

## Module 10

# Politique de sauvegarde





# Table des matières

Sur ce document . . . . .	1
Chers lectrices & lecteurs, . . . . .	1
À propos de DALIBO . . . . .	1
Remerciements . . . . .	2
Forme de ce manuel . . . . .	2
Licence Creative Commons CC-BY-NC-SA . . . . .	2
Marques déposées . . . . .	3
Versions de PostgreSQL couvertes . . . . .	3
<b>1/ PostgreSQL : Politique de sauvegarde</b>	<b>5</b>
1.1 Introduction . . . . .	6
1.1.1 Au menu . . . . .	6
1.2 Définir une politique de sauvegarde . . . . .	7
1.2.1 Objectifs . . . . .	8
1.2.2 Différentes approches . . . . .	8
1.2.3 RTO/RPO . . . . .	9
1.2.4 Industrialisation . . . . .	10
1.2.5 Documentation . . . . .	11
1.2.6 Règle 3-2-1 . . . . .	12
1.2.7 Autres points d'attention . . . . .	13
1.3 Conclusion . . . . .	15
1.4 Quiz . . . . .	16
<b>Les formations Dalibo</b>	<b>17</b>
Cursus des formations . . . . .	17
Les livres blancs . . . . .	18
Téléchargement gratuit . . . . .	18



## Sur ce document

<b>Formation</b>	Module I0
<b>Titre</b>	Politique de sauvegarde
<b>Révision</b>	24.04
<b>PDF</b>	<a href="https://dali.bo/i0_pdf">https://dali.bo/i0_pdf</a>
<b>EPUB</b>	<a href="https://dali.bo/i0_epub">https://dali.bo/i0_epub</a>
<b>HTML</b>	<a href="https://dali.bo/i0_html">https://dali.bo/i0_html</a>
<b>Slides</b>	<a href="https://dali.bo/i0_slides">https://dali.bo/i0_slides</a>

Vous trouverez en ligne les différentes versions complètes de ce document.

## Chers lectrices & lecteurs,

Nos formations PostgreSQL sont issues de nombreuses années d'études, d'expérience de terrain et de passion pour les logiciels libres. Pour Dalibo, l'utilisation de PostgreSQL n'est pas une marque d'opportunisme commercial, mais l'expression d'un engagement de longue date. Le choix de l'Open Source est aussi le choix de l'implication dans la communauté du logiciel.

Au-delà du contenu technique en lui-même, notre intention est de transmettre les valeurs qui animent et unissent les développeurs de PostgreSQL depuis toujours : partage, ouverture, transparence, créativité, dynamisme... Le but premier de nos formations est de vous aider à mieux exploiter toute la puissance de PostgreSQL mais nous espérons également qu'elles vous inciteront à devenir un membre actif de la communauté en partageant à votre tour le savoir-faire que vous aurez acquis avec nous.

Nous mettons un point d'honneur à maintenir nos manuels à jour, avec des informations précises et des exemples détaillés. Toutefois malgré nos efforts et nos multiples relectures, il est probable que ce document contienne des oublis, des coquilles, des imprécisions ou des erreurs. Si vous constatez un souci, n'hésitez pas à le signaler via l'adresse [formation@dalibo.com](mailto:formation@dalibo.com)<sup>1</sup> !

## À propos de DALIBO

DALIBO est le spécialiste français de PostgreSQL. Nous proposons du support, de la formation et du conseil depuis 2005.

Retrouvez toutes nos formations sur <https://dalibo.com/formations>

---

<sup>1</sup><mailto:formation@dalibo.com>

## Remerciements

Ce manuel de formation est une aventure collective qui se transmet au sein de notre société depuis des années. Nous remercions chaleureusement ici toutes les personnes qui ont contribué directement ou indirectement à cet ouvrage, notamment :

Jean-Paul Argudo, Alexandre Anriot, Carole Arnaud, Alexandre Baron, David Bidoc, Sharon Bonan, Franck Boudehen, Arnaud Bruniquel, Pierrick Chovelon, Damien Clochard, Christophe Courtois, Marc Cousin, Gilles Darold, Jehan-Guillaume de Rorthais, Ronan Dunklau, Vik Fearing, Stefan Fercot, Pierre Giraud, Nicolas Gollet, Dimitri Fontaine, Florent Jardin, Virginie Jourdan, Luc Lamarle, Denis Laxalde, Guillaume Lelarge, Alain Lesage, Benoit Lobréau, Jean-Louis Louër, Thibaut Madelaine, Adrien Nayrat, Alexandre Pereira, Flavie Perette, Robin Portigliatti, Thomas Reiss, Maël Rimbault, Julien Rouhaud, Stéphane Schildknecht, Julien Tachaires, Nicolas Thauvin, Be Hai Tran, Christophe Truffier, Cédric Villemain, Thibaud Walkowiak, Frédéric Yhuel.

## Forme de ce manuel

Les versions PDF, EPUB ou HTML de ce document sont structurées autour des slides de nos formations. Le texte suivant chaque slide contient le cours et de nombreux détails qui ne peuvent être données à l'oral.

## Licence Creative Commons CC-BY-NC-SA

Cette formation est sous licence **CC-BY-NC-SA**<sup>2</sup>. Vous êtes libre de la redistribuer et/ou modifier aux conditions suivantes :

- Paternité
- Pas d'utilisation commerciale
- Partage des conditions initiales à l'identique

### **Vous n'avez pas le droit d'utiliser cette création à des fins commerciales.**

Si vous modifiez, transformez ou adaptez cette création, vous n'avez le droit de distribuer la création qui en résulte que sous un contrat identique à celui-ci.

Vous devez citer le nom de l'auteur original de la manière indiquée par l'auteur de l'œuvre ou le titulaire des droits qui vous confère cette autorisation (mais pas d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent votre utilisation de l'œuvre). À chaque réutilisation ou distribution de cette création, vous devez faire apparaître clairement au public les conditions contractuelles de sa mise à disposition. La meilleure manière de les indiquer est un lien vers cette page web. Chacune de ces conditions peut être levée si vous obtenez l'autorisation du titulaire des droits sur cette œuvre. Rien dans ce contrat ne diminue ou ne restreint le droit moral de l'auteur ou des auteurs.

Le texte complet de la licence est disponible sur <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/legalcode>

---

<sup>2</sup><http://creativecommons.org/licenses/by-nc-sa/2.0/fr/legalcode>

Cela inclut les diapositives, les manuels eux-mêmes et les travaux pratiques. Cette formation peut également contenir quelques images et schémas dont la redistribution est soumise à des licences différentes qui sont alors précisées.

## Marques déposées

PostgreSQL® Postgres® et le logo Slonik sont des marques déposées<sup>3</sup> par PostgreSQL Community Association of Canada.

## Versions de PostgreSQL couvertes

Ce document ne couvre que les versions supportées de PostgreSQL au moment de sa rédaction, soit les versions 12 à 16.

Sur les versions précédentes susceptibles d'être encore rencontrées en production, seuls quelques points très importants sont évoqués, en plus éventuellement de quelques éléments historiques.

Sauf précision contraire, le système d'exploitation utilisé est Linux.

---

<sup>3</sup><https://www.postgresql.org/about/policies/trademarks/>





# 1/ PostgreSQL : Politique de sauvegarde



Photo de l'incendie du datacenter OVHcloud à Strasbourg du 10 mars 2021 fournie gracieusement par l'ITBR67<sup>1</sup>.

Cet incendie a provoqué de nombreux arrêts et pertes de données dans toute la France et ailleurs<sup>2</sup>.

---

<sup>1</sup><https://itbr67.fr/>

<sup>2</sup>[https://fr.wikipedia.org/wiki/Incendie\\_du\\_centre\\_de\\_donn%C3%A9es\\_d%27OVHcloud\\_%C3%A0\\_Strasbourg](https://fr.wikipedia.org/wiki/Incendie_du_centre_de_donn%C3%A9es_d%27OVHcloud_%C3%A0_Strasbourg)

## 1.1 INTRODUCTION



- Le pire peut arriver
- Politique de sauvegarde

### 1.1.1 Au menu



- Objectifs
- Approche
- Points d'attention

## 1.2 DÉFINIR UNE POLITIQUE DE SAUVEGARDE



- Pourquoi établir une politique ?
- Que sauvegarder ?
- À quelle fréquence sauvegarder les données ?
- Quels supports ?
- Quels outils ?
- Vérifier la restauration des sauvegardes

Afin d'assurer la sécurité des données, il est nécessaire de faire des sauvegardes régulières.

Ces sauvegardes vont servir, en cas de problème, à restaurer les bases de données dans un état le plus proche possible du moment où le problème est survenu.

Cependant, le jour où une restauration sera nécessaire, il est possible que la personne qui a mis en place les sauvegardes ne soit pas présente. C'est pour cela qu'il est essentiel d'écrire et de maintenir un document qui indique la mise en place de la sauvegarde et qui détaille comment restaurer une sauvegarde.

En effet, suivant les besoins, les outils pour sauvegarder, le contenu de la sauvegarde, sa fréquence ne seront pas les mêmes.

Par exemple, il n'est pas toujours nécessaire de tout sauvegarder. Une base de données peut contenir des données de travail, temporaires et/ou faciles à reconstruire, stockées dans des tables standards. Il est également possible d'avoir une base dédiée pour stocker ce genre d'objets. Pour diminuer le temps de sauvegarde (et du coup de restauration), il est possible de sauvegarder partiellement son serveur pour ne conserver que les données importantes.

La fréquence peut aussi varier. Un utilisateur peut disposer d'un serveur PostgreSQL pour un entrepôt de données, serveur qu'il n'alimente qu'une fois par semaine. Dans ce cas, il est inutile de sauvegarder tous les jours. Une sauvegarde après chaque alimentation (donc chaque semaine) est suffisante. En fait, il faut déterminer la fréquence de sauvegarde des données selon :

- le volume de données à sauvegarder et/ou restaurer ;
- la criticité des données ;
- la quantité de données qu'il est « acceptable » de perdre en cas de problème.

Le support de sauvegarde est lui aussi très important. Il est possible de sauvegarder les données sur un disque réseau (à travers SMB/CIFS ou NFS), sur des disques locaux dédiés, sur des bandes ou tout autre support adapté. Dans tous les cas, il est fortement déconseillé de stocker les sauvegardes sur les disques utilisés par la base de données.

Ce document doit aussi indiquer comment effectuer la restauration. Si la sauvegarde est composée de plusieurs fichiers, l'ordre de restauration des fichiers peut être essentiel. De plus, savoir où se trouvent les sauvegardes permet de gagner un temps important, qui évitera une immobilisation trop longue.

De même, vérifier la restauration des sauvegardes de façon régulière est une précaution très utile.

### 1.2.1 Objectifs



- Sécuriser les données
- Mettre à jour le moteur de données
- Dupliquer une base de données de production
- Archiver les données

L'objectif essentiel de la sauvegarde est la sécurisation des données. Autrement dit, l'utilisateur cherche à se protéger d'une panne matérielle ou d'une erreur humaine (un utilisateur qui supprimerait des données essentielles). La sauvegarde permet de restaurer les données perdues. Mais ce n'est pas le seul objectif d'une sauvegarde.

Une sauvegarde peut aussi servir à dupliquer une base de données sur un serveur de test ou de pré-production. Elle permet aussi d'archiver des tables. Cela se voit surtout dans le cadre des tables partitionnées où l'archivage de la table la plus ancienne permet ensuite sa suppression de la base pour gagner en espace disque.

Un autre cas d'utilisation de la sauvegarde est la mise à jour majeure de versions PostgreSQL. Il s'agit de la solution historique de mise à jour (export/import). Historique, mais pas obsolète.

### 1.2.2 Différentes approches



- Sauvegarde à froid des fichiers (ou physique)
- Sauvegarde à chaud en SQL (ou logique)
- Sauvegarde à chaud des fichiers (PITR)

À ces différents objectifs vont correspondre différentes approches de la sauvegarde.

La sauvegarde au niveau système de fichiers permet de conserver une image cohérente de l'intégralité des répertoires de données d'une instance arrêtée. C'est la sauvegarde à froid. Cependant, l'utilisation d'outils de snapshots pour effectuer les sauvegardes peut accélérer considérablement les temps de sauvegarde des bases de données, et donc diminuer d'autant le temps d'immobilisation du système.

La sauvegarde logique permet de créer un fichier texte de commandes SQL ou un fichier binaire contenant le schéma et les données de la base de données.

La sauvegarde à chaud des fichiers est possible avec le *Point In Time Recovery*.

Suivant les prérequis et les limitations de chaque méthode, il est fort possible qu'une seule de ces solutions soit utilisable. Par exemple :

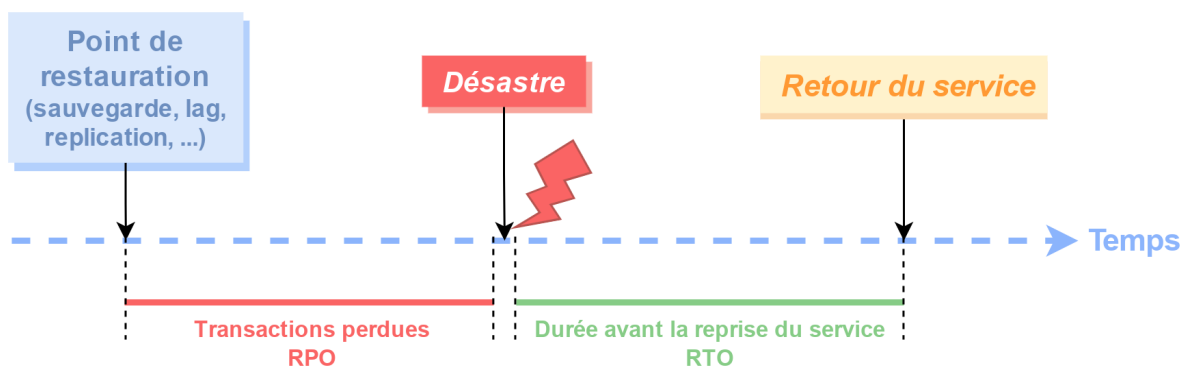
- si le serveur ne peut pas être arrêté, la sauvegarde à froid est exclue d'office ;
- si la base de données est très volumineuse, la sauvegarde logique devient très longue ;
- si l'espace disque est limité et que l'instance génère beaucoup de journaux de transactions, la sauvegarde PITR sera difficile à mettre en place.

### 1.2.3 RTO/RPO



La politique de sauvegarde découle du :

- **RPO** (*Recovery Point Objective*) : Perte de Données Maximale Admissible
  - faible ou importante ?
- **RTO** (*Recovery Time Objective*) : Durée Maximale d'Interruption Admissible
  - courte ou longue ?



Le RPO et RTO sont deux concepts déterminants dans le choix des politiques de sauvegardes.

La **RPO** (ou PDMA<sup>3</sup>) est la perte de données maximale admissible, ou quantité de données que l'on peut tolérer de perdre lors d'un sinistre majeur, souvent exprimée en heures ou minutes.

Pour un système mis à jour épisodiquement ou avec des données non critiques, ou facilement récupérables, le RPO peut être important (par exemple une journée). Peuvent alors s'envisager des solutions comme :

- les sauvegardes logiques (dump) ;
- les sauvegardes des fichiers à froid.

<sup>3</sup>[https://fr.wikipedia.org/wiki/Perte\\_de\\_donn%C3%A9es\\_maximale\\_admissible](https://fr.wikipedia.org/wiki/Perte_de_donn%C3%A9es_maximale_admissible)

Dans beaucoup de cas, la perte de données admissible est très faible (heures, quelques minutes), voire nulle. Il faudra s'orienter vers des solutions de type :

- sauvegarde à chaud ;
- sauvegarde d'instantané à un point donnée dans le temps (PITR) ;
- réplication asynchrone, voire synchrone.

La **RTO** (ou DMIA<sup>4</sup>) est la durée maximale d'interruption du service.

Dans beaucoup de cas, les utilisateurs peuvent tolérer une indisponibilité de plusieurs heures, voire jours. La durée de reprise du service n'est alors pas critique, on peut utiliser des solutions simples comme :

- la restauration des fichiers ;
- la restauration d'une sauvegarde logique (dump).

Si elle est plus courte, le service doit très vite remonter. Cela nécessite des procédures avec un minimum d'acteurs et de manipulation :

- réplication ;
- solutions HA (Haute Disponibilité).

Plus le besoin en RTO/RPO sera court, plus les solutions seront complexes à mettre en œuvre — et chères. Inversement, pour des données non critiques, un RTO/RPO long permet d'utiliser des solutions simples et peu coûteuses.

#### 1.2.4 Industrialisation



- Évaluer les coûts humains et matériels
- Intégrer les méthodes de sauvegardes avec le reste du SI
  - sauvegarde sur bande centrale
  - supervision
  - plan de continuité et de reprise d'activité

Les moyens nécessaires pour la mise en place, le maintien et l'intégration de la sauvegarde dans le SI ont un coût financier qui apporte une contrainte supplémentaire sur la politique de sauvegarde.

Du point de vue matériel, il faut disposer principalement d'un volume de stockage qui peut devenir conséquent. Cela dépend de la volumétrie à sauvegarder, il faut considérer les besoins suivants :

---

<sup>4</sup>[https://fr.wikipedia.org/wiki/Dur%C3%A9e\\_maximale\\_d%27interruption\\_admissible#RTO](https://fr.wikipedia.org/wiki/Dur%C3%A9e_maximale_d%27interruption_admissible#RTO)

- Stocker plusieurs sauvegardes. Même avec une rétention d'une sauvegarde, il faut pouvoir stocker la suivante durant sa création : il vaut mieux purger les anciennes sauvegardes une fois qu'on est sûr que la sauvegarde s'est correctement déroulée.
- Avoir suffisamment de place pour restaurer sans avoir besoin de supprimer la base ou l'instance en production. Un tel espace de travail est également intéressant pour réaliser des restaurations partielles. Cet espace peut être mutualisé. On peut utiliser également le serveur de pré-production s'il dispose de la place suffisante.

Avec une rétention d'une sauvegarde unique, il est bon de prévoir 3 fois la taille de la base ou de l'instance. Pour une faible volumétrie, cela ne pose pas de problèmes, mais quand la volumétrie devient de l'ordre du téraoctet, les coûts augmentent significativement.

L'autre poste de coût est la mise en place de la sauvegarde. Une équipe de DBA peut tout à fait décider de créer ses propres scripts de sauvegarde et restauration, pour diverses raisons, notamment :

- maîtrise complète de la sauvegarde, maintien plus aisé du code ;
- intégration avec les moyens de sauvegardes communs au SI (bandes, externalisation...);
- adaptation au PRA/PCA plus fine.

Enfin, le dernier poste de coût est la maintenance, à la fois des scripts et par le test régulier de la restauration.

### 1.2.5 Documentation



- Documenter les éléments clés de la politique :
  - perte de données
  - rétention
  - temps de référence
- Documenter les processus de sauvegarde et restauration
- Imposer des révisions régulières des procédures

Comme pour n'importe quelle procédure, il est impératif de documenter la politique de sauvegarde, les procédures de sauvegarde et de restauration ainsi que les scripts.

Au strict minimum, la documentation doit permettre à un DBA non familier de l'environnement de comprendre la sauvegarde, retrouver les fichiers et restaurer les données le cas échéant, le plus rapidement possible et sans laisser de doute. En effet, en cas d'avarie nécessitant une restauration, le service aux utilisateurs finaux est généralement coupé, ce qui génère un climat de pression propice aux erreurs qui ne fait qu'empirer la situation.

L'idéal est de réviser la documentation régulièrement en accompagnant ces révisions de tests de restauration : avoir un ordre de grandeur de la durée d'une restauration est primordial. On demandera toujours au DBA qui restaure une base ou une instance combien de temps cela va prendre.

### 1.2.6 Règle 3-2-1



- 3 exemplaires des données
- 2 sur différents médias
- 1 hors site (et hors ligne)
- Un RAID n'est pas une sauvegarde !
- Le *cloud* n'est pas une solution magique !

L'un des points les plus importants à prendre en compte est l'endroit où sont stockés les fichiers des sauvegardes. Laisser les sauvegardes sur la même machine n'est pas suffisant : si une défaillance matérielle se produisait, les sauvegardes pourraient être perdues en même temps que l'instance sauvegardée, rendant ainsi la restauration impossible.

Il est conseillé de suivre au moins la règle 3-2-1. Les données elles-mêmes sont le premier exemplaire.

Les deux copies doivent se trouver sur des supports physiques différents (et de préférence sur un autre serveur) pour parer à la destruction du support original (notamment une perte de disques durs). La première copie peut être à proximité pour faciliter une restauration.



Des disques en RAID ne sont pas une sauvegarde ! Ils peuvent parer à la défaillance d'un disque, pas à une fausse manipulation (`rm -rf /` ou `TRUNCATE` malheureux). La perte de la carte contrôleur peut entraîner la perte de toute la grappe. Un conseil courant est d'ailleurs de choisir des disques de séries différentes pour éviter des défaillances simultanées.

Le troisième exemplaire doit se trouver à un autre endroit, pour parer aux scénarios les plus catastrophiques (cambriolage, incendie...). Selon la criticité, le délai nécessaire pour remonter rapidement un système fonctionnel, et le budget, ce troisième exemplaire peut être une copie manuelle sur un disque externe stocké dans un coffre, ou une infrastructure répliquée complète avec sa copie des sauvegardes à l'autre bout de la ville, voire du pays.

Pour limiter la consommation d'espace disque des copies multiples, les durées de rétention peuvent différer. La dernière sauvegarde peut résider sur la machine, les 5 dernières sur un serveur distant, des bandes déposées dans un site sécurisé tous les mois.

Stocker vos données dans le *cloud* n'est pas une solution miracle. Un datacenter peut brûler entièrement<sup>5</sup>. Dans la formule choisie, il faut donc bien vérifier que le fournisseur sauvegarde les données

<sup>5</sup><https://dali.bo/20210310-ovh-incendie-strasbourg>



sur un autre site. Mais on a aussi vu un hébergeur perdre *toutes* les données de ses clients à cause d'un ransomware<sup>6</sup>.



N'hésitez donc pas à faire vous-même une copie de vos données. Il ne faut pas non plus négliger le risque d'une attaque (piratage, malveillance ou *ransomware*...), qui s'en prendra à toute sauvegarde accessible en ligne. Une copie physique hors ligne est donc chaudement recommandée pour les données les plus critiques.

### 1.2.7 Autres points d'attention



- Sauvegarder les fichiers de configuration
- **Tester la restauration**
  - De nombreuses catastrophes auraient pu être évitées avec un test
  - Estimation de la durée

La sauvegarde ne concerne pas uniquement les données. Il est également fortement conseillé de sauvegarder les fichiers de configuration du serveur et les scripts d'administration. Ils sont parfois générés par l'outil d'industrialisation, gérés par un outil externe (Patroni...), ou versionnés ailleurs. L'idéal est de les copier avec les sauvegardes. On peut également déléguer cette tâche à une sauvegarde au niveau système, vu que ce sont de simples fichiers. Les principaux fichiers de PostgreSQL à prendre en compte sont : `postgresql.conf`, `postgresql.auto.conf`, `pg_hba.conf`, `pg_ident.conf`. Ces fichiers ont parfois des clauses d'inclusion pour en appeler d'autres. Cette liste n'est en aucun cas exhaustive.

Il s'agit donc de recenser l'ensemble des fichiers et scripts nécessaires si l'on désiret recréer le serveur depuis zéro.

Enfin, même si les sauvegardes se déroulent correctement, il est indispensable de tester si elles se restaurent sans erreur. Une erreur de copie lors de l'externalisation peut, par exemple, rendre la sauvegarde inutilisable.

<sup>6</sup><https://dcmag.fr/cloudnordic-le-datacenter-qui-a-perdu-toutes-les-donnees-de-ses-clients-lors-dune-attaque-par-ransomware/>



*Just that backup tapes are seen to move, or backup scripts are run for a lengthy period of time, should not be construed as verifying that data backups are properly being performed.*

Que l'on voit bouger les bandes de sauvegardes, ou que les scripts de sauvegarde fonctionnent pendant une longue période, ne doit pas être interprété comme une validation que les sauvegardes sont faites.

Le test de restauration permet de vérifier l'ensemble de la procédure : ensemble des objets sauvegardés, intégrité de la copie, commandes à passer pour une restauration complète (en cas de stress, vous en oublierez probablement une partie). De plus, cela permet d'estimer la durée nécessaire à la restauration.

Nous rencontrons régulièrement en clientèle des scripts de sauvegarde qui ne fonctionnent pas, et jamais testés. Vous trouverez sur Internet de nombreuses histoires de catastrophes qui auraient été évitées par un simple test. Entre mille autres :

- une base disparaît à l'arrêt et ses sauvegardes sont vides<sup>7</sup> : erreur de manipulation, script ne vérifiant pas qu'il a bien fait sa tâche, monitoring défaillant ;
- perte de 6 heures de données chez Gitlab en 2017<sup>8</sup> : procédures de sauvegarde et de réplication incomplètes, complexes et peu claires, outils mal choisis ou mal connus, sauvegardes vides et jamais testées, distraction d'un opérateur — Gitlab a le mérite d'avoir tout soigneusement documenté<sup>9</sup>.

Restaurer régulièrement les bases de test ou de préproduction à partir des sauvegardes de la production est une bonne idée. Cela vous évitera de découvrir la procédure dans l'urgence, le stress, voire la panique, alors que vous serez harcelé par de nombreux utilisateurs ou clients bloqués.

---

<sup>7</sup>[http://www.pgdba.org/post/restoring\\_data/#a-database-horror-story](http://www.pgdba.org/post/restoring_data/#a-database-horror-story)

<sup>8</sup><https://about.gitlab.com/2017/02/01/gitlab-dot-com-database-incident/>

<sup>9</sup><https://about.gitlab.com/2017/02/10/postmortem-of-database-outage-of-january-31/>

## 1.3 CONCLUSION



- Les techniques de sauvegarde de PostgreSQL sont :
  - complémentaires
  - automatisables
- La maîtrise de ces techniques est indispensable pour assurer un service fiable.
- **Testez vos sauvegardes !**

L'écosystème de PostgreSQL offre tout le nécessaire pour effectuer des sauvegardes fiables. Le plan de sauvegarde doit être fait sérieusement, et les sauvegardes testées. Cela a un coût, mais un désastre détruisant toutes vos données sera incommensurablement plus ruineux.

## 1.4 QUIZ



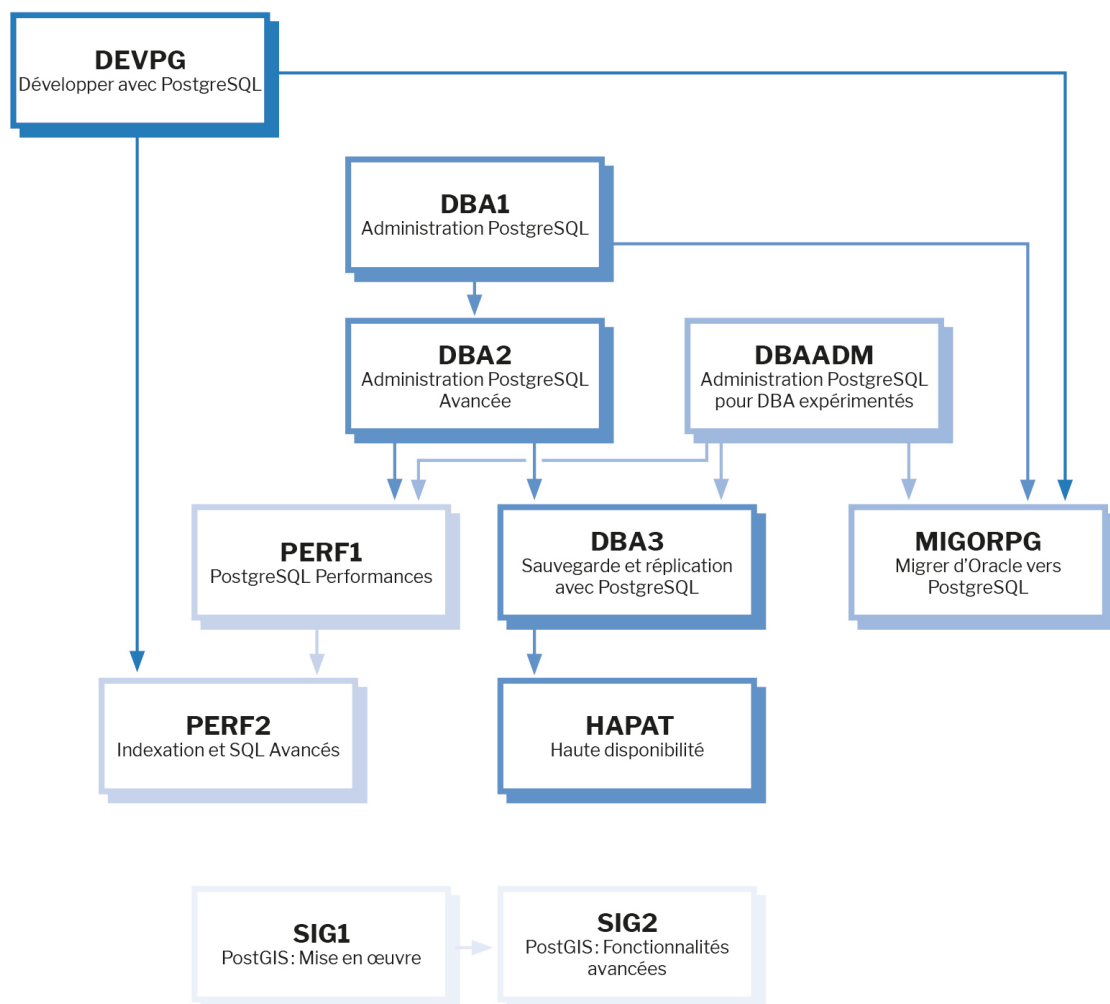
[https://dali.bo/i0\\_quiz](https://dali.bo/i0_quiz)

# Les formations Dalibo

Retrouvez nos formations et le calendrier sur <https://dali.bo/formation>

Pour toute information ou question, n'hésitez pas à nous écrire sur [contact@dalibo.com](mailto:contact@dalibo.com).

## Cursus des formations



Retrouvez nos formations dans leur dernière version :

- DBA1 : Administration PostgreSQL  
<https://dali.bo/dba1>
- DBA2 : Administration PostgreSQL avancé  
<https://dali.bo/dba2>
- DBA3 : Sauvegarde et réplication avec PostgreSQL  
<https://dali.bo/dba3>
- DEVPG : Développer avec PostgreSQL  
<https://dali.bo/devpg>
- PERF1 : PostgreSQL Performances  
<https://dali.bo/perf1>
- PERF2 : Indexation et SQL avancés  
<https://dali.bo/perf2>
- MIGORPG : Migrer d'Oracle à PostgreSQL  
<https://dali.bo/migorpg>
- HAPAT : Haute disponibilité avec PostgreSQL  
<https://dali.bo/hapat>

### Les livres blancs

- Migrer d'Oracle à PostgreSQL  
<https://dali.bo/dlb01>
- Industrialiser PostgreSQL  
<https://dali.bo/dlb02>
- Bonnes pratiques de modélisation avec PostgreSQL  
<https://dali.bo/dlb04>
- Bonnes pratiques de développement avec PostgreSQL  
<https://dali.bo/dlb05>

### Téléchargement gratuit

Les versions électroniques de nos publications sont disponibles gratuitement sous licence open source ou sous licence Creative Commons.



