

Module R55

Patroni : Architecture



24.04

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
1/ Patroni : Architecture	5
1.1 Au menu	6
1.2 Patroni	7
1.3 Réplication PostgreSQL	9
1.4 DCS	11
1.5 Deux grappes de serveurs	13
1.6 Questions	15
Les formations Dalibo	17
Cursus des formations	17
Les livres blancs	18
Téléchargement gratuit	18

Sur ce document

Formation	Module R55
Titre	Patroni : Architecture
Révision	24.04
PDF	https://dali.bo/r55_pdf
EPUB	https://dali.bo/r55_epub
HTML	https://dali.bo/r55_html
Slides	https://dali.bo/r55_slides

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.com¹ !

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>

¹<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**². 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>

²<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 déposées³ par PostgreSQL Community Association of Canada.

Versions de PostgreSQL couvertes

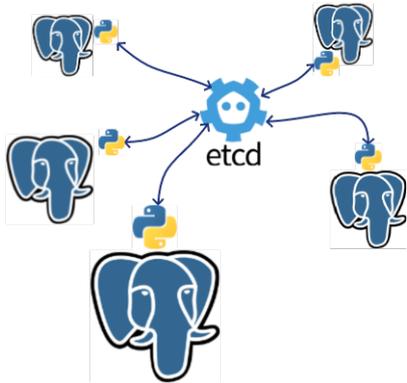
Ce document ne couvre que les versions supportées de PostgreSQL au moment de sa rédaction, soit les versions 12 à 16.

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.

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

1/ Patroni : Architecture



1.1 AU MENU



- Patroni
- Réplication PostgreSQL
- DCS

Ce module est une introduction à Patroni. Il permet de découvrir quels sont ses pré-requis, les différents éléments nécessaires, leur rôle et leurs interactions.

Les prochains chapitres abordent Patroni lui-même, son rapport avec la réplication PostgreSQL et sa dépendance à un DCS.

1.2 PATRONI



- Un démon `patroni` par instance PostgreSQL
- Les démons `patroni` coopèrent entre eux
- Chaque démon administre son instance PostgreSQL locale
 - et impose de passer par lui

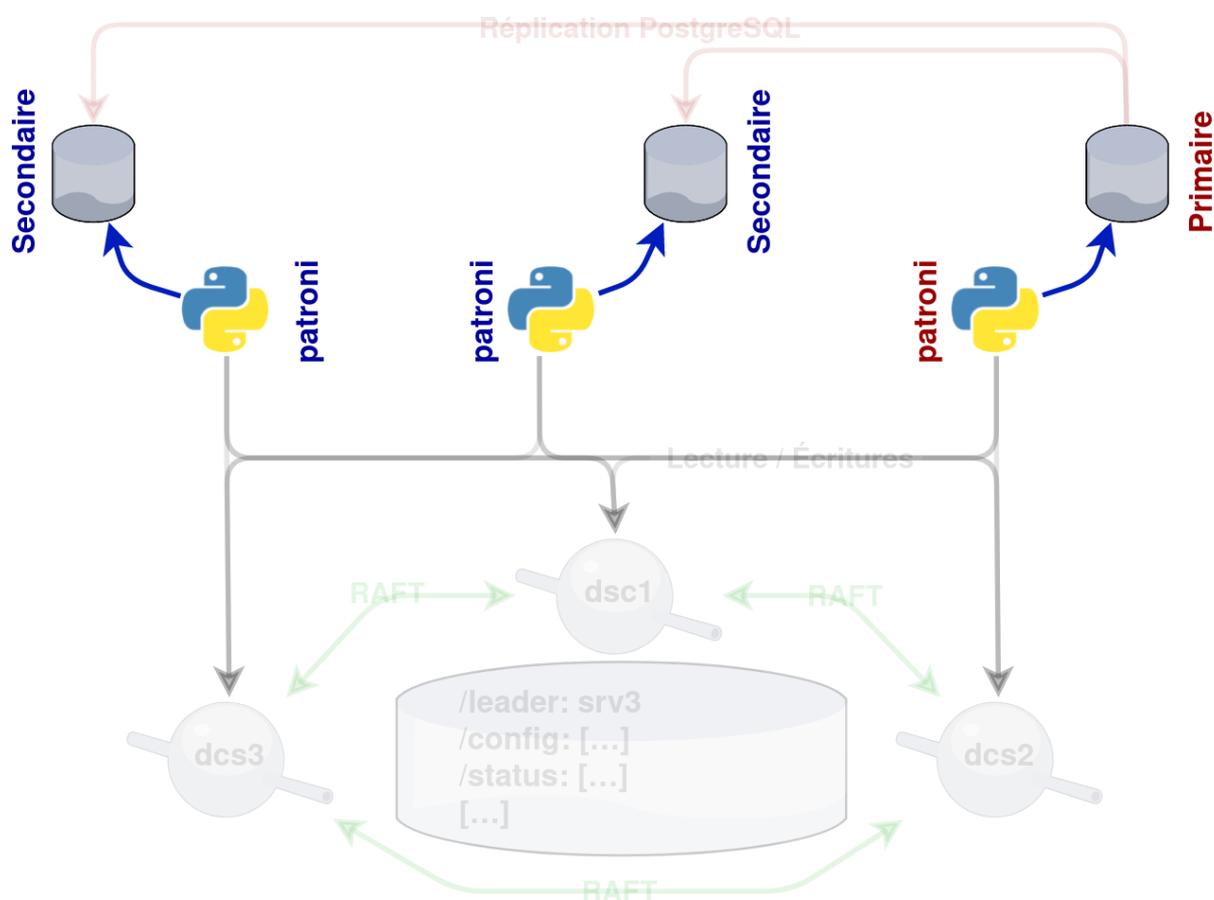


Figure 1/ .1: Étape 1 architecture Patroni

Patroni est un gestionnaire de cluster PostgreSQL, capable d'en assurer la haute disponibilité de service. Il maintient l'agrégat en condition opérationnelle, le supervise et provoque une bascule automatique en cas d'incident.

Patroni est en réalité lui-même un agrégat : chaque instance PostgreSQL du cluster est contrôlée par son propre démon `patroni`, écrit en python. Tous ces démons surveillent les évènements au sein de l'agrégat et coopèrent pour maintenir le service disponible.

Chaque démon `patroni` a la responsabilité de créer, configurer, démarrer, superviser, arrêter, promouvoir ou rétrograder son instance locale, en fonction des évènements détectés au sein du cluster.

L'application des modifications de la configuration de PostgreSQL est effectuée par Patroni qui se charge de la répercuter sur tous les nœuds.



Le démarrage et l'arrêt du service PostgreSQL sur chaque nœud ne doivent plus être gérés par le système et doivent être désactivés. Toutes les actions de maintenances (arrêt, démarrage, rechargement de configuration, promotion) doivent être faites en utilisant Patroni plutôt que les moyens traditionnels (`pg_ctl`, `systemctl`, etc).

1.3 RÉPLICATION POSTGRESQL



Patroni configure la réplication physique entre les instances pour :

- assurer la réplication des données
 - en mode synchrone et/ou asynchrone
 - aussi en cascade
- maintenir la réplication après bascule

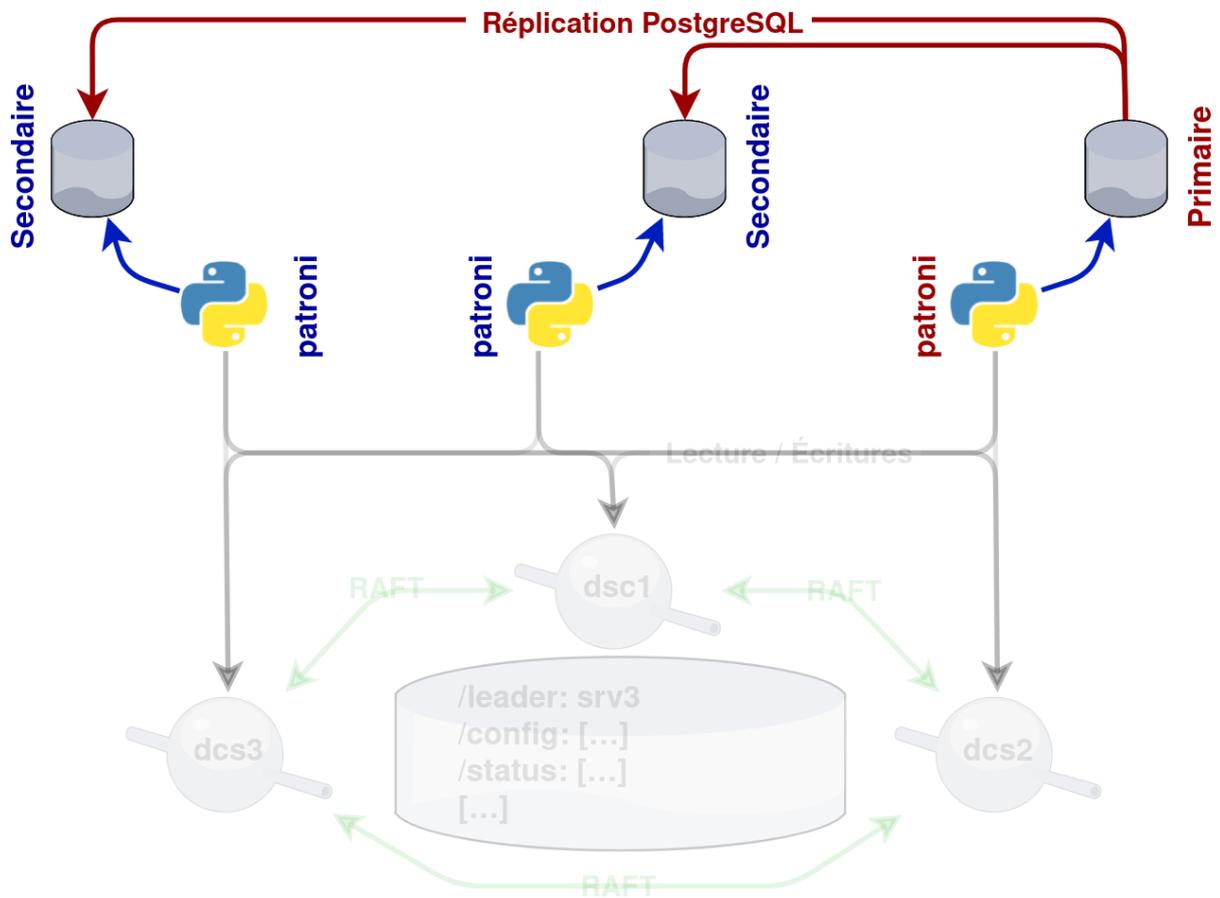


Figure 1/ .2: Étape 2 architecture Patroni

Patroni se base sur la réplication physique native de PostgreSQL pour assurer la haute disponibilité des données. Maîtrisant la création et configuration des instances, il est capable d'assurer automatiquement la mise en œuvre de cette réplication.

Patroni configure et maintient cette réplication physique en mode synchrone ou asynchrone, avec ou sans cascade de réplication, en fonction de la configuration demandée.

Suite à une bascule, il reconfigure automatiquement les secondaires afin que ceux-ci se reconnectent au nouveau primaire.

Optionnellement, il est aussi capable de resynchroniser un ancien primaire en tant que secondaire suite à un incident.

Pour effectuer ces différentes opérations, il s'appuie sur des outils fournis avec PostgreSQL comme `pg_basebackup`¹ et `pg_rewind`² pour construire ou reconstruire une instance secondaire. Mais cette capacité est encore étendue grâce à la possibilité d'utiliser des outils de la communauté comme `barman`³, `pgbackrest`⁴ ou `wal-g`⁵.

¹<https://docs.postgresql.fr/current/app-pgbasebackup.html>

²<https://docs.postgresql.fr/current/app-pgrewind.html>

³<https://www.pgbarman.org/>

⁴<https://pgbackrest.org/>

⁵<https://github.com/wal-g/wal-g>

1.4 DCS



- Stockage distribué par consensus
- écritures distribuées et atomiques
- source de vérité de l'agrégat
- hautement disponible

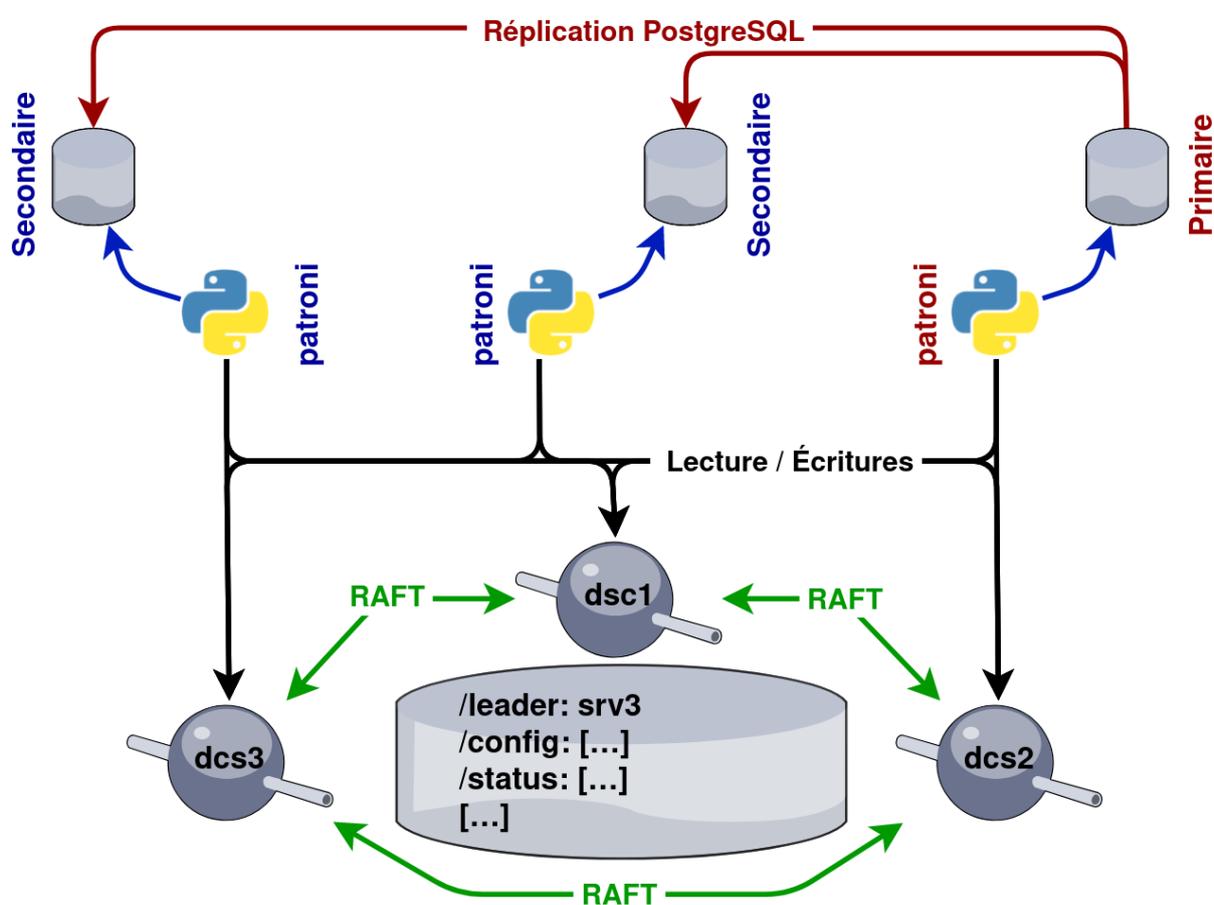


Figure 1/ .3: Étape 3 architecture Patroni

Les démons patroni s'appuient sur un stockage de données distribué (DCS, *Distributed Consensus Store*⁶) pour partager l'état des nœuds et leur configuration au sein du cluster.

Par exemple les processus Patroni y écrivent lequel d'entre eux est le *leader*, pouvant héberger

⁶ou parfois *Distributed Configuration Store*

l'instance primaire PostgreSQL. Ils reposent sur les fonctionnalités du DCS qui assurent des écritures atomiques et homogènes au sein du cluster, empêchant toute *race condition* et donc que les démons patroni ne voient des valeurs différentes.

Le DCS est la « source de vérité » du cluster, le notaire arbitrant de façon fiable et officielle le *leader* du cluster. Les processus patroni lui font entièrement confiance et respectent scrupuleusement les informations qu'ils y trouvent.

À tel point que par défaut, si le leader Patroni ne peut plus joindre le DCS, il rétrograde immédiatement son instance locale en secondaire. Si cette opération échoue ou est trop longue, il est même capable de déclencher un reset du serveur.

Le but étant la haute disponibilité, le DCS ne doit pas devenir un SPoF (*single point of failure*). Il doit donc lui aussi être déployé en agrégat afin d'assurer une tolérance aux pannes et une disponibilité maximale du service.

Patroni supporte plusieurs DCS différents, s'adaptant ainsi au mieux à l'environnement dans lequel il est déployé. En voici une liste à titre indicatif : etcd (v2 ou v3), Consul, Zookeeper, Exhibitor et Kubernetes.

1.5 DEUX GRAPPES DE SERVEURS

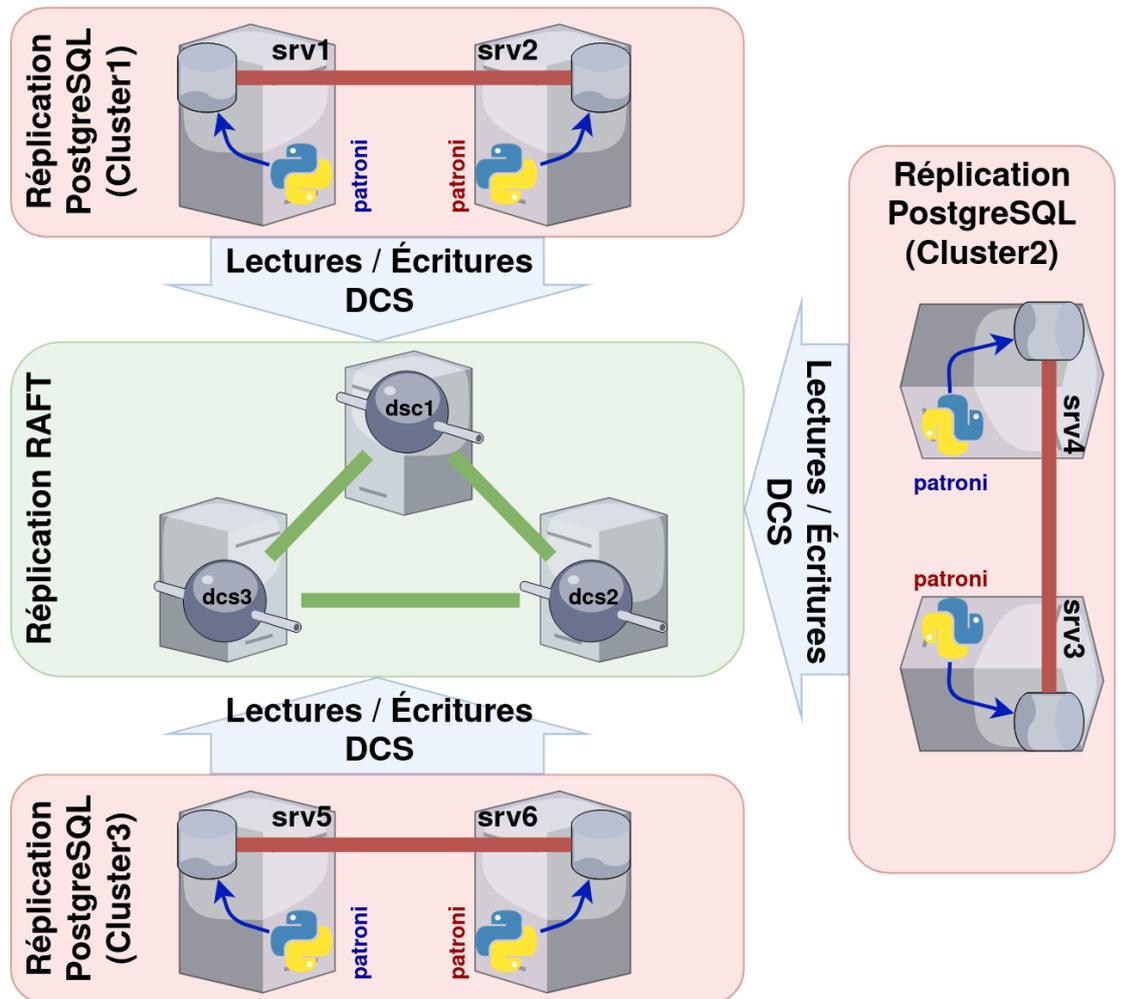


Figure 1/ .4: Agrégats de serveurs pour Patroni

Ce schéma présente les deux types d'agrégats :

- un unique agrégat de trois serveurs minimum héberge les démons DCS ;
- plusieurs agrégats PostgreSQL indépendants hébergent les démons patroni et leur instance PostgreSQL respective (2 instances minimum).



Un cluster Patroni/PostgreSQL est client du cluster DCS. Un même cluster DCS peut gérer plusieurs clusters Patroni/PostgreSQL différents.

Ces clusters ont des outils, procédures et maintenances différentes qu'il faut comprendre en détail, documenter et superviser. Différents modules de formation abordent chacun des services de cette architecture :

- un module de rappel concernant la réplication de PostgreSQL⁷ ;
- un module dédié au DCS etcd⁸ ;
- un module dédié à Patroni⁹ lui-même.

⁷https://dali.bo/r56_html

⁸https://dali.bo/r57_html

⁹https://dali.bo/r58_html

1.6 QUESTIONS



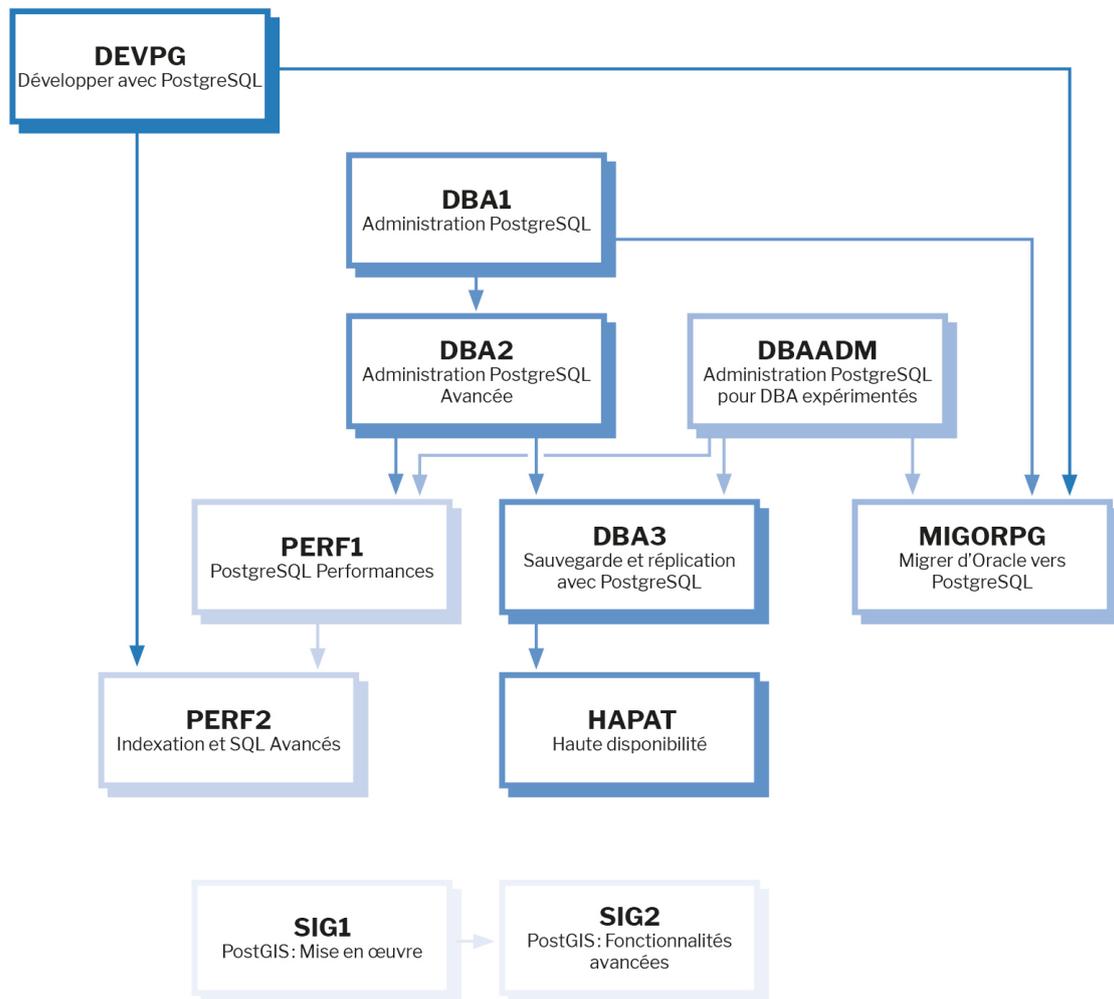
- C'est le moment !

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.

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.

