Récupération — Restauration — Reconstruction

Guide de reconstruction. Il décrit comment récupérer une base, un site, un conteneur ou une machine entière à partir des sauvegardes.

Le fichier `.md` de cette page : ouvrir le `.md`

Ce que cette page couvre : restaurer, dans l'ordre du plus courant au plus grave — une base de données, un site web, un conteneur, la clé de démarrage du serveur, puis la machine entière. Elle suppose que les sauvegardes existent ; leur mise en place fait l'objet de la page Sauvegarde.

Cette page est écrite pour être lue dans l'urgence. Peu de prose, les commandes en évidence, chaque section autonome. Si tu la lis un jour calme, la dernière section est celle qui compte : elle explique comment l'éprouver avant d'en avoir besoin.

Contexte : un serveur Ubuntu qui héberge des conteneurs Incus, dont un conteneur web sous ISPConfig, et un NAS qui reçoit les sauvegardes. Les adresses, noms d'hôtes et chemins sont des exemples à transposer.

Convention : chaque bloc de commandes indique la machine où il s'exécute.

Le principe qui gouverne tout

Le chemin de restauration n'est pas l'inverse du chemin de sauvegarde.

Toutes les sauvegardes sont des *tirages* : la machine de sauvegarde va chercher les données par le démon rsync. Et les modules sont en `read only = true` — le démon ne peut donc rien réécrire.

C’est délibéré : un démon en écriture permettrait d’écraser les originaux depuis le réseau avec le seul mot de passe.

Un seul module du parc est inscriptible — celui qui téléverse les photos vers la galerie. Il ne vise qu’un répertoire : ce n’est pas un chemin de restauration.

Toute restauration passe par un autre canal : `incus file push` pour un conteneur, `scp` ou `ssh` sinon.

Ne pas juger une sauvegarde à la date de ses fichiers. `rsync -a` préserve la date de la source : un fichier daté de 2022 dans une sauvegarde faite cette nuit est parfaitement normal. C’est le journal qui dit si la copie a eu lieu, pas l’horodatage.

Où sont les copies

Quatre niveaux, et ils ne servent pas aux mêmes pannes. Savoir lequel ouvrir fait gagner le plus de temps.

Niveau	Contenu	Fraîcheur	À utiliser quand
Quotidien — `/media/nas1/Backups/`	Données des services, sites, vidages de bases	La nuit dernière	Perte d’un fichier, d’une base, d’un site
Mensuel — `/media/nas1/Backups/incus-exports/`	Export complet des conteneurs, deux mois conservés	Le 1er du mois	Perte d’un conteneur entier
Hors site — Proton Drive	Copie du mensuel, plus l’image de la clé de démarrage	Continu, versions conservées 30 jours	Perte de la machine : incendie, vol, panne

Niveau	Contenu	Fraîcheur	À utiliser quand
Disques externes	nas1 et nas2, séparément	Jusqu’à deux mois	Erreur ancienne, corruption, rançongiciel — ils sont débranchés

⚠ Un disque débranché depuis longtemps peut porter d’anciens noms de répertoires. Les noms évoluent au fil des réorganisations, et un disque hors ligne ne suit pas. Chercher large avant de conclure qu’une sauvegarde manque :

```
ls -d /media/<disque>/Backups/*web-srv*
```

⚠ Le hors site conserve les versions antérieures trente jours, pas davantage. Une archive corrompue ou effacée par erreur reste récupérable pendant un mois ; passé ce délai, la version saine a disparu en ligne. Contre une erreur découverte tard — et contre une compromission du compte, qui atteindrait aussi les versions antérieures —, ce sont les disques externes qui protègent, précisément parce qu’ils sont débranchés.

⚠ `incus export` ne capture pas les disques montés depuis l’hôte. L’archive mensuelle du conteneur web contient donc son système et ses vidages de bases, mais pas le contenu de `/var/www`. Ces fichiers-là ne sont hors site que par le disque externe.

Restaurer une base de données

Le cas le plus fréquent, et le plus simple.

Les vidages sont produits avec `--databases` : ils contiennent leur propre `CREATE DATABASE` et `USE`, donc rien à créer au préalable.

Si le vidage est encore dans le conteneur — c’est le cas courant, une base

abîmée sur une machine saine.

[dans le conteneur web-srv]

```
ls -l /home/webadmin/backups/mysql_DB/
```

```
mysql < /home/webadmin/backups/mysql_DB/c1blog.sql
```

Pas de `-u` ni de `-p` : les identifiants sont dans `~/.my.cnf` du compte `webadmin`.

Si le vidage local a été perdu, le reprendre depuis le NAS. Le démon ne pouvant pas écrire, on pousse depuis l'hôte.

[nas-host]

```
ls -l /media/nas1/Backups/web-srv_MySQL_db/
```

```
incus file push --uid 0 --gid 0 \
  /media/nas1/Backups/web-srv_MySQL_db/c1blog.sql \
  web-srv/home/webadmin/backups/mysql_DB/c1blog.sql
```

Puis l'import ci-dessus, dans le conteneur.

⚠ Vérifier la taille avant d'importer. Un vidage tronqué s'importe sans erreur jusqu'à l'endroit où il s'arrête, et laisse une base à moitié remplie — plus difficile à diagnostiquer qu'une base vide.

Restaurer un site web

[nas-host]

```
ls -la /media/nas1/Backups/web-srv_www/
```

```
incus file push -r \
  /media/nas1/Backups/web-srv_www/blog/ \
  web-srv/var/www/clients/client1/web1/web/
```

⚠ Vérifier les propriétaires après restauration — c'est l'erreur classique. Les modules rsync déclarent `uid = web1` et `gid = client1` ; une restauration par un autre canal ne les rétablit pas. Un site dont les fichiers appartiennent à `root` ne fonctionne pas sous ISPConfig, et le symptôme est une erreur 500 sans rapport apparent avec les permissions.

[dans le conteneur web-srv]

```
ls -l /var/www/clients/client1/web1/web/ | head
```

```
sudo chown -R web1:client1 /var/www/clients/client1/web1/web/
```

Le contenu des sites n'étant pas dans l'archive mensuelle, si le NAS est perdu en même temps, la seule copie est le disque externe 1 — avec jusqu'à deux mois de retard.

Les fichiers de mots de passe, qui ne se voient pas

⚠ Sans eux, un site remonté répond — mais sa protection a disparu. Deux fichiers d'authentification sont dans les sauvegardes, et un `ls` ordinaire ne les montre pas : ils commencent par un point.

```
/media/nas1/Backups/Config/web-srv/etc/apache2/.htpasswd-routeur
/media/nas1/Backups/web-srv_www/webdav/utilisateur.htdigest
```

Le premier protège l'administration du routeur, le second le partage WebDAV.

Celui du WebDAV revient avec le contenu du partage ; celui du routeur se repose à la main dans `/etc/apache2/`, car il n'appartient à aucun site.

Les chercher avec `find`, jamais avec `ls` :

```
find /media/nas1/Backups -name '.htpasswd*' -o -name '*.htdigest'
```

L'export XML de WordPress n'est pas une sauvegarde du site : articles et commentaires, mais ni fichiers téléversés, ni thème, ni extensions, ni réglages. Reconstruire depuis lui seul donne le texte sans les images, dans un thème par défaut.

Restaurer un conteneur

[nas-host]

```
ls -l /media/nas1/Backups/incus-exports/
```

```
incus import /media/nas1/Backups/incus-exports/2026-09/navidrome.tar.gz
```

```
incus start navidrome
```

Trois choses ne sont pas dans l'archive, et il faut les rétablir à la main :

1. Les périphériques disque montés depuis l'hôte. Pour le conteneur web, le disque `web` doit être rattaché de nouveau — voir la page Sauvegarde, et la page de l'hôte pour la commande exacte.

2. `boot.autostart`, qui n'a pas nécessairement la valeur voulue après un

import.

3. Rien pour l’adresse IP : elle est statique dans le netplan du conteneur, donc elle voyage avec lui. ⚠ S’assurer en revanche que l’ancienne instance ne tourne pas déjà sur la même adresse — deux machines sur une même IP produisent des symptômes déroutants.

Vérifier une archive sans la restaurer, ce qui est beaucoup plus rapide :

```
tar tzf /media/nas1/Backups/incus-exports/2026-09/web-srv.tar.gz \
  > /dev/null && echo OK
```

Restaurer la configuration du serveur de musique

Deux choses seulement ont été sauvegardées, et c’est délibéré : les réglages — lecteurs déclarés, favoris, et la base des écoutes, qu’aucune réanalyse ne rend — et le code des extensions. Tout le reste du cache du serveur se refabrique.

 Le chemin de restauration n’est pas l’inverse du chemin de sauvegarde, et deux détails décident du résultat. Serveur arrêté :

[nas-host]

```
S=/media/nas1/Backups/Configuration_logiciels/lyrion
sudo systemctl stop lyrionmusicserver
sudo rsync -av "$S/prefs/" /var/lib/squeezeboxserver/prefs/
sudo cp "$S/persist.db" /var/lib/squeezeboxserver/prefs/persist.db
sudo rm -f /var/lib/squeezeboxserver/prefs/persist.db-wal \
           /var/lib/squeezeboxserver/prefs/persist.db-shm
sudo rsync -av "$S/InstalledPlugins/" \
```

```
  /var/lib/squeezeboxserver/cache/InstalledPlugins/
sudo chown -R squeezeboxserver:nogroup /var/lib/squeezeboxserver
sudo systemctl start lyrionmusicserver
```

⚠️ Les fichiers `-wal` et `-shm` d'une ancienne session doivent être retirés. Ce sont les journaux de transactions de la base précédente. Laissés à côté d'une base restaurée, SQLite tenterait de leur appliquer des écritures qui ne correspondent plus à elle — et le dégât ne se verrait qu'à l'usage, dans un historique d'écoute incohérent.

⚠️ Le `chown` final n'est pas une politesse. La copie a été faite par root ; sans lui, le serveur ne peut plus écrire dans ses propres réglages, et il démarre en donnant l'illusion de fonctionner.

🧩 Restaurer les extensions les rend sans passer par leurs dépôts, donc sans dépendre de ce qu'Internet en aura fait entre-temps. C'est exactement ce que la sauvegarde de ce répertoire achète : un dépôt tiers disparu ne se retrouve pas, et le serveur efface lui-même la liste de ce qui était installé dès qu'il démarre sans les trouver.

La première analyse après une restauration est longue — la bibliothèque et les pochettes se refabriquent entièrement. C'est normal, et c'est le prix accepté pour ne pas copier plusieurs gigaoctets chaque nuit.

Restaurer la clé de démarrage du serveur

`/boot` vit sur une clé USB, la carte mère ne sachant pas démarrer sur le NVMe en BIOS hérité. Si elle lâche, la machine ne démarre plus — et une copie déposée sur son propre disque serait inaccessible, puisqu'il faudrait démarrer pour la lire. L'image est donc répliquée hors site, chez Proton Drive.

C'est une image du périphérique entier : table de partition, secteur

d’amorçage, chargeur, contenu — et l’UUID de la partition, si bien que le `fstab` du système restauré reconnaît la nouvelle clé sans retouche. Elle est refaite à chaque changement de noyau, automatiquement ; le mécanisme est décrit dans la page Sauvegarde.

Identifier la clé neuve — l’étape à ne pas bâcler.

[nas-host]

```
lsblk -o NAME,SIZE,TYPE,MOUNTPOINTS,MODEL
```

 `dd` écrit sans rien demander et sans rien pouvoir annuler. Se tromper de périphérique détruit le disque visé en quelques secondes — et sur cette machine, les candidats voisins sont le NVMe du système et le disque de données de 12,7 To. Relever le nom et la taille dans la sortie ci-dessus, et vérifier que la taille correspond à une clé USB avant de taper la commande suivante. Dans le doute, débrancher la clé, relancer `lsblk`, la rebrancher, relancer : le nom qui apparaît est le bon.

Écrire l’image :

```
gunzip -c /chemin/vers/boot-usb-AAAA-MM-JJ.img.gz \
  | sudo dd of=/dev/sdX bs=4M status=progress conv=fsync
```

`status=progress` affiche l’avancement — sans lui, `dd` reste muet pendant plusieurs minutes. `conv=fsync` force l’écriture réelle avant de rendre la main, faute de quoi la commande peut se terminer alors que des données sont encore en tampon.

La décompression se fait à la volée, sans écrire l’image entière au passage ; `sudo` ne porte que sur `dd`, qui est seul à écrire sur le périphérique.

Vérifier avant de redémarrer :

```
sudo blkid /dev/sdX*
```

```
grep boot /etc/fstab
```

L'UUID rendu par `blkid` doit être celui qu'attend le `fstab`. S'ils diffèrent, l'image ne correspond pas à ce système.

⚠ La clé neuve doit être au moins aussi grande que l'image. Une image de 16 Go ne s'écrit pas sur une clé de 16 Go d'une autre marque, celles-ci différant de quelques mégaoctets. Prendre plus grand : l'espace excédentaire reste simplement inutilisé.

Une image périmée reste utile : elle rend la clé amorçable, avec la bonne table de partition et le bon UUID — la partie qu'on ne saurait pas refaire de mémoire. Le reste se régénère par `grub-install` et `update-grub` une fois le système démarré. Elle ne devient inutilisable que si plus aucun noyau qu'elle contient n'est présent sur le disque système.

⚠ Cette procédure n'a pas encore été éprouvée sur cette machine : elle est écrite d'après la nature de la copie, non d'après un essai réussi. L'éprouver demande une clé de rechange et un redémarrage — à faire un jour calme.

Perte totale du serveur

L'ordre compte : chaque étape dépend de la précédente.

1. Matériel et système. Réinstaller Ubuntu Server en suivant la page de l'hôte. ⚠ Si la carte mère est la même, refaire le montage `/boot` sur clé USB avant tout le reste — c'est une décision de partitionnement, elle ne se rattrape pas après coup.

2. Incus. Dépôt Zabbly `lts-6.0`, épinglé à 600 — sans quoi apt installe silencieusement la version d'Ubuntu, plus ancienne. Pool ZFS nommé `local`. Profils `default` et `macvlan`, ce dernier avec `parent: eno1`.

3. Les conteneurs, depuis Proton Drive. `incus import` pour chacun. Ils reviennent avec leurs adresses statiques, leurs configurations, et — pour le conteneur web — les vidages de bases du mois.

4. Le disque web. Recréer la partition, puis rattacher le périphérique au conteneur.

5. Le contenu des sites. Depuis le NAS si son disque est récupérable ; sinon depuis le disque externe 1, avec jusqu'à deux mois de retard. Méthode et avertissement sur les propriétaires : voir plus haut.

6. Les bases, depuis les vidages du conteneur restauré.

7. Les services de l'hôte — partages de fichiers, diffusion audio, serveur de musique — depuis les sauvegardes de configuration, si le NAS est récupérable.

⚠ Le système d'exploitation n'est restauré par aucune sauvegarde : il est reconstruit depuis le wiki. C'est un choix — une image système se périme, une procédure écrite se met à jour. Il implique que la qualité de ces pages est un composant de la sauvegarde.

Ouvrir le disque externe chiffré

Le disque 1 est chiffré avec VeraCrypt : toute restauration depuis lui exige un mot de passe.

Ce mot de passe est rangé dans deux gestionnaires indépendants — Proton Pass, hors site, et une base KeePassXC sur les appareils mobiles. Aucun des deux n'est enfermé dans le disque qu'il sert à ouvrir : la dépendance circulaire — le mot de passe du disque de secours rangé dans le disque de secours — n'existe pas ici.

⚠ Ouvrir ce disque sans son propriétaire exige un accès à Proton Pass. Le chiffrement qui rend acceptable de confier le disque à un tiers est le même qui

le rend inutilisable sans lui. C’est un arbitrage à décider à l’avance, pas dans l’urgence.

Éprouver cette page avant d’en avoir besoin

Une sauvegarde jamais restaurée est une hypothèse, pas une sauvegarde. Et une procédure de restauration jamais suivie est une fiction rassurante.

Deux fois l’an, le plus petit essai utile — il valide d’un coup l’archive, la procédure, et le souvenir qu’on en a :

[nas-host]

```
incus import \
  /media/nas1/Backups/incus-exports/2026-09/languagetool.tar.gz \
  languagetool-essai
```

```
incus start languagetool-essai
```

Vérifier que le service répond, puis supprimer l’instance d’essai. Choisir le conteneur le plus léger et le moins critique.

Le nom donné en second argument crée une instance distincte plutôt que d’écraser celle qui tourne. C’est ce qui rend l’essai sans risque.

Une fois l’an, vérifier que chaque module de sauvegarde répond encore : un module dont le chemin a disparu échoue chaque nuit sans rien dire.

```
P=/home/hostadmin/scripts/rsync/auto/rsync_pass
for m in $(rsync --list-only --password-file=$P test@192.168.0.11:: \
           | awk '{print $1}'); do
```

```
  printf '%-26s ' "$m"
  rsync --list-only --password-file=$P "test@192.168.0.11::$m/" \
    >/dev/null 2>&1 && echo OK || echo ÉCHEC
done
```

À répéter en changeant l’adresse pour chaque machine du parc. Dix secondes chacune.

Le gain : le mécanisme est éprouvé et confirmé fonctionnel, ce qui évite des sauvegardes qui échouent en silence.