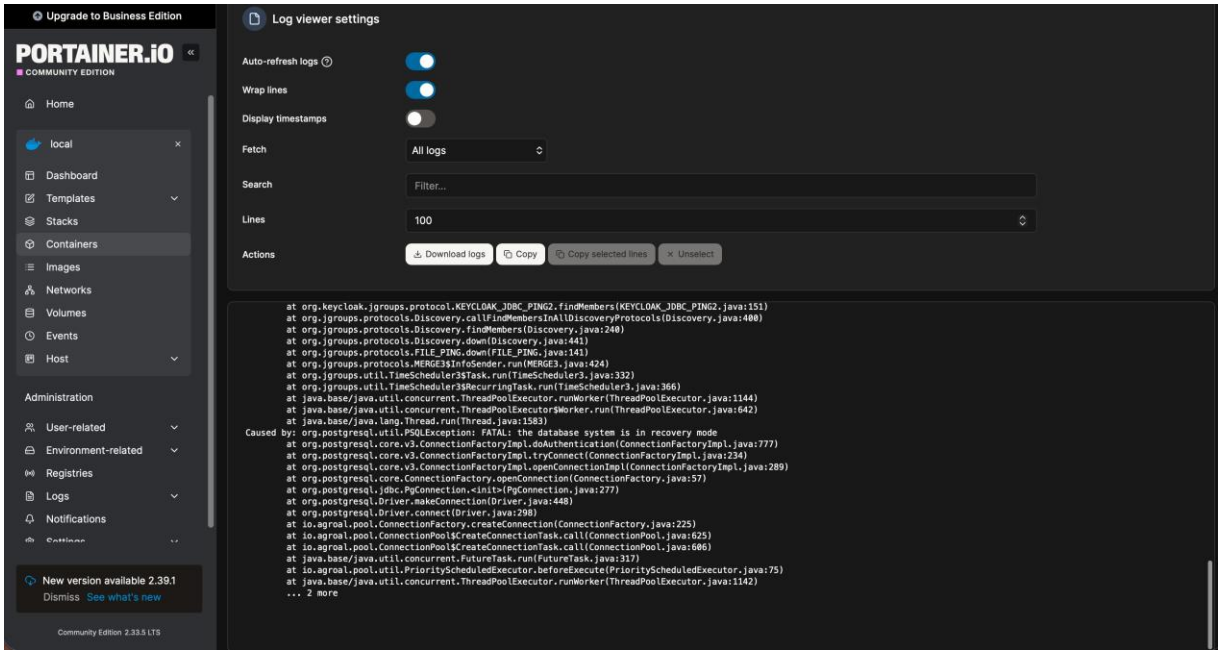

Fig. *Logs keycloak_app — Stack trace JGroups/PostgreSQL*

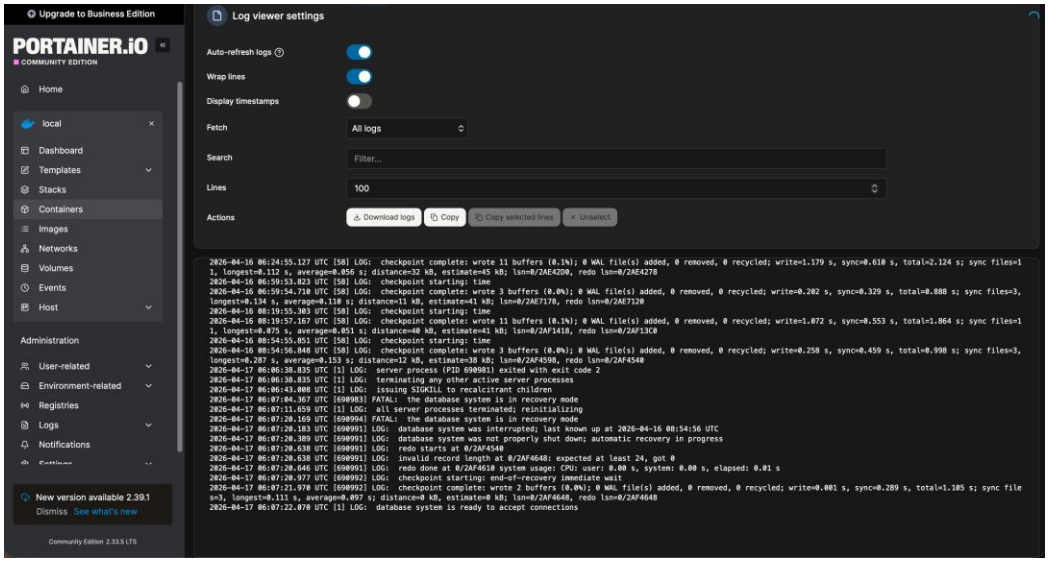

Fig. *Logs keycloak_postgres — Recovery et 'ready to accept connections'*

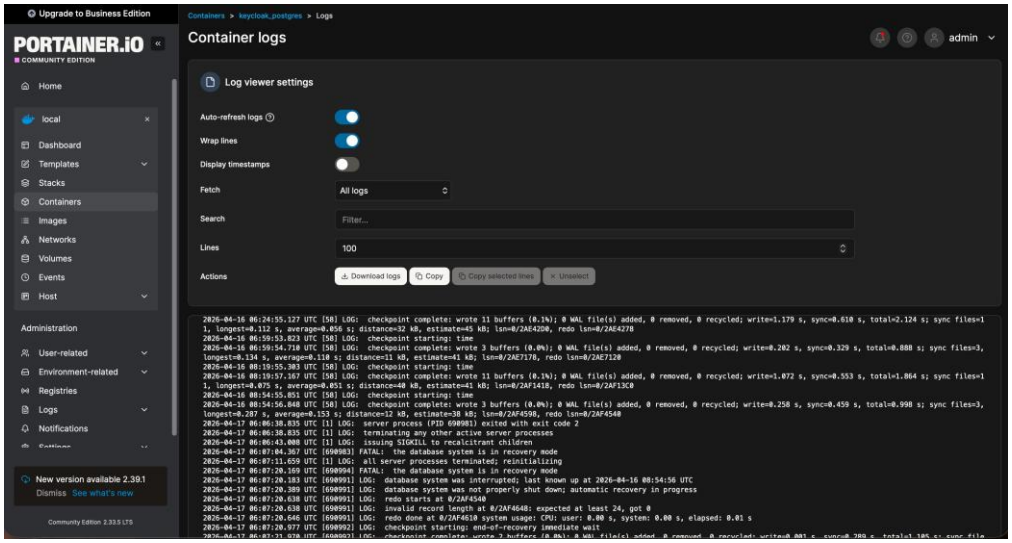

Fig. *Logs keycloak_postgres — Checkpoints et recovery complète*

8.2 Script de monitoring automatique

Un script bash vérifié la disponibilité de Keycloak toutes les 5 minutes via l'Endpoint de sante /health/ready. Si Keycloak ne répond pas, le script redémarrage automatiquement le conteneur et journalise l'événement. Ce mécanisme de self-healing garantit une remise en service rapide.

```
#!/bin/bash
KEYCLOAK_URL="https://keycloak.nym-it.local:8449/health/ready"
STATUS=$(curl -sk -o /dev/null -w "%{http_code}" --max-time 10 "$KEYCLOAK_URL")
if [ "$STATUS" = "200" ]; then
    echo "[$(date)] OK" >> /var/log/keycloak_monitor.log
else
    echo "[$(date)] ALERTE (HTTP $STATUS)" >> /var/log/keycloak_monitor.log
    docker restart keycloak_app
fi
```

Cron : */5 * * * * root /opt/scripts/monitor_keycloak.sh

8.3 Nettoyage automatique des loginstate GLPI

Le cron de nettoyage des loginstates GLPI (incident GLPI-03) s'exécute chaque minute pour purger les entrées corrompues :

```
* * * * * root docker exec glpi-db-1 mariadb -u glpi -pglpi glpi \
    -e 'DELETE FROM glpi_plugin_samlsso_loginstates;'
```

8.4 Commandes de diagnostic

Action	Commande
Logs Keycloak	docker logs keycloak_app --tail 50 -f
Sante Keycloak	curl -sk https://keycloak.nym-it.local:8449/health/ready
État conteneurs	docker compose ps
Test LDAP	ldapsearch -x -H ldap://192.168.1.1 -D CN=Sync_keycloack,... -W
Vérifier certificat	openssl s_client -connect keycloak.nym-it.local:8449
Test DNS	nslookup keycloak.nym-it.local
Sessions actives	Keycloak > Administration > Sessions
Events auth	Keycloak > Administration > Events

9. Sécurité

9.1 Gestion des certificats

L'infrastructure utilise des certificats auto-signes générés avec openssl. Chaque service dispose de son propre certificat. En production, il serait recommandé de migrer vers une PKI interne ou des certificats Let's Encrypt.

Aspect	Implémentation actuelle	Recommandation production
Type de certificat	Auto-signe par service	PKI interne ou Let's Encrypt
Distribution	Copie manuelle dans les trusts stores	GPO ou déploiement automatise
Vérification	openssl x509 -text -noout	Monitoring d'expiration avec alerte

9.2 Bonnes pratiques appliquées

17. Credentials administrateur stockes dans un fichier .env non versionne, jamais en clair dans docker-compose.yml
18. Compte de service LDAP (Sync_keycloack) avec droits READ_ONLY uniquement
19. Durée de vie des tokens limitée, refresh token active
20. Communications Docker internes en HTTP (réseau isole), HTTPS obligatoire en sortie
21. Segmentation réseau : services Docker sur 192.168.20.0/24, isoles du LAN principal

9.3 Utilisateurs externes

Keycloak permet l'ajout d'utilisateurs externes sans les intégrer au domaine AD. Des comptes peuvent être créés localement dans Keycloak avec des rôles et groupes spécifiques. A terme, des Identity Providers externes (Google, Microsoft) pourraient être configures pour les partenaires.

10. Bilan et compétences

10.1 Résultats

Objectif	Statut	Notes
Déploiement Keycloak Docker	Réalisé	Keycloak 26.5.2 + PostgreSQL 17
Fédération Active Directory	Réalisé	14 utilisateurs synchronisés
SSO SAML Nextcloud	Réalisé	Nextcloud 33 + plugin user_saml
SSO SAML GLPI	Réalisé	GLPI 11 + plugin samlSSO 1.2.5
Portail SSO unifie	Réalisé	OIDC + liens IdP-initiated SAML
Supervision	Réalisé	Portainer + script monitoring + cron

10.2 Compétences BTS SIO SISR

22. Administrer les systèmes : déploiement Docker, configuration Nginx, gestion des conteneurs et des services Linux
23. Gérer les identités : fédération LDAP/AD, SSO SAML 2.0 et OIDC, gestion centralisée des comptes
24. Sécuriser les accès : certificats SSL, reverse proxy HTTPS, contrôle d'accès centralisé, séparation des réseaux
25. Diagnostiquer et résoudre : résolution de 19 incidents techniques, analyse de logs, investigation en base de donnees
26. Superviser : monitoring automatique avec self-healing, surveillance Portainer, maintenance par cron
27. Documenter : documentation technique complète avec captures d'écran et procédures de résolution

10.3 Perspectives

28. Migration vers des certificats Let's Encrypt ou PKI interne
29. Création d'un realm dédié (hors master) pour isoler les configurations
30. Intégration de l'authentification multi-facteurs (MFA/2FA) via Keycloak
31. Extension du SSO a d'autres services (UrBackup, Portainer)
32. Configuration d'Identity Providers externes (Google, Microsoft) pour les partenaires

11. Glossaire

Ce glossaire définit les termes techniques utilises dans cette documentation.

Protocoles

Terme	Définition
SSO	Single Sign-On : mécanisme d'authentification unique pour plusieurs applications.
SAML 2.0	Security Assertion Markup Language : standard XML d'échange d'assertions d'authentification entre un IdP et des SP.
OIDC	OpenID Connect : protocole d'identité construit sur OAuth 2.0, utilisant des tokens JWT.
LDAP	Lightweight Directory Access Protocol : protocole d'interrogation d'annuaires.
OAuth 2.0	Protocole d'autorisation (base d'OIDC).

Acteurs et concepts SSO

Terme	Définition
IdP	Identity Provider : serveur central d'authentifiçation (Keycloak dans ce projet).
SP	Service Provider : application qui délégué l'authentifiçation a l'IdP (Nextcloud, GLPI).
Assertion SAML	Document XML signe contenant les informations de l'utilisateur authentifie.
Realm	Espace d'administration Keycloak regroupant utilisateurs, clients et configurations.
Client SAML/OIDC	Représentation d'une application dans Keycloak avec ses URLs, certificats et mappers.
ACS URL	Assertion Consumer Service URL : endpoint du SP ou Keycloak envoie la réponse SAML.
Entity ID	Identifiant unique d'un acteur SAML (doit correspondre exactement entre IdP et SP).
SP-initiated SSO	L'utilisateur accédé au SP, qui redirige vers l'IdP pour authentifiçation.
IdP-initiated SSO	L'utilisateur s'authentifie d'abord sur l'IdP, puis accédé au SP directement.

Terme	Définition
JIT User Création	Création automatique de compte dans l'application lors du premier login SSO.
sAMAccountName	Attribut AD : nom de connexion court (ex: yanis.rjiba).

12. Annexes

Annexe A — docker-compose.yml Keycloak

Voir captures section 3.1 pour le contenu complet.

Annexe B — Configuration Nginx

Voir captures section 3.3 pour les Virtual hosts.

Annexe C — Sources et références

33. Documentation Keycloak : https://www.keycloak.org/documentation
34. Plugin samlSSO GLPI: https://github.com/derricksmith/phpsaml
35. Nextcloud SAML: https://docs.nextcloud.com
36. Docker Hub Keycloak: https://hub.docker.com/r/keycloak/keycloak
37. Nginx: https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/