

JoSy S3 - 7 & 8 octobre 2025
Strasbourg - IPHC

RETEX-S3 MinIO

NFS pour calculer, S3 pour partager : moderniser la diffusion des résultats scientifiques



[Anne-Sophie LEDOUX](#) | [UMR 1283/8199 EGENODIA EGID PRECIDIAB](#)



Sommaire

1. **Présentation : unité, équipe informatique, infrastructure**
2. **Contexte, besoins et limites**
3. **Ecosystème local/national**
4. **Pourquoi Objet ? S3 ? MinIO ?**
5. **Architecture déployée**
6. **Workflow collaboratif**
7. **Automatisation, installation**
8. **Limites MinIO Community Edition**
9. **Bénéfices et limites du déploiement labo**
10. **Perspectives**

L'unité

UMR8199 - (Epi)génomique fonctionnelle métabolique et des dysfonctions dans le diabète de type 2 et des maladies associées

- ▶ **Domaine : recherche en génomique et métabolomique appliquée au diabète et à l'obésité**
- ▶ **Membre fondateur de l'Institut européen sur le diabète et l'obésité (EGID)**
 - 5 UMR
 - 6 plateformes technologiques
- ▶ **Partenaire du centre national de médecine de précision du diabète - PreciDIAB**
- ▶ **Laboratoire de biologie médicale accrédité COFRAC ISO 15189**



L'équipe informatique

Equipe IT infrastructure & data security - 2 personnes (+ stagiaires)

► Missions principales :

- **Gestion et sécurisation** des infrastructures IT (serveurs, stockage, réseaux, virtualisation)
- **Administration** systèmes et réseaux, gestion des accès et supervision
- **Mise en œuvre et suivi du processus informatique** (qualité, conformité ISO 15189)
- **Cybersécurité** : protection des données, surveillance des incidents, conformité SSI
- **Support technique** aux utilisateurs et aux plateformes scientifiques
- **Pilotage et accompagnement des projets IT stratégiques**
- **Accueil et encadrement de stagiaires** (assistance utilisateurs, projets spécifiques SI)

L'infrastructure

- ▶ 500 utilisateurs
- ▶ env. 7 Po de données
- ▶ 150 serveurs physiques & virtuels
- ▶ Cluster HPC Calcul intensif : 12 GPU - 2000 CPU - 10 To RAM
- ▶ Salle serveur dédiée (hébergement local, maîtrise complète des données et de la sécurité)
- ▶ PRA hébergé à la DGDNUM - Université de Lille (1,2 Po de réplication)

Contexte

- ▶ **Infrastructure mutualisée** au service des plateformes génomique (NGS) & métabolomique



- ▶ **Stockage principal** : baie PowerScale (NFS) pour workflows de calcul & analyses

- ▶ **Contraintes fortes** :

- RGPD (données personnelles/santé)
- ISO 15189 (biologie médicale)
- Sécurité & traçabilité renforcées

Besoins des utilisateurs

- ▶ **Résultats massifs** : plusieurs To (ex. génomes)
- ▶ **Diffusion externe** : collaborateurs, cliniciens, partenaires industriels
- ▶ **Limitations actuelles** :
 - NFS non adapté à l'externe
 - F*EX obsolète, non sécurisé, non conforme
- ▶ **Besoins clés** :
 - Partage simple et ergonomique
 - Sécurisé et conforme aux normes
 - Sans exposer directement le stockage interne

Limites de F*EX (ancien service)

- ▶ Service d'échange de fichiers volumineux (File Exchange) - 2014
- ▶ Dépôt temporaire
- ▶ Lien de téléchargement généré par les bioinfos et envoyé par mail
- ▶ Service vieillissant & non sécurisé
- ▶ Limites :
 - Pas de chiffrement
 - Pas de traçabilité
 - Pas conforme RGPD
 - Pas ergonomique

F*EX - Frams' Fast File EXchange



Ecosystème local - Nextcloud

Cloud souverain du labo

▶ Avantages :

- Interface conviviale, synchronisation multi-plateformes
- Bon pour les documents bureautiques et petits fichiers
- Déploiement et maîtrise locale

▶ Limites :

- Peu adapté aux gros volumes bioinfo
- Performance limitée pour le calcul scientifique
- Pas pensé pour des centaines de To/Po

Ecosystème national - FileSender

RENATER FileSender

▶ Avantages :

- Partage simple et sécurisé de fichiers volumineux
- Service hébergé par la communauté ESR

▶ Limites : 100 à 150 Go max par fichier

- Durée max : 30 jours (fichiers supprimés ensuite)
- Pas d'intégration directe dans les workflows (pas d'API S3, pas d'automatisation)
- Usage limité au transfert temporaire, pas un stockage pérenne

Pourquoi penser au stockage objet ?

- ▶ Besoin de **gérer des volumes massifs** (plusieurs To)
- ▶ Limites des solutions existantes : **F*EX, Nextcloud, FileSender**
- ▶ Nécessité d'un **accès automatisable** (API, intégration pipelines, scripts)
- ▶ Enjeux de **diffusion sécurisée et maîtrisée** (partenaires, collaborations)
- ▶ Volonté de **séparer les usages** :
 - NFS = calcul interne (workflows, HPC)
 - Autre : S3 ? = diffusion externe

Qu'est-ce-que le stockage objet ?

- ▶ **Stockage objet** (par opposition au bloc/fichier)
- ▶ Organisation en **buckets** (équivalent de dossiers)
- ▶ Chaque **objet** = fichier + métadonnées + identifiant unique
- ▶ **Protocoles principaux :**
 - S3 (AWS, standard de facto, utilisé par MinIO, Ceph, Scality, Cloudian...)
 - Swift (OpenStack)
 - Azure Blob Storage API
 - Google Cloud Storage API
 - CDMI - Cloud Data Management Interface (standard SNIA - Storage Networking Industry Association, peu implémenté)

Dans la pratique ...

- ▶ **Surcoût d'espace** : le stockage objet occupe ~10 à 15 % de plus qu'un stockage fichier équivalent (métadonnées + indexation)

200 à 300 Go pour 2 To !!

- ▶ **Exemple labo :**

- Bucket = ligan
- Objet = ligan/projetX/D1834.tar

Name:

D1834.tar

Size:

112.6 GiB

Last Modified:

3 days ago

ETAG:

cb2492ca3ac90dd594995e497f81a5fc-7208

Pourquoi S3 s'est imposé ?

- ▶ **Première API objet simple et RESTful (2006)** → adoption massive dès le départ
- ▶ **Interopérabilité** : implémentée par presque tous les acteurs (MinIO, Ceph, Scality, Dell ECS...)
- ▶ **Ecosystème d'outils riche** : mc, rclone, Cyberduck, SDKs (Python boto3, Go, Java...)
- ▶ **Adoption par les clouds publics** : Azure et Google ont dû ajouter une compatibilité S3
- ▶ **Autres protocoles en retrait** : Swift en déclin, CDMI resté théorique, APIs propriétaires cantonnées à leurs écosystèmes
- ▶ **Résultat** : *Aujourd'hui, parler stockage objet = parler S3*

Pourquoi S3 au labo ?

- ▶ **Intégration facilité** dans les pipelines bioinfo
- ▶ **Auto-hébergement indispensable** : maîtrise du SI, conformité pour données médicales sensibles
- ▶ **Alignement ISO 15189** : sécurité, traçabilité, exigences réglementaires des labos biomédicaux
- ▶ **Séparation claire des usages** :
 - NFS = calcul interne (workflows, HPC)
 - S3 = diffusion externe (partage, collaborations)

Pourquoi pas S3 PowerScale ?

Fonctionnalité S3 disponible sur PowerScale (via OneFS) sans ajouts de nœuds dédiés, mais plusieurs points de vigilance :

- ▶ **Sécurité** : surface d'attaque supplémentaire, gestion des clés limitée, droits moins fins qu'en NFS/SMB.
- ▶ **Risque d'ouverture trop large** : peu adapté à un service collaboratif finement découpé
- ▶ **Usage** : conçu pour un stockage centralisé de gros volumes hétérogènes, peu adapté aux petits fichiers ou à un service agile transversal.
- ▶ **Impact** : risque sur les performances et la consommation si usage massif.
- ▶ **Support** : gestion SSL/TLS, compatibilité partielle S3, fonctions avancées limitées.

Solutions S3 institutionnelles FR

▶ S3NS (Thales + Google Cloud)

- Stockage objet opéré en France
- Objectif : certification SecNumCloud (ANSSI) - été 2025

▶ OVHcloud - Object Storage 3AZ

- Compatible API S3, haute résilience (3 zones de dispo)
- Certifications : HDS (santé), ISO 27001

▶ Leviia - S3 souverain

- Start-up française, stockage objet basé en France
- Certifié HDS, conforme RGPD

Choix de l'auto-hébergement pour garder la maîtrise totale des données sensibles (santé, génomique, métabolomique).

Pourquoi MinIO ?

- ▶ **Compatibilité totale avec l'API AWS S3**
- ▶ **Solution open source** et déployable en VM (pas besoin de cluster lourd)
- ▶ **Léger et performant**, adapté aux environnements de recherche
- ▶ **Auto-hébergement** → maîtrise locale des données sensibles
- ▶ **Communauté active**, mises à jour régulières
- ▶ Comparé à d'autres **solutions S3 ou équivalentes** :
 - **Ceph (RGW S3)** : puissant mais complexe, demande une infra dédiée lourde
 - **OpenStack Swift (API Swift)** : robuste mais lourd à administrer
 - **AWS S3** : service externe, impossible pour données médicales sensibles
 - **Wasabi / Backblaze** : cloud public, non conforme RGPD/ISO 15189

MinIO : une success story open source

2014:Création de MinIO Inc. par **Anand Babu Periasamy** (ex-GlusterFS) → objectif : fournir un **stockage objet open source** simple et compatible **API Amazon S3**.

2015-2016 : Premières versions open source → adoptées massivement pour des projets **cloud-native** et **Kubernetes**.

2018-2019 : MinIO devient l'une des solutions S3 les plus téléchargées sur GitHub (plus de **250M de pulls Docker**).

2020 : MinIO remplace son ancienne licence Apache 2.0 par **AGPLv3**, pour freiner l'utilisation non contributive par les hyperscalers.

2021-2022 : Croissance forte dans l'**HPC**, l'**IA** et les environnements Big Data ; adoption par de grandes entreprises (VMware, Nutanix, etc.)

... mais un modèle commercial qui déçoit

2023 : Déploiement au laboratoire UMR8199 (*nectar, VM Proxmox, 5 To PowerScale, usage bioinfo/métabo*)

2023-2024 : Évolution commerciale → rétrogradation de fonctionnalités dans la Community Edition (OIDC, LDAP+locaux) au profit de l'Enterprise Edition

2025 : Accès STS (credentials temporaires) retiré des dernières versions Community

Architecture déployée - VM

Ressources VM *nectar* sur un cluster Proxmox DMZ

- ▶ OS : Debian 12
- ▶ CPU / RAM : 8 vCPU - 16 Go RAM
- ▶ Disques :
 - /dev/sdb1 - 50 Go → / (system / config MinIO)
 - /dev/sda - 200 Go → /srv/s3 (cache / logs applicatifs)
 - /dev/sdc - 5 To → /srv/s3/data (stockage objet)
- ▶ Création des disques :
 - via **storage NFS** monté sur le cluster Proxmox DMZ
 - NFS provisionné depuis la baie Isilon PowerScale
 - choix lié au **manque de stockage local** sur les serveurs physiques Proxmox

Architecture déployée - VM

The screenshot displays the Proxmox Virtual Environment (VE) 8.1.4 interface. The left sidebar shows a tree view of the Datacenter (dmz-egid) with several virtual machines (VMs) listed, including '107 (nectar)' which is currently selected. The main panel shows the configuration for 'Virtual Machine 107 (nectar) on node 'dmz-hyp07''. The configuration is organized into a left-hand menu and a right-hand table.

Left-hand menu:

- Summary
- Console
- Hardware
- Cloud-Init
- Options
- Task History
- Monitor
- Backup
- Replication
- Snapshots
- Firewall
 - Options
- Alias
- IPSet
- Log
- Permissions

Right-hand table:

Add Remove Edit Disk Action Revert		
Memory		16.04 GiB
Processors		8 (1 sockets, 8 cores) [x86-64-v2-AES]
BIOS		Default (SeaBIOS)
Display		Default
Machine		Default (i440fx)
SCSI Controller		VirtIO SCSI single
CD/DVD Drive (ide2)		Dmzprox:iso/debian-12.11.0-amd64-netinst.iso,media=cdrom,size=670M
Hard Disk (scsi0)		Dmzprox:107/vm-107-disk-0.qcow2,cache=writeback,discard=on,iothread=1,size=50G
Hard Disk (scsi1)		Dmzprox:107/vm-107-disk-1.qcow2,cache=writeback,discard=on,iothread=1,size=200G
Hard Disk (scsi2)		Dmzprox:107/vm-107-disk-2.qcow2,cache=writeback,discard=on,iothread=1,size=5T
Network Device (net0)		e1000=BC:24:11:4D:E6:74,bridge=vibr0
Network Device (net1)		e1000=BC:24:11:C8:A4:C2,bridge=vibr1

Docker vs VM dédiée : un choix assumé

Raisons du choix VM + binaire systemd :

▶ Simplicité & stabilité

- Pas de couche supplémentaire (Docker) à maintenir
- Cycle de vie maîtrisé (upgrade, logs, monitoring systemd)

▶ Traçabilité & conformité ISO 15189

- Auditabilité claire du service
- Moins d'intermédiaires techniques à justifier

Un besoin de maîtrise et de robustesse

▶ Contrôle complet

- Certificat TLS officiel (egid.fr)
- Gestion explicite des chemins de stockage (/srv/s3)
- Scripts maison intégrés

▶ Production sensible

- Données médicales → nécessité de réduire les couches d'abstraction
- VM dédiée = isolement clair et robuste

▶ Image Docker adaptée pour tester ou faire un POC rapide

Services déployés

- ▶ **MinIO Community Edition RELEASE.2025-04-22T22-12-26Z**
 - API S3 → port 9000 (HTTPS)
 - Console web → port 9001 (HTTPS)
- ▶ **TLS activé avec un certificat officiel egid.fr**
 - Certificat délivré par l'autorité de certification interne
 - MinIO utilise ce certificat (pas d'auto-signé, pas de reverse proxy)
- ▶ **Organisation en buckets par plateforme (ligan, impact-pm)**

Architecture déployée - Limites & Ressources

Limite technique à signaler

- ▶ Stockage actuel basé sur **NFS distant (PowerScale Isilon)**
 - bon compromis de départ
 - l'idéal serait un stockage interne dédié au S3, pour de meilleures performances et résilience

Ressources utiles

- ▶ Dépôt officiel
 - <https://github.com/minio/minio>
- ▶ Site officiel MinIO
 - <https://min.io/>

Architecture actuelle - choix transitoire

Contexte

- ▶ L'instance **MinIO (VM Nectar)** repose actuellement sur un **datastore NFS distant** (hébergé sur le cluster PowerScale / Isilon)
- ▶ Les objets S3 sont donc physiquement stockés sur un **backend NFS**, bien que servis via l'API S3 MinIO
- ▶ Ce choix permettait de **valider la solution** sans modifier l'infrastructure existante

Avantages du choix NFS

- ▶ Réutilisation du **stockage mutualisé existant**, sans investissement matériel.
- ▶ **Simplicité de mise en œuvre** pour la phase de test / POC.
- ▶ **Pas de perte de données** : intégration directe avec les sauvegardes et snapshots PowerScale.

Architecture actuelle - choix transitoire

Limites identifiées

- ▶ **Dépendance** à la couche NFS → possible goulot d'étranglement en I/O.
- ▶ **Moins de résilience** que des disques locaux ou un cluster objet natif.
- ▶ **Exposition indirecte** du stockage interne via la VM.

Évolution prévue

- ▶ **Migration vers un stockage interne dédié à S3 :**
 - disques SSD/NVMe ou volumes Ceph,
 - performances accrues et meilleure isolation,
 - architecture plus cohérente (vrai backend objet).
- ▶ **Maintien d'un mode hybride possible :**
 - NFS → usage interne / calcul,
 - S3 → diffusion et collaboration.

Scripts d'administration MinIO

Deux scripts pour la gestion automatisée des comptes et droits

Script	Exécuté par	Objectif	Droits attribués	Exemple d'appel
<code>create_minio_user.sh</code>	Admin (serveur <i>Nectar</i>)	Créer un compte bioinfo interne	Accès RW sur un bucket (ex. <i>ligan</i>)	<code>sudo /opt/minio-admin/scripts/create_minio_user.sh sgressier MdP ligan</code>
<code>create_collab_user.sh</code>	Bioinfo (VM <i>custom</i>) via sudo délégué	Créer un compte collaborateur temporaire	Accès RO sur <code>results/<projet></code> pendant 30 jours	<code>ssh sgressier@nectar sudo /opt/minio-admin/scripts/create_collab_user.sh chu-liege ligan results/chu-liege 30</code>

Journalisation & cycle de vie

▶ Journalisation automatique

- Toutes les actions sont tracées dans `/opt/minio-admin/logs/actions.log`
- Chaque événement est horodaté : création, policy, attribution, suppression planifiée

▶ Cycle de vie des comptes

- `create_minio_user.sh` : utilisateur interne, **persistant**
- `create_collab_user.sh` : collaborateur externe, **supprimé automatiquement après 30j**

Workflow MinIO - Gestion des résultats et partage

Acteurs et responsabilités

▶ Admin (Nectar)

- Configure MinIO et installe les scripts d'admin
- Crée comptes bioinfo
- Configure sudoers pour déléguer aux bioinfo
- Supervise logs et désactivation auto (30j)

▶ Bioinfo (Custom VM)

- Configure alias mc
- Crée son dossier projet dans ligand/results/...
- Dépose les fichiers résultats
- Lance le script collaborateur sur Nectar via SSH + sudo

Workflow MinIO - Gestion des résultats et partage

Acteurs et responsabilités

► Collaborateur (CLI / Console Web)

- Configure alias mc ou utilise console web
- Accède uniquement à `ligan/results/chu-liege/` (ReadOnly)
- Télécharge les résultats
- Compte désactivé automatiquement après 30 jours

Envoi des credentials

► Cas 1 - Accès > 7 jours

(compte collaborateur temporaire, max 30 jours)

- Création d'un compte dédié (ex. chu-liege)
 - Login transmis par email
 - Mot de passe transmis par SMS
 - Accès limité au sous-dossier, désactivation auto après 30j
- Jamais dans le même canal**

► Cas 2 - Accès ≤ 7 jours

(partage ponctuel de résultats)

- Génération d'une **URL pré-signée** avec `mc share download`
- Lien valable max 7 jours
- Pas besoin de compte ni de credentials
- Idéal pour livraison ponctuelle (ex. fichier unique ou archive compressée)

Installation du client mc - custom VM

► Installation simple (Linux) - *admin*

```
wget https://dl.min.io/client/mc/release/linux-amd64/mc -O  
/usr/local/bin/mc
```

```
chmod +x /usr/local/bin/mc
```

► Certificat TLS - *admin*

- soit installé localement (CA interne)
- soit utilisation temporaire de --insecure lors des tests internes

► Configuration de l'alias vers Nectar - *bioinfo*

```
mc alias set nectar https://nectar.egid.fr:9000 ACCESSKEY  
SECRETKEY
```

► Usage de base - *bioinfo*

```
mc cp result.vcf nectar/ligan/projetX/
```

```
mc ls nectar/ligan/projetX/
```

Création du collaborateur

Via un script shell - *bioinfo*

```
./create_collab_user.sh <collab> <bucket> <path> <duration_days>
```

▶ Automatisation des tâches *admin* :

- Création d'un compte collaborateur local
- Application d'une policy JSON (RO/RW, scope limité)
- Cycle de vie : suppression automatique après 30 jours

▶ Script développé par notre équipe pour :

- éviter d'alourdir les tâches admin,
- donner plus d'autonomie aux bioinformaticiens,
- garantir un cycle de vie maîtrisé et conforme ISO 15189.

Exemple de policy JSON

Policy lecture seule sur un répertoire spécifique

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:GetObject"],
      "Resource": ["arn:aws:s3:::ligan/projetX/*"]
    }
  ]
}
```

- ▶ Limite l'accès à **lecture seule** (GetObject) sur le répertoire ligan/projetX/.
- ▶ Autres déclinaisons possibles :
 - ▶ Lecture/écriture (s3:PutObject)
 - ▶ Scope plus large ou plus restreint (un seul fichier, un sous-dossier...)

Extrait du script

Extrait du script shell développé en interne

```
# Création d'un utilisateur local avec credentials aléatoires
ACCESSKEY=$(openssl rand -hex 12)
SECRETKEY=$(openssl rand -hex 24)

mc admin user add nectar $ACCESSKEY $SECRETKEY

# Application de la policy JSON générée
mc admin policy set nectar ${USER}_policy user=$ACCESSKEY

# Suppression programmée après 30 jours
echo "mc admin user remove nectar $ACCESSKEY" | at now +30 days
```

► Génère automatiquement :

- Credentials aléatoires générés et stockés dans un espace sécurisé
- Création et Application de la policy
- Suppression automatique de l'accès

Console web

MINIO
OBJECT STORE
AGPL LICENSE

User

- Object Browser
- Access Keys
- Documentation

Administrator

- Buckets
- Policies
- Identity
- Monitoring
- Events
- Configuration
- License

Object Browser

Filter Buckets

Name	Objects	Size	Access
impact	1	3.7 GiB	R/W
ligan	4	685.5 KiB	R/W

Fonctionnalité *Metrics*

Suivi intégré de l'instance MinIO :

- ▶ **Buckets et objets stockés** (volumétrie en direct)
- ▶ **Etat des serveurs et disques** (online/offline)
- ▶ **Usage global** (capacité utilisée en GiB)
- ▶ **Uptime de l'instance**
- ▶ **Accès rapide aux logs et audits**
- ▶ Permet aux administrateurs d'avoir une **vue simple et rapide** de l'état du service.
- ▶ **Limite Community Edition** : pas de dashboards avancés ni d'alertes (réservés à la version Enterprise).

Metrics

Metrics



Info Usage Traffic Resources

Server Information

Sync ↻

Buckets

2

Browse →

Objects

5

Reported Usage

3 GiB

Time since last Heal Activity	n/a
Time since last Scan Activity	n/a
Uptime	n/a

Servers

1 **0**

● Online ● Offline

Drives

1 **0**

● Online ● Offline

Backend type Erasure ✓

Standard storage class parity n/a

Reduced redundancy storage class parity n/a

Servers (1)

nectar.egid.fr:9000 ●

1/1 ● Drives 1/1 ● Network 1 month 20 days Up time

Version: 2025-04-22T22:12:26Z ^

Drives (1)

	Drive Name /srv/s3/data	Drive Status ● Online Drive
--	-----------------------------------	--------------------------------

Limites MinIO Community Edition

Version actuelle : RELEASE.2025-04-22T22-12-26Z

- ▶ **Pas de fédération IdP** : disparition du bouton de connexion IdP (OIDC/SAML) dans les releases récentes → plus possible de connecter une fédération externe - *donc pas de SSO (d'où recours aux scripts)*
- ▶ **LDAP exclusif** : impossible de faire cohabiter comptes LDAP/AD et comptes locaux → problématique pour les collaborateurs externes
- ▶ **Accès STS (credentials temporaires)** retiré des dernières versions Community - *STS reste disponible uniquement dans la version Enterprise avec IdP (LDAP/OIDC)*
- ▶ **Conséquence** : obligation de fonctionner en comptes locaux, avec un script maison pour la gestion des accès externes
- ▶ **Monitoring limité**

Politique commerciale MinIO

- ▶ MinIO a tendance à **rérograder certaines fonctionnalités** de la Community Edition au fil des versions (ex. OIDC, LDAP+locaux, STS)
- ▶ Volonté de pousser vers l'offre **Enterprise \$\$\$** ou le **cloud MinIO** (pay-as-you-go via partenaires comme Elestio)
- ▶ Résultat : la **Community reste utilisable, mais avec des fonctionnalités volontairement limitées.**

Bénéfices observés

- ▶ **Ergonomie** : accès simple, commandes claires (mc cp, mc ls, etc.)
- ▶ **Maîtrise locale** : service auto-hébergé, souveraineté des données
- ▶ **Sécurité** : TLS officiel egid.fr, politiques JSON fines, cycle de vie contrôlé
- ▶ **Interopérabilité** : API S3 standard, compatible outils bioinfo et clients S3
- ▶ **Autonomie bioinformaticiens** : script de gestion des accès → moins de charge admin
- ▶ **Conformité** : RGPD & ISO 15189 (traçabilité, durée de vie limitée des accès)

Limites rencontrées - labo

- ▶ **Stockage objet basé sur NFS distant (PowerScale) : compromis initial, mais peut ne pas être optimal**
- ▶ **Push des résultats encore manuel (à intégrer dans les pipelines bioinfos)**
- ▶ **Maintenance manuelle (mises à jour MinIO, scripts, policies)**
- ▶ **Scalabilité limitée → dépendance à la capacité Proxmox/NFS**
- ▶ **Pas de haute disponibilité native (instance unique nectar)**

Perspectives

- ▶ **Intégration pipelines bioinfo** (automatiser le push S3)
- ▶ **Améliorer la gestion des politiques** (templates RO/RW standardisés)
- ▶ **Utiliser un stockage interne dédié à S3**
- ▶ **Renforcer l'intégration sécurité** : logs, monitoring, dashboard Wazuh/ELK
- ▶ **Élargir les usages** : diffusion vers plus de partenaires, ouverture à d'autres plateformes
- ▶ **Anticiper un remplacement de MinIO** : explorer des alternatives libres (Ceph RGW, Garage, SeaweedFS) pour garantir la **pérennité**

Alternatives à MinIO

Solutions libres & S3-compatibles :

▶ Ceph RADOS Gateway (RGW)

- Robuste, soutenu par Red Hat et la communauté HPC
- Déploiement/maintenance lourds

▶ Garage (Deuxfleurs, FR)

- Léger, simple, adapté petites/moyennes structures
- Projet encore jeune, communauté restreinte

▶ SeaweedFS

- Flexible (stockage fichiers + objet), installation rapide
- Documentation plus limitée

▶ Zenko (Scality)

- Orienté multi-cloud, interopérabilité forte (AWS/Azure/GCP)
- Davantage pensé pour hybridation que pour usage interne

Merci

Questions ?