

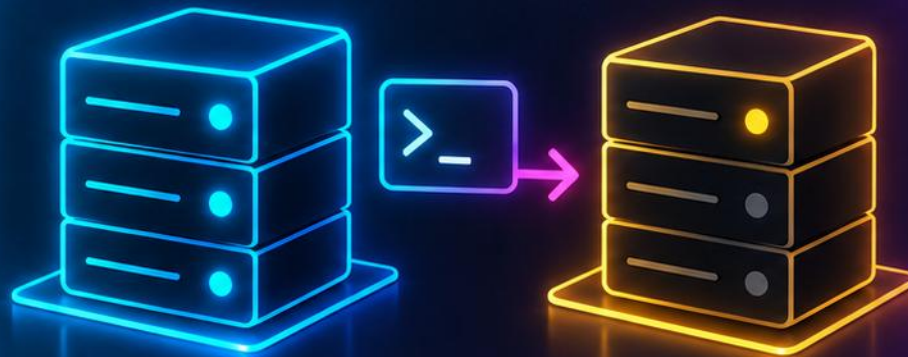
Backups : sauvegarder et restaurer vos données

Quoi copier, où ranger les copies, et comment savoir qu'elles fonctionnent



GitLab, 31 janvier 2017 : une commande lancée sur le mauvais serveur efface 300 Go

- **Un ingénieur veut vider un serveur** de base de données secondaire
- **Il se croit sur le serveur secondaire**, il est sur la production
- **Quand il arrête la commande**, environ 300 Go de la base ont disparu



GitLab avait plusieurs sauvegardes, et aucune ne pouvait servir

- L'export de chaque nuit échouait en silence : l'outil `pg_dump` était en version 9.2, la base en version 9.6
- Les e-mails d'alerte étaient rejetés avant d'arriver
- Le site est sauvé par une copie faite à la main six heures plus tôt, pour un test

Bilan : 5 000 projets, 5 000 commentaires et 700 comptes perdus, 18 heures d'arrêt.



Les données se perdent de cinq façons, pas seulement par une panne de disque

	Cause	Exemple
	Erreur humaine	Un <code>DROP TABLE</code> lancé sur la production
	Bug dans le code	Un <code>UPDATE</code> sans <code>WHERE</code>
	Panne ou sinistre	L'incendie d'un centre de données d'OVH, en mars 2021
	Règle de l'hébergeur	La base gratuite de Render expire après 30 jours
	Piratage	Un rançongiciel chiffre le serveur

La réplication, le RAID et la synchronisation ne sont pas des backups

- **Une base répliquée** recopie chaque écriture, y compris le DROP TABLE
- **Des disques en RAID** protègent d'une panne de disque, pas d'une mauvaise commande
- **Un dossier synchronisé** propage la suppression en quelques secondes



Un backup est une copie figée à une date, que l'application ne peut plus modifier.

Une application en production, ce sont quatre choses à protéger

Élément	Sa copie de sécurité
 Le code	Git et GitHub : c'est déjà fait
 La base de données	Un export , le fichier qui permet de recréer les tables et leurs lignes
 Les fichiers envoyés	Une archive, ou un second bucket , un dossier dans un stockage d'objets
 Les secrets .env	Un gestionnaire de mots de passe, jamais Git

La base et les fichiers se sauvegardent ensemble.

La règle 3-2-1 : trois copies, deux supports, une copie hors site

3

3 copies
des données :
l'original et deux
sauvegardes

2

2 supports
différents :
le disque du serveur
et un bucket

1

1 copie hors site
ailleurs que chez
l'hébergeur de
l'application

Si le serveur brûle, il reste une copie.

Le RPO est la perte de données maximale que vous acceptez



RPO = *Recovery Point Objective*, l'objectif de point de reprise.

- Un backup à 2 h 30, un incident à 14 h : tout ce qui a été écrit entre les deux est perdu
- Un backup par nuit donne un RPO de 24 heures
- Un backup par heure donne un RPO d'une heure



Le RTO est le temps d'arrêt maximal que vous acceptez

RTO = *Recovery Time Objective*, l'objectif de délai de reprise.

- Il court de l'incident au retour en ligne
- Il dépend de la taille du backup et de l'endroit où il se trouve
- Il dépend surtout d'une chose : quelqu'un a-t-il déjà fait la restauration ?



Le RPO et le RTO se décident avec le client. Ils fixent le prix de la solution.

Trois réglages de plus : la rétention, la confidentialité, l'alerte



La rétention : la durée pendant laquelle vous gardez les anciens backups. Un réglage courant : 7 quotidiens, 4 hebdomadaires, 6 mensuels



La confidentialité : un export contient les e-mails et les messages privés. Le bucket reste privé, l'export n'entre jamais dans Git



L'alerte : un backup automatique prévient quelqu'un quand il échoue

Trois types de backup : plus il est léger à faire, plus il est lourd à restaurer

Type	Ce qu'il copie	Pour restaurer
Complet	Tout, chaque jour	1 fichier
Différentiel	Les changements depuis le dernier complet	2 fichiers
Incrémental	Les changements depuis le dernier backup	Le complet et tous les incrémentaux, dans l'ordre

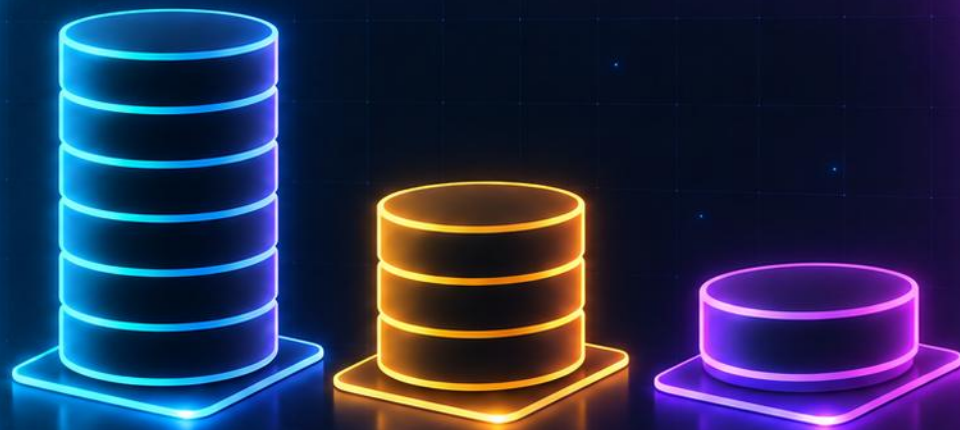


Sur une semaine, la même application occupe 7 Go, 2 Go ou 1,3 Go

Une application de 1 Go, dont 50 Mo changent chaque jour, avec un backup complet le dimanche.

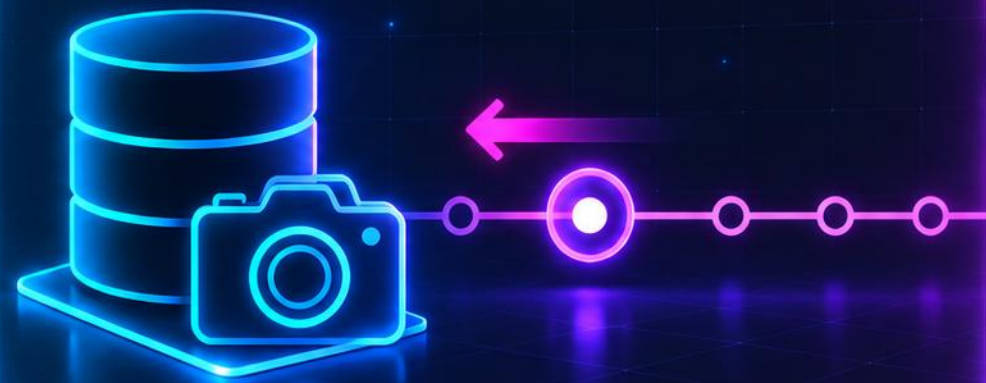
- **Complet** chaque jour : 7 Go
- **Différentiel** : environ 2 Go
- **Incrémental** : 1,3 Go

En pratique, on combine : un complet par semaine, des incrémentaux entre deux.



Instantané et restauration à un instant donné : les deux mots des hébergeurs

- Un **instantané** (*snapshot*) est une photo d'un disque entier, prise par l'hébergeur. Il se restaure vite, mais il reste chez le même hébergeur
- La **restauration à un instant donné** (*point-in-time recovery, PITR*) rejoue le journal des écritures de la base. Vous revenez à 13 h 59, une minute avant le DROP TABLE



Pour un projet étudiant, la copie hors site coûte 0 €

Service	Gratuit	Ensuite
Cloudflare R2	10 Go	0,015 \$ par Go et par mois
Backblaze B2	10 Go	0,007 \$ par Go et par mois
Amazon S3	Rien de permanent	0,023 \$ par Go et par mois
Hetzner	Rien	6,49 € par mois pour 1 To
Hostkey	Rien	4 € par mois pour 250 Go

Trente exports de 50 Mo et 2 Go de fichiers tiennent dans 10 Go. Cloudflare et Amazon demandent une carte bancaire.

Un backup automatique fait cinq choses, chaque nuit

1. **Exporter** la base dans un fichier
2. **Archiver** les fichiers envoyés
3. **Envoyer** les deux dans un bucket, hors du serveur
4. **Nettoyer** les backups plus vieux que la rétention
5. **Prévenir** quelqu'un : réussite ou échec

L'outil change selon la stack, les étapes non.



Trois pièges : la version de `pg_dump`, `rclone sync`, le bucket sans versionnement



pg_dump doit avoir la version de la base, ou une version plus récente. Sinon il s'arrête : c'est la panne de GitLab



rclone sync rend la copie identique à la source, suppressions comprises. Pour un backup, on copie avec **rclone copy**



Un fichier supprimé dans un bucket est perdu, sauf si le **versionnement** garde ses anciennes versions



Un backup qui échoue en silence est pire que pas de backup



- Le script envoie un message à la fin, par e-mail ou par **webhook**, une adresse fournie par Discord, Slack ou Teams qui affiche le texte reçu dans un salon



- Si le serveur est éteint, aucun message ne part



- Healthchecks.io fait l'inverse : il attend un signal chaque nuit et vous écrit quand le signal manque

Les cinq étapes dans un script bash, à adapter à votre installation

Pour une base PostgreSQL, des fichiers sur le disque et un bucket chez Cloudflare R2.

`/srv/chat/backup.sh` • exemple à adapter

```
set -euo pipefail
```

```
...  
trap 'notify "Backup du chat : ÉCHEC"' ERR
```

```
pg_dump --format=custom --file="$DIR/db-$STAMP.dump" "$DATABASE_URL"
```

```
tar -czf "$DIR/uploads-$STAMP.tar.gz" -C /srv/chat/storage uploads
```

```
rclone copy "$DIR" r2:chat-backups
```

```
find "$DIR" -type f -mtime +6 -delete
```

```
rclone delete r2:chat-backups --min-age 30d
```

```
notify "Backup du chat : OK ($STAMP)"
```

Cron lance ce script chaque nuit :

`30 2 * * * /srv/chat/backup.sh`

Selon l'hébergement, c'est cron, le panneau ou GitHub Actions qui lance le backup



VPS nu : cron, sur le serveur



VPS avec Coolify ou Dokploy :
le panneau, sur le serveur



PaaS ou Vercel :
GitHub Actions, de l'extérieur



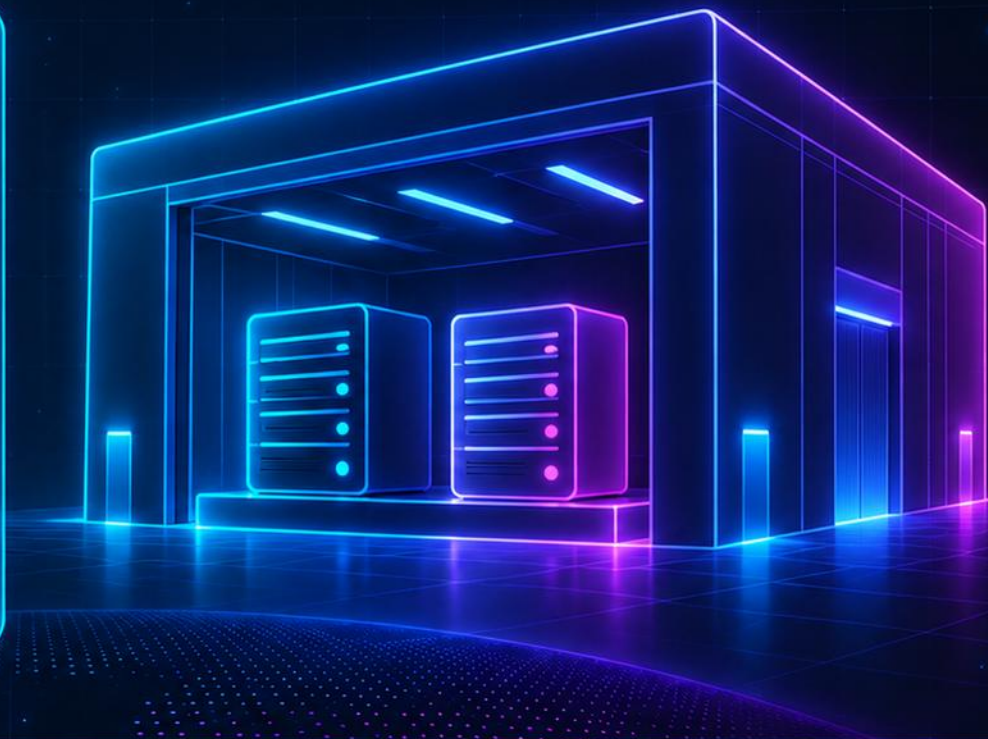
Dans tous les cas :
une copie dans un bucket.

Le backup vendu avec un VPS reste dans le même centre de données

Chez OVHcloud :

- Un **backup quotidien** du disque, gardé 24 heures, inclus avec chaque VPS
- L'**option Premium garde 7 jours**, à partir de 0,30 € HT par mois
- Les copies sont stockées dans le **même centre de données** que le VPS

C'est une copie, pas la copie hors site.



Coolify et Dokploy règlent le backup de la base dans leur interface



L'onglet Backups d'une base :

une fréquence, une rétention, un stockage S3 pour la copie hors site



Les fichiers : le panneau archive les volumes, les dossiers où un conteneur garde ses fichiers d'un déploiement à l'autre



Coolify crée ces archives de fichiers, mais n'a pas de bouton pour les restaurer



Laravel et Django ont un paquet de backup, Node n'en a pas

Stack	Outil	À savoir
 Laravel	<code>spatie/laravel-backup</code>	Exporte, envoie, nettoie, alerte. Ne restaure pas
 Django	<code>django-dbbbackup</code>	Exporte et restaure la base et les fichiers
 Node	Aucun paquet de référence	Le panneau, un script et cron, ou GitHub Actions

Suivez la documentation officielle, pas un tutoriel de 2022.

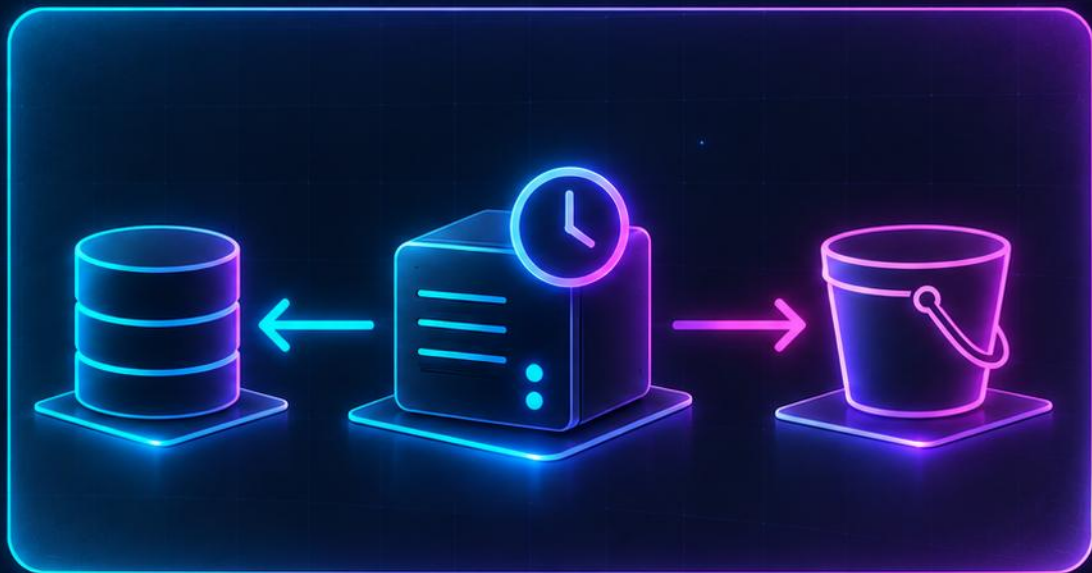
Sur un PaaS, les backups dépendent du plan que vous payez

Plateforme	Plan gratuit	Plan payant
Render	Aucun backup	Retour en arrière sur 3 ou 7 jours, dès 6 \$ par mois
Railway	Aucun backup	Backups des volumes, plan Pro à 20 \$ par mois
Neon	6 heures en arrière	Jusqu'à 7 ou 30 jours
Supabase	Aucun backup	Un backup par jour, plan Pro à 25 \$ par mois

Toutes ces copies restent chez le même fournisseur : la règle 3-2-1 n'est pas remplie.

Un workflow planifié exporte la base et l'envoie dans un bucket, sans SSH

- **Le déclencheur** `schedule` lance le workflow à heure fixe
- **La machine de GitHub** se connecte à la base, par le réseau, avec son adresse de connexion
- **Elle se connecte au bucket**, en HTTPS, avec une clé d'API
- **Limite** : dans un dépôt public, GitHub désactive les workflows planifiés après 60 jours sans activité



Tant qu'un backup n'a pas été restauré, vous ne savez pas s'il fonctionne

L'essai se fait sur votre PC, sans risque pour la production.

1. Télécharger le dernier backup du bucket

2. Restaurer dans une base locale vide

3. Vérifier les lignes, les dates, les images

4. Noter le temps et les commandes dans un fichier `RESTORE.md`

Le temps mesuré se compare à votre RTO.



L'essai de restauration se refait tous les trois mois

- **Un essai réussi** une fois ne prouve rien pour la suite
- **Entre-temps** : la base change de version, un dossier de fichiers s'ajoute, une clé d'API expire
- **Mettez un rappel** dans l'agenda du groupe
- **Refaites aussi l'essai après chaque changement important** : nouvelle version de la base, nouvel hébergeur



Pour votre projet : cinq cases à cocher, et des points en fonctionnalités supplémentaires

- ☒ Un backup automatique chaque nuit
- ☒ Une copie dans un bucket qui n'est pas chez votre hébergeur
- ☒ Une rétention d'au moins sept jours
- ☒ Un message en cas d'échec
- ☒ Une restauration essayée, notée, puis refaite tous les trois mois



Décrivez votre stratégie de backup dans le rapport et mettez-la en place.