Récupération — Restauration — Reconstruction

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

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.

---

1. 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 serait un vecteur de destruction, et un mot de passe compromis permettrait d'écraser les originaux depuis le réseau. Le

prix de cette sécurité est qu’on ne restaure pas en inversant une commande de sauvegarde.

⚠️ Une seule exception dans tout le parc, et elle ne sert pas à restaurer : un module d’envoi vers le répertoire des galeries du site de photos, en `read only = false`. Il existe pour téléverser des images, pas pour remettre en place une sauvegarde. Ne pas s’en servir comme chemin de retour — il ne vise qu’un répertoire, et l’employer ainsi ferait du seul module inscriptible un outil de restauration générale.

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.

---

2. 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	Miroir du mensuel, plus la clé de démarrage	Continu	Perte de la machine : incendie, vol, panne

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

⚠ Le miroir hors site n’a pas de révisions. Une archive corrompue ou effacée par erreur est répliquée telle quelle à la passe suivante. Contre l’erreur et la malveillance, 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.

---

3. 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_22.04_MySQL_db/
```

```
incus file push /media/nas1/Backups/web-srv_22.04_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.

---

4. Restaurer un site web

[nas-host]

```
ls -l /media/nas1/Backups/web-srv_22.04_www/
```

```
incus file push -r /media/nas1/Backups/web-srv_22.04_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’est pas dans l’archive mensuelle (voir plus haut). Si le NAS est perdu en même temps, la seule copie est le disque externe 1, avec jusqu’à deux mois de retard.

 L’export XML de WordPress n’est pas une sauvegarde du site. Il contient les articles, pages, commentaires et catégories — pas les fichiers téléversés, pas le thème, pas les extensions, pas les réglages. Reconstruire depuis le seul XML donne le texte sans les images, dans un thème par défaut.

---

5. 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
```

---

6. Restaurer la clé de démarrage du serveur

La carte mère ne sait pas démarrer sur le NVMe en BIOS hérité : `/boot` vit sur une clé USB. 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. C'est pourquoi cette clé est répliquée hors site, chez Proton Drive, seul ajout au périmètre de la synchronisation en dehors des exports de conteneurs.

 L'image n'est pas rafraîchie automatiquement, et elle vieillit. `/boot` change à chaque nouveau noyau — donc à chaque mise à jour automatique. Une image prise il y a des mois ne contient plus les noyaux installés depuis, et la couverture de cette clé est une photographie datée, pas une sauvegarde continue. Ce n'est pas fatal : voir la note en fin de section. Mais tant qu'un rafraîchissement régulier n'est pas en place, il faut lire « la clé est couverte » comme « la clé était couverte le jour où l'image a été prise ».

□□ La copie est une image de périphérique, prise avec `dd` — et c'est ce qui rend la restauration simple. Une image intégrale contient la table de partition, le secteur d'amorçage, le chargeur et le contenu ; elle emporte aussi l'UUID de la partition, si bien que le `fstab` du système restauré reconnaît la nouvelle clé sans retouche. Une copie fichier par fichier aurait exigé de recréer la partition, de la marquer amorçable et de réinstaller le chargeur.

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 :

```
sudo dd if=/chemin/vers/image-cle-boot.img 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.

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.

 Ce qu’une image périmée vaut encore, et c’est plus qu’on ne croit. Elle rend une clé amorçable, avec la bonne table de partition et le bon UUID — ce qui est justement la partie qu’on ne saurait pas refaire de mémoire. Le contenu, lui, se régénère depuis le système restauré : une fois démarré, fût-ce sur un noyau plus ancien que l’image contient encore, `grub-install` et `update-grub` reconstruisent le reste. Une image vieille de six mois n’est donc pas inutile, elle est une rampe de lancement. Elle cesse de l’être seulement 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. Le seul essai qui la validerait — écrire l’image sur une clé neuve et démarrer dessus — demande une clé de rechange et un redémarrage, et vaut d’être fait un jour calme plutôt que découvert un jour de panne.

---

7. 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, ni ici ni ailleurs. Il est reconstruit depuis le wiki. C'est un choix, pas un oubli : une image système se périme, une procédure écrite se met à jour et s'améliore. Mais cela veut dire que la qualité de ces pages est un composant de la sauvegarde.

---

8. Ouvrir le disque externe chiffré

Le disque 1 est chiffré avec VeraCrypt. Toute restauration depuis lui exige donc un mot de passe, et cette dépendance mérite d'être comprise avant d'en avoir besoin.

Elle est résolue, et volontairement. Deux gestionnaires indépendants contiennent les identifiants :

Gestionnaire	Où	Nature
Proton Pass	Hors site	L’ensemble des identifiants, accessible depuis n’importe quel appareil connecté
KeePassXC	Sur les appareils mobiles, propagé par Syncthing	Second gestionnaire complet, indépendant du premier

Ce qui rend le montage solide : aucun des deux gestionnaires n’est enfermé dans le disque qu’il sert à ouvrir. La dépendance circulaire classique — *le mot de passe du disque de secours est dans le disque de secours* — n’existe pas ici.

⚠ Syncthing est de la réplication, pas de la sauvegarde. Une base KeePassXC corrompue ou effacée se propage aux appareils. La redondance réelle vient de l’existence de Proton Pass comme second gestionnaire, pas d’une copie du fichier.

⚠ Et le point qu’on préfère ne pas regarder : si le disque doit être ouvert par quelqu’un d’autre, dans une situation où tu ne serais pas disponible, cette personne a besoin d’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 toi. C’est un arbitrage entre confidentialité et récupérabilité — il se décide, il ne se corrige pas.

---

9. 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 d’autre qu’une ligne dans un journal que personne ne lit.

```
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.

□□ Ce contrôle n’est pas théorique. Passé sur trente-quatre modules, il en a trouvé deux morts : l’un pointait vers un répertoire disparu quand un service a déménagé, l’autre vers un chemin de configuration qu’une application avait changé plusieurs versions plus tôt. Tous deux échouaient en silence depuis des années. Dix secondes de vérification contre des années d’illusion.